

Research Article

Forecasting CDS Term Structure Based on Nelson–Siegel Model and Machine Learning

Won Joong Kim,¹ Gunho Jung,² and Sun-Yong Choi ²

¹Department of Industrial and Management Engineering, POSTECH, Gyeongbuk 37673, Republic of Korea

²Department of Financial Mathematics, Gachon University, Gyeonggi 13120, Republic of Korea

Correspondence should be addressed to Sun-Yong Choi; sunyongchoi@gachon.ac.kr

Received 18 March 2020; Revised 18 May 2020; Accepted 20 June 2020; Published 14 July 2020

Guest Editor: Baogui Xin

Copyright © 2020 Won Joong Kim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this study, we analyze the term structure of credit default swaps (CDSs) and predict future term structures using the Nelson–Siegel model, recurrent neural network (RNN), support vector regression (SVR), long short-term memory (LSTM), and group method of data handling (GMDH) using CDS term structure data from 2008 to 2019. Furthermore, we evaluate the change in the forecasting performance of the models through a subperiod analysis. According to the empirical results, we confirm that the Nelson–Siegel model can be used to predict not only the interest rate term structure but also the CDS term structure. Additionally, we demonstrate that machine-learning models, namely, SVR, RNN, LSTM, and GMDH, outperform the model-driven methods (in this case, the Nelson–Siegel model). Among the machine learning approaches, GMDH demonstrates the best performance in forecasting the CDS term structure. According to the subperiod analysis, the performance of all models was inconsistent with the data period. All the models were less predictable in highly volatile data periods than in less volatile periods. This study will enable traders and policymakers to invest efficiently and make policy decisions based on the current and future risk factors of a company or country.

1. Introduction

A credit default swap (CDS) is a credit derivative based on credit risk, similar to a bond. The prices of both CDSs and bonds change depending on the risk of the reference entity. If the reference entity has a higher risk, then the CDS spread is set higher. To manage credit risk, we can use a CDS contract. The CDS seller (protection seller) insures the protection buyer's risk in the event of a credit default, such as bankruptcy of the reference entity, debt repudiation, or, in the case of a sovereign bond, a moratorium. There are two ways for a protection seller to compensate the protection buyer's loss. The first is to buy the underlying asset at face value; the second is to pay the difference between the remaining value and the face value. In this way, the protection buyer can hedge his or her credit risk and give the CDS spread to the protection seller.

A CDS spread is an insurance fee that a protection buyer pays to the protection seller, often quarterly. Its value is

determined by factors such as the probability of credit default and recovery rate. The recovery rate is the percentage of the bond value that the reference entity offers to the protection buyer when a credit default happens. Therefore, if the recovery rate is high, the CDS spread will be low. The CDS spread will be high if the default rate is high, which indicates a high probability of credit default. Because the CDS spread indicates the bankruptcy risk of institutions or countries, it is an important economic index that is being actively traded. According to the Bank for International Settlements, the total outstanding notional amount of CDS contracts was \$7809 billion in the first half of 2019.

To date, numerous studies have been conducted on the prediction of financial asset values. For example, Li and Tam [1] forecasted stock price movements of different volatilities using a recurrent neural network (RNN) and support vector machine (SVM). Chen et al. [2] predicted the movement of the Chinese stock market using a long short-term memory- (LSTM-) based model. Gao et al. [3] also used LSTM to

predict stock prices. However, few studies have been conducted on forecasting the CDS term structure. Shaw et al. [4] used the Nelson–Siegel model to make 1-, 5-, and 10-day forecasts of the CDS curve and compared its efficiency with that of the random-walk method. They showed that, although the 1-day forecast was not very effective, the accuracy of the 5- and 10-day forecasts outperformed those of the random-walk model. Avino and Nneji [5] predicted daily quotes of iTraxx Europe CDS indices using linear and nonlinear forecasting models, such as autoregressive (AR) and Markov switching AR models. They found that the AR model often outperforms Markov switching models, but Markov switching models offer a good in-sample fit for iTraxx index data. Sensoy et al. [6] used permutation entropy to test the weak-form efficiency of CDS markets in some countries. They found that CDS markets could be efficient during crisis periods, which implies that the impact of a crisis on CDS market efficiency is limited, and Asian markets outperformed the other tested markets in terms of efficiency. In addition, they showed a negative linear correlation between a country's CDS efficiency and daily CDS levels. Neftci et al. [7] asserted that CDS markets provide unique information on default probability. They showed that the information provided by a CDS regarding the default risk of a sovereign bond is more accurate than the information from a bond spread provided by the corresponding treasury using a stochastic differential equation based on the Markov process. Duyvesteyn and Martens [8] used the structural model for a sovereign bond from Gray et al. [9] to predict how exchange rate returns and volatility changes affect market CDS spread movements. The model results, such as default probability and spreads, were strongly correlated with CDS spreads. Their results also rejected their hypothesis that changes in sovereign credit spreads are correlated to changes in sovereign market spreads.

As mentioned above, several studies have attempted to predict various financial market indices with machine-learning methods; however, research on CDS term structure is limited. CDS term structure reflects the conditions for monetary policy and companies' future risk expectations. CDS spread can be classified into two types. The first one is sovereign CDS, which has a country as its reference entity. Sovereign CDS spreads reflect the creditworthiness of a country. That is, the sovereign CDS spread can be considered as a measure of the sovereign credit risk [10]. Furthermore, the sovereign CDS spreads contain some components that are attributed to global risk, according to Pan and Singleton [11] and Longstaff et al. [12]. Studies on sovereign CDS include Pan and Singleton [11], Longstaff et al. [12], Blommestein et al. [10], Galarionis et al. [13], Srivastava et al. [14], Ho [15], and Augustin [16]. The other type of CDS is written with respect to one single reference entity, the so-called single-name CDS. In addition, CDS sector indices are based on the most liquid 5-year term, are equally weighted, and reflect an average midspread calculation of the given index's constituents. However, single-name CDS spreads are much less liquid than indices [17–19]. In several studies, the creditworthiness of individual industries was investigated using CDS sector data [19–22].

The CDS term structure is important because it integrates the future risk expectations of both markets and companies by offering CDS spreads over time. Thus, we can confirm various types of information from the CDS term structure, such as firm leverage and volatility, as shown by Han and Zhou [23]. Furthermore, understanding the implications of the term structure also provides us with a method of extracting this information and predicting the effect of financial events and risk on it. Despite the large number of studies on CDS, studies that attempt to forecast its term structure remain few.

In this study, we analyze the CDS term structure, particularly sovereign CDS, forecast it using machine-learning models, and identify the most suitable model for predicting CDS term structure. We consider model-driven and data-driven methods: the Nelson–Siegel model, RNN, SVR, LSTM, and GMDH. The Nelson–Siegel model, as a model-driven method, was devised to fit the yield term structure; however, in this study, it was fitted to the CDS term structure to extract the term structure parameters and forecast the CDS term structure with the AR(1) model. RNN, SVR, LSTM, and GMDH are machine-learning models that specialize in predicting time-series data. RNN memorizes previous information and uses it to predict future information. LSTM is basically the same as RNN; however, it memorizes only significant information based on some calculations. SVR is derived from the structural risk minimization principle [24] and has been used for prediction in many fields [25–27]. Among the machine-learning methods, a GMDH network is a system identification method that has been used in various fields of engineering to model and forecast the nature of unknown or complex systems based on a given set of multi-input-single-output data pairs [28–30].

Machine learning is widely used in various fields to analyze data and forecast future flow. For example, Yan and Ouyang [31] compared the efficiency of the LSTM model in predicting financial time-series data with that of other machine-learning models, such as SVM and K-nearest neighbor. Baek and Kim [32], Yan and Ouyang [31], Cao et al. [33], and Fischer and Krauss [34] also analyzed and forecasted financial data using machine learning. Machine learning is widely used in medical research. Thottakkara et al. [35], Motka et al. [36], Boyko et al. [37], and Tighe et al. [38] studied and predicted various illnesses and clinical data with machine-learning models. Many studies have also been performed to predict weather conditions using machine learning. Choi et al. [39], Haupt and Kosovic [40], Rhee and Im [41], and James et al. [42] conducted research on forecasting weather conditions. Ma et al. [43] and Li et al. [44] used a convolutional neural network (CNN) to predict a transportation network. Furthermore, GMDH has been widely used for time-series prediction [45–47]. As in these studies, we will apply machine-learning methods to forecast the CDS term structure and identify the most efficient method. There are not many studies on financial data using machine-learning methods compared to other

areas, and to the best of our knowledge, this work is the first to present a forecasting model for CDS data. Therefore, although there are many prediction methods, we especially focus on methods which are generally used in the prediction of time-series data, such as LSTM, RNN, SVR, and GMDH.

Methodologically, we adopt Nelson–Siegel as a model-driven method and RNN, LSTM, SVR, and GMDH as data-driven methods to predict the CDS term structure for the period (2008–2019). We optimize the data-driven models using a grid search algorithm with the Python technological stack. Furthermore, these tests are explored using subperiod analyses to investigate changes in the model performances over the experimental period. Specifically, we split the entire sample period into two subperiods: January 2008–December 2011 (subperiod 1) and January 2012–December 2019 (subperiod 2), because subperiod 1 contains financial market turbulence due to the global financial crisis and European debt crisis. Through this subperiod analysis, we investigate the change in the forecasting performance of all methods in both high-variance and relatively low-variance data. This kind of subperiod analysis is common in other studies [48–51].

In time-series forecasting, sequence models, either RNN, LSTM, or a combination of both, are frequently used owing to considerations of time. The sequence model recognizes time as an order and can check how it changes according to the order; therefore, it can be applied to data, such as weather and finance. According to Siami-Namini and Namin [52] and McNally et al. [53], neural network (NN) models, such as RNN and LSTM, outperformed conventional algorithms, as measured by their autoregressive integrated moving averages (ARIMAs), when using financial data or bitcoin prices. McNally et al. [53] also evaluated the performance of LSTM using volatile Bitcoin data, and Cortez et al. [54] used data from the Republic of Guatemala to predict emergency events. Furthermore, LSTM is known to be better than RNN because it is modified to correct the disadvantages of RNN; however, it appears to depend on the dataset. For example, Samarawickrama and Fernando [55] demonstrated that LSTM exhibited higher accuracy than RNN when predicting stock prices. However, Selvin et al. [56] also compared RNN with LSTM in forecasting stock prices and found that RNN outperformed LSTM. Therefore, in this study, we used both RNN and LSTM to confirm whether LSTM outperforms RNN when forecasting CDS spreads. Ultimately, the motivation for conducting this study is to compare the CDS forecasting performance between the Nelson–Siegel model and the RNN, LSTM, SVR, and GMDH models, to determine the difference between model-driven and data-driven methods.

This paper is organized as follows: in the next section, we review our dataset and present a statistical summary of the CDS term structure; we describe our methods: Nelson–Siegel, RNN, SVR, LSTM, and GMDH, and we explain hyperparameter optimization and its application to the CDS term structure; Section 3 presents our forecasting results on CDS term structure with various error estimates and demonstrates the performance of each model; and Section 4 provides a summary and concluding remarks.

2. Data Description and Methods

2.1. Data Description. The CDS spread can be classified into several categories. The classification method usually depends on the frame of the credit event. The full restructuring clause is the standard term. Under this condition, any restructuring event could be a credit event. The modified restructuring clause limits the scope of opportunistic behavior by sellers when restructuring agreements do not result in a loss. While restructuring agreements are still considered as credit events, the clause limits the deliverable obligations to those with a maturity of less than 30 months after the termination date of the CDS contract. Under the modified contract option, any restructuring event, except the restructuring of bilateral loans, could be a credit event. Additionally, the modified-modified restructuring term is introduced because modified restructuring has been too severe in its limitation of deliverable obligations. Under this term, the remaining maturity of deliverable assets must be less than 60 months for restructured obligations and 30 months for all other obligations. Under the no restructuring contract option, all restructuring events are excluded under the contract as “trigger events.”

For this type of CDS, we will use a full restructuring sovereign CDS spread dataset because other datasets are unavailable for long periods. Sovereign CDS spread reflects the market participants’ perceptions of a country’s credit ratings. Our data cover the period from October 2008 to October 2019 and maturities of six months and 1, 2, 3, 4, 5, 7, 10, 20, and 30 years. All data were sourced from Datastream and correspond to the daily closing price of the CDS spread. The term structure of the CDS spread normally shows upward sloping curves, as seen in Figure 1. Furthermore, CDS spreads seem to be lower as they get closer to the current date with no exceptions. Table 1 provides summary statistics of the CDS data. We can also verify that spreads with longer maturities have higher prices in terms of both mean and percentile. It is interesting to note that the standard deviation is also higher when the maturity is longer, which implies that the market predictions are highly unstable for longer periods.

2.2. Nelson–Siegel Model. Nelson and Siegel [57] proposed a parsimonious model, and it is widely used to predict the interest rate term structure. The formula is as follows:

$$f_t(\tau) = \beta_{1t} + \beta_{2t} \frac{1 - e^{-\lambda_t \tau}}{\lambda_t \tau} + \beta_{3t} \left(\frac{1 - e^{-\lambda_t \tau}}{\lambda_t \tau} - e^{-\lambda_t \tau} \right), \quad (1)$$

where λ_t is the time-decay parameter; τ is the maturity; and β_{1t} , β_{2t} , and β_{3t} are the three Nelson–Siegel parameters. β_{1t} is the long-term component of the yield curve as it does not decay to 0 and remains constant for all maturities. β_{2t} is the short-term factor, which starts at 1 but quickly decays to 0. Finally, β_{3t} starts at 0 and increases before decaying back to 0; hence, it is medium term, which creates a hump in the yield curve.

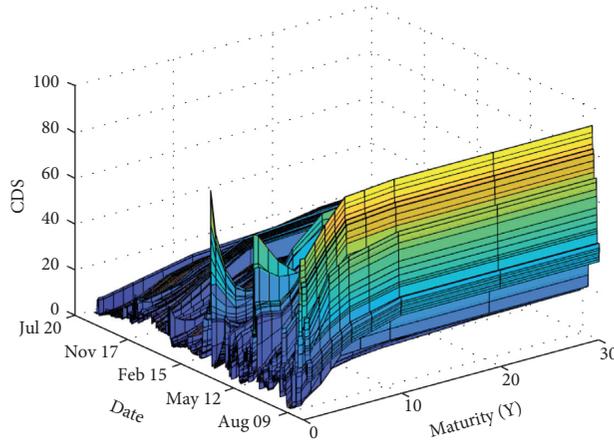


FIGURE 1: CDS term structure from 2008 to 2019.

TABLE 1: Summary statistics for the historical CDS term structure.

Index	Mean	Std. dev.	1st per.	10th per.	25th per.	Median	75th per.	90th per.	99th per.
CDS6M	12.081	9.995	1	4.745	6.263	9.122	13.688	22.98	58
CDS1Y	12.305	9.718	2	4.84	6.54	9.715	14.158	23.39	57.43
CDS2Y	14.117	10.126	4.02	6.21	7.5	11.406	16.49	23.54	62
CDS3Y	16.083	10.838	5.2	7.29	9.063	13.35	19.48	26.57	68
CDS4Y	19.038	11.557	6.29	8.57	11.33	16.005	23.04	30.743	73
CDS5Y	22.154	12.35	7.91	10.38	13.88	19.16	27.783	35.561	78
CDS7Y	26.715	11.819	11.95	14.95	18.8	23.57	32.499	40.731	78
CDS10Y	31.02	11.813	14.12	17.695	23.790	27.963	36.783	44.774	80
CDS20Y	33.903	11.986	17.83	18.77	24.33	34.01	39.597	47.952	80
CDS30Y	34.709	12.214	16.788	19.04	24.159	35.485	40.493	49.265	80

per.: percentile.

The Nelson–Siegel model is a simple but effective method for modeling a term structure, and various studies have used the model to predict the yield curve or other term structures. For example, Shaw et al. [4] forecasted CDS using the Nelson–Siegel model to fit the CDS curve. Guo et al. [58] used the Nelson–Siegel model to model the term structure of implied volatility. GrØnborg and Lunde [59] used it to model the term structure of future oil contracts and forecast the prices of these contracts, while West [60] determined the future price of agricultural commodities. In particular, the CDS term structure has a strong relationship with the interest rate term structure. For example, Chen et al. [61] found that interest rate factors not only affected credit-spread movements but also forecasted future credit risk dynamics. They claimed that the different frequency components of interest rate movements affected the CDS term structure in various industrial sectors and credit rating classes. Specifically, worsening credit conditions tend to lead to future easing of monetary policy, leading to lower current forward interest rate curves. On the contrary, positive shocks to the interest rate narrow the credit spread at long maturities. Tsuruta [62] tried to decompose the yield and CDS term structure into risk and nonrisk structures and found that credit risk components have a negative relationship to the local equity market.

In this study, we attempted to fit the CDS curve to the Nelson–Siegel model by estimating the time-decay parameter λ_t and Nelson–Siegel parameters β_{1t} , β_{2t} , and β_{3t} . We can estimate Nelson–Siegel parameters using various models, such as autoregressive-moving-average (ARMA) and ARIMA, and select the most accurate model. For example, Shaw et al. [4] used the AR(1) process to estimate β_{1t} , β_{2t} , and β_{3t} . Here, we used the AR(1) process to estimate Nelson–Siegel parameters and time-decay parameters. The error measures mean squared error (MSE), root MSE (RMSE), mean percentage error (MPE), mean absolute percentage error (MAPE), and mean absolute error (MAE) to compare the efficiency of this method with that of other methods, such as RNN or LSTM.

2.3. SVR. SVR is a field of machine-learning models derived from SVM. SVM is an algorithm that returns a hyperplane that separates the training samples into two labels, positive and negative. We refer to the distance between the closest point and the hyperplane as the “margin,” and the goal of SVM is to identify the hyperplane that maximizes the margin. There are two types of margin. The first type is a hard margin, which is for linearly separable datasets, meaning that every point does not violate its label. In other

words, all the points can be classified into their labels with a hyperplane. The second one is a soft margin, which is for nonseparable cases. In this case, some points in the dataset, called “outliers,” are incorrectly classified. There are two ways to select a soft margin hyperplane. On the one hand, we can make the margin larger and take more errors (outliers). This is usually used for datasets that have only a small number of outliers. On the other hand, we can choose a hyperplane that has a small margin and minimize the empirical errors. This is useful for datasets with dense point distributions, where it is difficult to separate the data explicitly.

Additionally, the kernel trick can be used for linearly nonseparable datasets. Kernel represents a function that maps origin data points to a higher dimensional dataset that is separable. The reason it is called the “kernel trick” is that, although the dimension of the dataset is increased, the cost of the algorithm does not increase much.

SVM originated from the statistical learning theory introduced by Vapnik and Chervonenkis. The characteristic idea of SVM is to minimize the structural risk, while artificial neural networks (ANNs) minimize the empirical risk. Furthermore, SVM theoretically demonstrates better forecasting than articular neural networks, according to Gunn et al. [63] and Haykin [64].

SVR is derived from SVM. It is a nonlinear kernel-based approach, and the main idea is to identify a function whose deviation from the actual data is located within the pre-determined scale. SVR is applied to a given dataset

$\{(x_i, y_i)\}_{i=1}^n$, where x_i is the input vector, y_i is the output, and n is the total number of data points. The following formulation was introduced by Pérez-Cruz et al. [65]. SVR assumes that the function is a nonlinear function of the form $f(x) = (w^T \phi(x) + b)$, where w and b are the weight and constant, respectively. $\phi(x)$ denotes a mapping function in the feature space. Then, weight vector w and the constant b are estimated by minimizing the following optimization problem:

$$\min_{w, b, \zeta_i, \zeta_i^*} \left\{ \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\zeta_i + \zeta_i^*) \right\}, \quad (2)$$

$$\text{subject to } y_i - (w^T \phi(x_i) + b) \leq \epsilon + \zeta_i, \quad (3)$$

$$\begin{aligned} (w^T \phi(x_i) + b) - y_i &\leq \epsilon + \zeta_i^*, \\ \zeta_i, \zeta_i^* &\geq 0, \end{aligned} \quad (4)$$

where $C > 0$ is the prespecified value and ζ_i and ζ_i^* are slack variables indicating the upper and lower constraints, respectively. Setting $\zeta_i = 0$ and $\zeta_i^* = 0$, equations (3) and (4) become the ϵ -loss function introduced by Vapnik. C is the regularization parameter, and $\phi(\cdot)$ is a nonlinear transformation to a higher dimensional space, also known as feature space.

Using Lagrange multipliers and the Karush–Kuhn–Tucker condition, the dual problem for the optimization problem (2)–(4) can be obtained:

$$\begin{aligned} \text{maximize : } L_d &= \epsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) - \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \phi^T(x_i) \phi(x_j), \\ \text{subject to} & \\ \sum_{i=1}^n (\alpha_i - \alpha_i^*) &= 0, 0 \leq \alpha_i, \alpha_i^* \leq C. \end{aligned} \quad (5)$$

To solve the above problem, we do not identify the nonlinear function $\phi(\cdot)$. The solution can be obtained as

$$f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(x, x_i) + b, \quad (6)$$

where $K(x, x_i)$ is called the kernel function, defined as $K(x_i, x_j) = (\phi(x_i) \cdot \phi(x_j))$. Any kernel function satisfying Mercer’s condition can be used as the kernel function (see Mohri et al. [66]).

The selection of the kernel has a significant impact on its forecasting performance. It is a common practice to estimate a range of potential settings and use cross-validation over the training set to determine the best one. In this research, we use three kernel functions: polynomial, Gaussian, and Sigmoid, as presented in Table 2.

Cao and Tay [67] provided a sensitivity of SVMs to the parameters C and ϵ . C and ϵ play an important role in the performance of SVR. Therefore, it is necessary to choose these parameters properly.

2.4. RNN. An ANN is a classification or prediction process that imitates human neurons. The output of a simple ANN model is generated by multiplying weights assigned to input data. After comparing the output data and the real values to be predicted, we create new weights adjusted according to the error. The step in which weights are multiplied by the input data is called forward propagation, and the step in which the error is calculated and weights are adjusted is called backpropagation. The final goal of the ANN model is to determine the weights that minimize the error between the predicted and target values.

A CNN is a machine-learning method that uses a neural network algorithm. It consists of convolution layers, pooling layers, and neural network layers. A convolution layer uses a “filter” to analyze data, typically vectorized image data. The filter analyses small sections while moving over the entire dataset, and each section expresses a “feature” of the data with pooling layers.

TABLE 2: Summary of kernels.

Types of kernel	Kernel $K(u, v)$
Polynomial kernel	$(u \cdot v)^d$
Gaussian kernel	$\exp(-\ u - v\ ^2/2\sigma^2)$
Sigmoid kernel	$\tanh(\eta u \cdot v + \gamma)$

An RNN is another representative neural network model that has a special hidden layer. While a simple neural network has a backpropagation algorithm and adjusts its weights to reduce prediction errors, the RNN has a hidden layer that is modified by the hidden layer of the previous state. Each time the algorithm operates, the RNN hidden layer affects the next hidden layer of the algorithm. Because of its characteristics, RNN is an optimized method to analyze and predict nonlinear time-series data, such as stock prices. It is an algorithm operating in sequence with input and output data. It can return a single output from one or more input data and return more than one output from one or more input data. One of its characteristics is that it returns the output in every hidden time-step layer and simultaneously sends it as input data to the next layer; we demonstrate the simplified structure in Figure 2. RNN has a memory cell in the hidden layer, which returns the output through various activation functions, such as the sigmoid and softmax functions. The memory cell memorizes the output from the previous time-step and uses it as input data recurrently. For instance, at a specific time t , the output of the previous time-step $t - 1$ and input of time-step t are used as input data, and the output is among the input data of the next time-step $t + 1$.

The greatest difference between RNN and CNN or multilayer perceptron (MLP) is that CNN and MLP do not consider previous state data in later steps, but RNN considers both the output of the previous state and the input of the present state. Furthermore, as it is optimized to deal with sequential data, it is used in text, audio, and visual data processing.

However, RNN has a vanishing gradient problem in long backpropagation processes. The algorithm of an RNN is based on gradient descent and modifies its weights in each time-step after one forward propagation process. Weights are modified with error differentials so that these rapidly converge to zero with repetitive backpropagation—this is called the vanishing gradient problem. To solve this problem in long-term time-series data, LSTM is widely used.

2.5. LSTM. To solve the vanishing gradient problem of RNN, Hochreiter and Schmidhuber [68] proposed LSTM, while Gers and Schmidhuber [69] added a forget gate to improve it. RNN considers all previous time-step memories, whereas LSTM chooses only the necessary memories to convey to the next time-step, using an algorithm in a special cell called the LSTM cell. Each of the cells has a forget gate, input gate, output gate, and long short-term memory (c_t, h_t) that pass these cells, as shown in Figure 3.

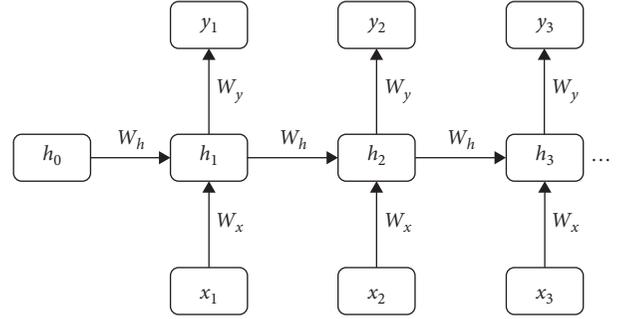


FIGURE 2: RNN cell.

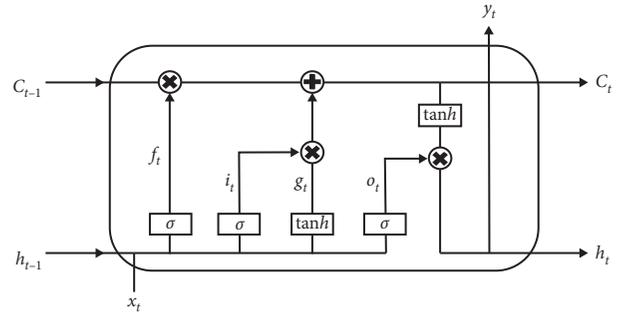


FIGURE 3: LSTM cell.

Input data x_t are deleted, filtered, and added to the long-term memory in the forget gate. The forget gate generally uses a sigmoid function as an activation function that transposes input data and short-term memory into numbers ranging from zero to one. This implies that if the output of the forget gate is close to zero, then most of the information will not pass through; if the output is close to one, then most of the information will pass to the next cell. Next, the input gates decide which data from input x_t and short-term memory must be added after substitution to g_t and i_t .

g_t generates new candidate vectors that could be added to the present cell state, and i_t decides the amount of the information that g_t generated to save. i_t uses the sigmoid function in the same way as the forget gate with the same meaning, i.e., if the value of i_t is close to one, then most of g_t will pass through, and if it is close to zero, then most g_t would not be taken in this cell. c_t is computed with the input gate value and forget gate value. By multiplying f_t with c_{t-1} , the amount of information from the previous time-step cell that will be memorized is determined. Finally, the output gate decides which data will be the output y_t of each cell, considering the memory term and o_t .

The processes performed by each gate are expressed as follows:

$$\text{Input activation : } a_t = \tanh(W_a \cdot x_t + U_a \cdot \text{out}_{t-1} + b_a)$$

$$\text{Input gate : } i_t = \sigma(W_i \cdot x_t + U_i \cdot \text{out}_{t-1} + b_i)$$

$$\text{Forget gate : } f_t = \sigma(W_f \cdot x_t + U_f \cdot \text{out}_{t-1} + b_f)$$

$$\text{Output gate : } o_t = \sigma(W_o \cdot x_t + U_o \cdot \text{out}_{t-1} + b_o)$$

W and U are the weights of x and out, respectively. For example, W_i is the weight of input data x to input gate i .

To develop an LSTM model, we must assign the initial values of c_t and h_t . As mentioned by Zimmermann et al. [70], we set both initial memory term values as zero. LSTM is broadly applied to forecast time-series data; however, owing to its complexity, Chung et al. [71] designed a simpler model called a gated recurrent unit (GRU) while adopting the advantages of LSTM. GRU consists of a reset gate, which decides how to add new input data to the previous cell memory, and an update gate, which decides the amount of memory of the previous cell to save. However, as our dataset is not very large, we used the LSTM model and compared its performance in forecasting the CDS term structure with RNN.

2.6. GMDH. GMDH is a machine-learning method based on the principle of heuristic self-organizing, proposed by Ivakhnenko [72]. The advantage of GMDH is that various considerations, including the number of layers, neurons in hidden layers, and optimal model structure, are determined automatically. In other words, we can apply GMDH to model complex systems without a priori knowledge of the systems.

Suppose that there is a set of n variables consisting of x_1, x_2, \dots, x_n and one y variable. The GMDH algorithm represents a model as a set of neurons in which different pairs in each layer are connected via quadratic polynomials, and they generate new neurons in the next layer [28, 73]. Figure 4 shows the simplified structure. The formal identification problem of the GMDH algorithm is to identify a function \hat{f} that can be used to forecast the output \hat{y} for a given input vector $X = (x_1, x_2, \dots, x_n)$ as close as possible to its actual output y instead of actual function f . Therefore, we can describe the M observations of multi-input and single output data pairs as follows:

$$y_i = f(x_{i1}, x_{i2}, \dots, x_{in}), \quad i = 1, 2, \dots, M. \quad (7)$$

We train a GMDH network to predict the output \hat{y} for any given input vector $X = (x_{i1}, x_{i2}, \dots, x_{in})$, which is given as

$$\hat{y}_i = \hat{f}(x_{i1}, x_{i2}, \dots, x_{in}), \quad i = 1, 2, \dots, M. \quad (8)$$

Now, the GMDH network is determined by minimizing the squared sum of differences between sample outputs and model predictions, that is,

$$\min \sum_{i=1}^M [\hat{f}(x_{i1}, x_{i2}, \dots, x_{in}) - y_i]^2. \quad (9)$$

The general connection between input and output variables can be expressed by a series of Volterra functions:

$$y = w_0 + \sum_{i=1}^n a_i x_i + \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j + \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n w_{ijk} x_i x_j x_k + \dots, \quad (10)$$

where $X = (x_1, x_2, \dots, x_n)$ is the input variable vector and $A = (w_1, w_2, \dots, w_n)$ is the weight vector. Equation (10) is

known as the Kolmogorov–Gabor polynomial [28, 45, 72, 74, 75].

In this study, we use the second-order polynomial function of two variables, which is written as

$$\hat{y} = G(x_i, x_j) = w_0 + w_1 x_i + w_2 x_j + w_3 x_i x_j + w_4 x_i^2 + w_5 x_j^2. \quad (11)$$

The main objective of the GMDH network is to build the general mathematical relation between the inputs and output variables given in equation (10). The weights a_i in equation (11) are estimated using regression techniques so that the difference between actual output (y) and the calculated output (y_i) is minimized, described as

$$e = \sum_{i=1}^n (\hat{y}_i - y_i) \longrightarrow \min. \quad (12)$$

These parameters can be obtained from multiple regression using the least squares method, and we can compute them by solving some matrix equations. Refer to [28, 29, 46, 76] for a detailed description of the parameter estimation process. The GMDH network can be associated with various algorithms, such as the genetic algorithm [77, 78], singular value decomposition [28], and back-propagation [29, 46, 73, 79–81]. We also improved the GMDH network using backpropagation.

2.7. Hyperparameter Optimization. Hyperparameter optimization refers to the problem of determining the optimal values of hyperparameters that must be set up in advance to perform training and that can complete the generalized performance of the training model to the highest level. In the deep-learning model, for example, the learning rate, batch size, etc. can be regarded as hyperparameters, and in some cases, they can be added as targets for exploration as hyperparameters that determine the structure of the deep-learning model, such as the number of layers and the convolution filter size. Hyperparameter optimization typically includes manual search, grid search, and random search.

Manual search is a way for users to set hyperparameters individually and compare performances according to their intuition. After selecting the candidate hyperparameter values and performing training using them, the performance results measured against the verification dataset are recorded, and this process is repeated several times to select the hyperparameter values that demonstrate the highest performance. This is the most intuitive method; however, it has some problems. First, it is relatively difficult to ensure that the optimal hyperparameter value to be determined is actually optimal because the process of determining the optimal hyperparameter is influenced by the user's selections. Second, the problem becomes more complicated when attempting to search for several types of hyperparameters at once. Because there are some types of hyperparameters that have mutually affecting relationships with others, it is difficult to apply an existing intuition to each single hyperparameter.

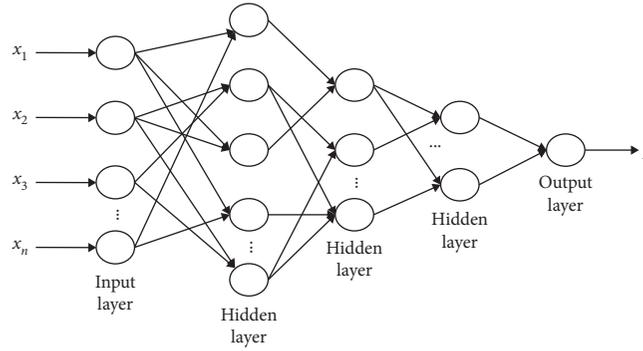


FIGURE 4: GMDH cell.

Grid search is a method of selecting candidate hyperparameter values within a specific section to be searched at regular intervals, recording the performance results measured for each of them, and selecting the hyperparameter values that demonstrated the highest performance (see Hsu et al. [82]). The user determines the search target, length of the section, interval, etc., but more uniform and global search is possible than in the previous manual search. On the contrary, the more the hyperparameters to be searched that are set at one time, the longer the overall search time, and it increases exponentially.

Random search (see Bergstra and Bengio [83]) is similar to grid search but differs in that the candidate hyperparameter values are selected through random sampling. This method can reduce the number of unnecessary repetitions and simultaneously search for values located between predetermined intervals so that the optimal hyperparameter value can be determined more quickly. Random search has the disadvantage that unexpected results can be obtained by testing various combinations other than the values set by the user.

The grid search and random search algorithms are illustrated in Figure 5. In this study, we use the grid search algorithm because it is the simplest and is most widely used for determining optimal hyperparameters [84]. Although a random search can perform much better than grid search for high-dimensional problems, according to Hutter et al. [85], our data are simple time-series data, and the candidate parameter set is limited; thus, we use the grid search algorithm [86, 87]. The Python technological stack was used for experiments. We implemented the machine-learning algorithms and grid search via “Keras,” “TensorFlow,” and “GmdhPy.”

3. Empirical Results

We used 2886 daily time-series data points on CDS term structure from October 2008 to October 2019. Because international financial markets from 2008 to 2011 were unstable, we divided these data into two subperiods, and we measured the forecasting performance of the five methods we used in both high-variance and relatively low-variance data. The first training dataset is from 1st October 2008 to 22nd January 2019 (full period), the second one is from 1st

October 2008 to 9th September 9th 2011 (subperiod 1), and the third one is from 2nd January 2012 to 22nd January 2019 (subperiod 2). We selected our test dataset as the last 200 days (from 23rd January 2019 to 29th October 2019, test dataset 1) for each maturity in the full period, the subperiod 2, and last 80 days (from 12th September 2011 to 30th December 2011, test dataset 2) for the subperiod 1. There is a gap between subperiod 1 and subperiod 2 because of the test dataset 2 for subperiod 1 training set. These all cases are summarized in Table 3. Summary statistics for the test dataset are provided in Tables 4 and 5. Test dataset 2 has higher standard deviations than test dataset 1. Through this subperiod analysis, we compared the prediction power of the models in a relatively volatile period (subperiod 2) and a less volatile period (subperiod 1). We used grid search to optimize the parameters in RNN, LSTM, SVR, and GMDH and calculated the RMSE, MSE, MAPE, MPE, and MAE to compare the performance of these five models. Figures 6–11 show the performance of the Nelson–Siegel, RNN, LSTM, SVR, and GMDH models with the test datasets for each maturity.

Our main findings can be summarized as follows: first, as shown in Figures 6–11, every model provides accurate predictions of CDS term structure. Figures 12–14 also show that machine-learning methods have similar accuracy and outperformed the Nelson–Siegel with AR(1) model. This proves that machine-learning models can be applied to forecasting CDS time-series data and that the Nelson–Siegel model fits both the interest rate term structure and CDS term structure. Furthermore, GMDH, SVR, and RNN have very similar accuracies in all periods and maturities. Second, comparing the Nelson–Siegel model with the four machine-learning methods in predictive power, the Nelson–Siegel model shows the poorest performance for all test sets. That is, machine-learning algorithms are more effective in predicting CDS spread than the Nelson–Siegel model, based on interest rate term structures, which play an important role in determining CDS spread levels. Third, among the machine-learning methods, GMDH presents the best prediction results. The error of the GMDH was found to be the lowest among the five methods, as shown in Tables 6–8. In addition, we expected LSTM to outperform RNN, but the RNN model slightly outperformed the LSTM model. However, this result remains debatable, as mentioned in Introduction.

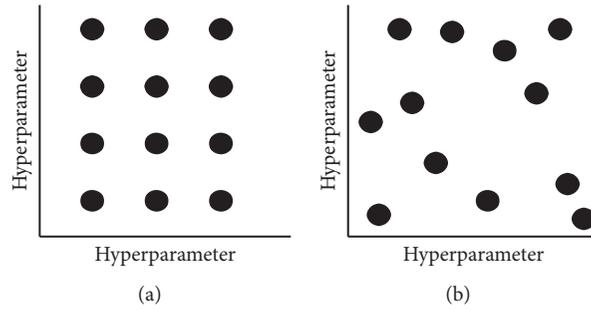


FIGURE 5: Comparison between (a) grid search and (b) random search.

TABLE 3: Descriptions of the training and test datasets.

Case	Training set	Test dataset
Case 1	Full period (2008/10/01–2019/01/22)	Test dataset 1 (2019/01/23–2019/10/29)
Case 2	Subperiod 1 (2008/10/01–2011/09/09)	Test dataset 2 (2011/09/12–2011/12/30)
Case 3	Subperiod 2 (2012/01/02–2019/01/22)	Test dataset 1 (2019/01/23–2019/10/29)

TABLE 4: Summary statistics for test dataset 1 for the full period and subperiod 2 training sets.

	CDS6M	CDS1Y	CDS2Y	CDS3Y	CDS4Y	CDS5Y	CDS7Y	CDS10Y	CDS20Y	CDS30Y
Mean	5.18	5.07	5.42	6.32	7.52	9.25	13.22	15.93	18.39	18.76
Std. dev.	1.84	1.85	1.92	1.86	1.78	1.56	1.21	1.52	1.43	1.89

TABLE 5: Summary statistics for test dataset 2 for the subperiod 1 training set.

	CDS6M	CDS1Y	CDS2Y	CDS3Y	CDS4Y	CDS5Y	CDS7Y	CDS10Y	CDS20Y	CDS30Y
Mean	13.99	14.44	19.74	25.55	30.83	35.83	40.10	44.76	48.17	49.23
Std. dev.	3.77	4.02	3.85	3.73	3.79	4.44	4.98	4.63	3.42	3.20

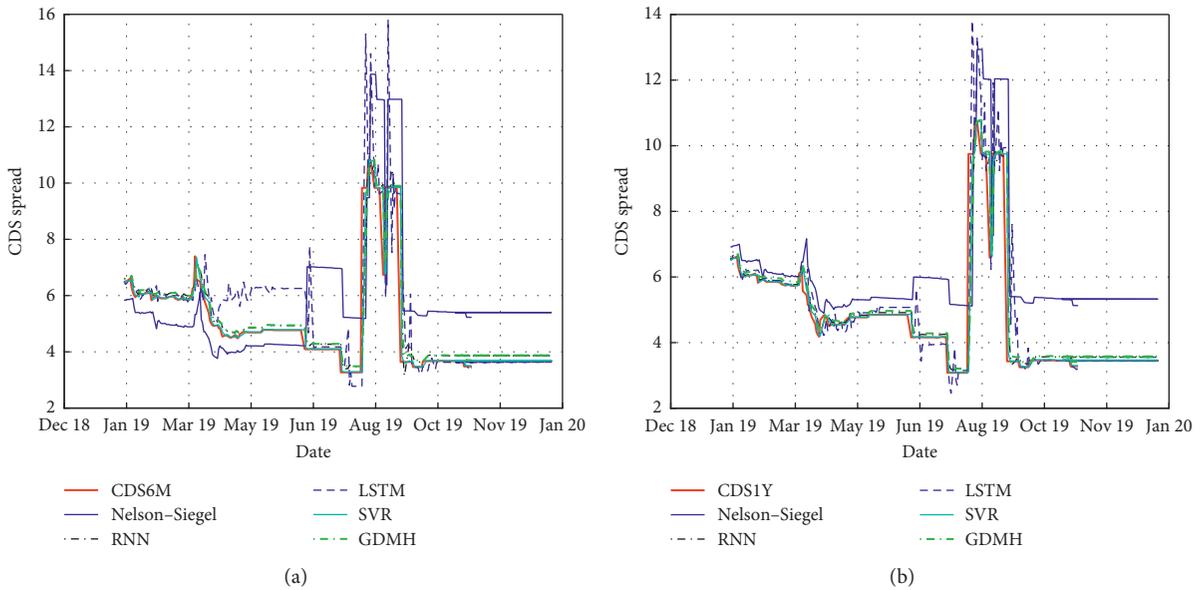


FIGURE 6: Continued.

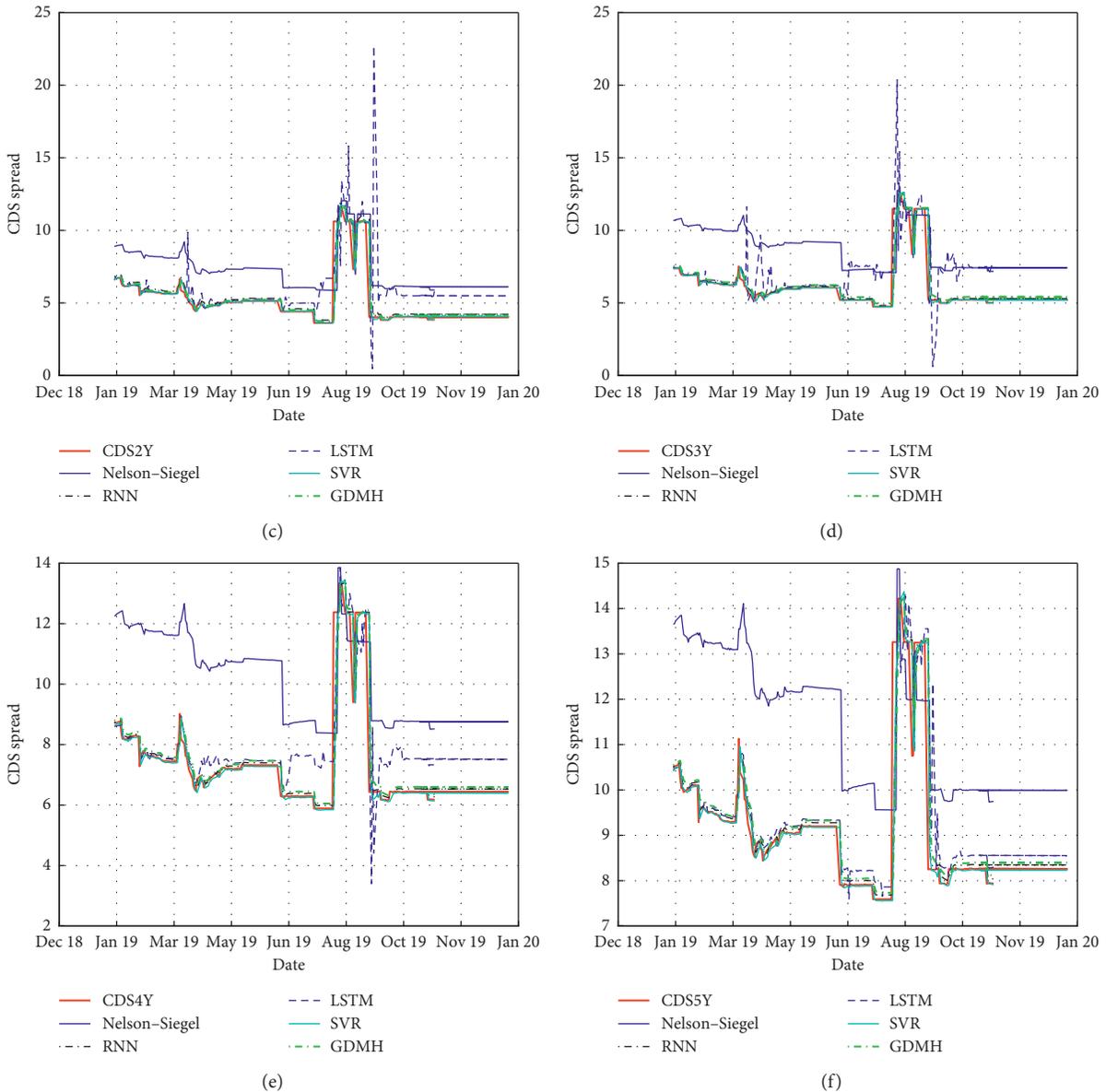


FIGURE 6: Predictions of each model and target CDS term structure from six months to five years maturity for test dataset 1 with the full period training set (case 1).

Performance comparisons between machine-learning algorithms are finding different conclusions in different studies [55, 56, 88–91]. Fourth, the periods with higher standard deviations are generally harder to predict accurately, as seen in Tables 7 and 8. Additionally, the maturities with higher standard deviations are generally harder to predict accurately, as seen in Figures 12–14. The changes in the standard deviation and in the forecasting error are similar for most error measures except MAPE and MPE, as shown in Figure 13.

4. Summary and Concluding Remarks

The purpose of this study is to compare the prediction of CDS term structure between the Nelson-Siegel, RNN,

LSTM, SVR, and GMDH models. We determined the most suitable model to predict time-series data, especially the CDS term structure. The CDS spread is a default risk index for a country or company; hence, this study is useful because it not only offers the best time-series forecasting model but also predicts future risk.

Existing studies on the prediction of CDS term structure and other risk indicators using machine-learning models remain few; most focus on stock price prediction. This study is significant because it demonstrated that various machine-learning models can be applied to other time-series data, and further research on various time-series data using machine-learning models is expected. This study also confirmed that data-driven methods, such as RNN, LSTM, SVR, and GMDH,

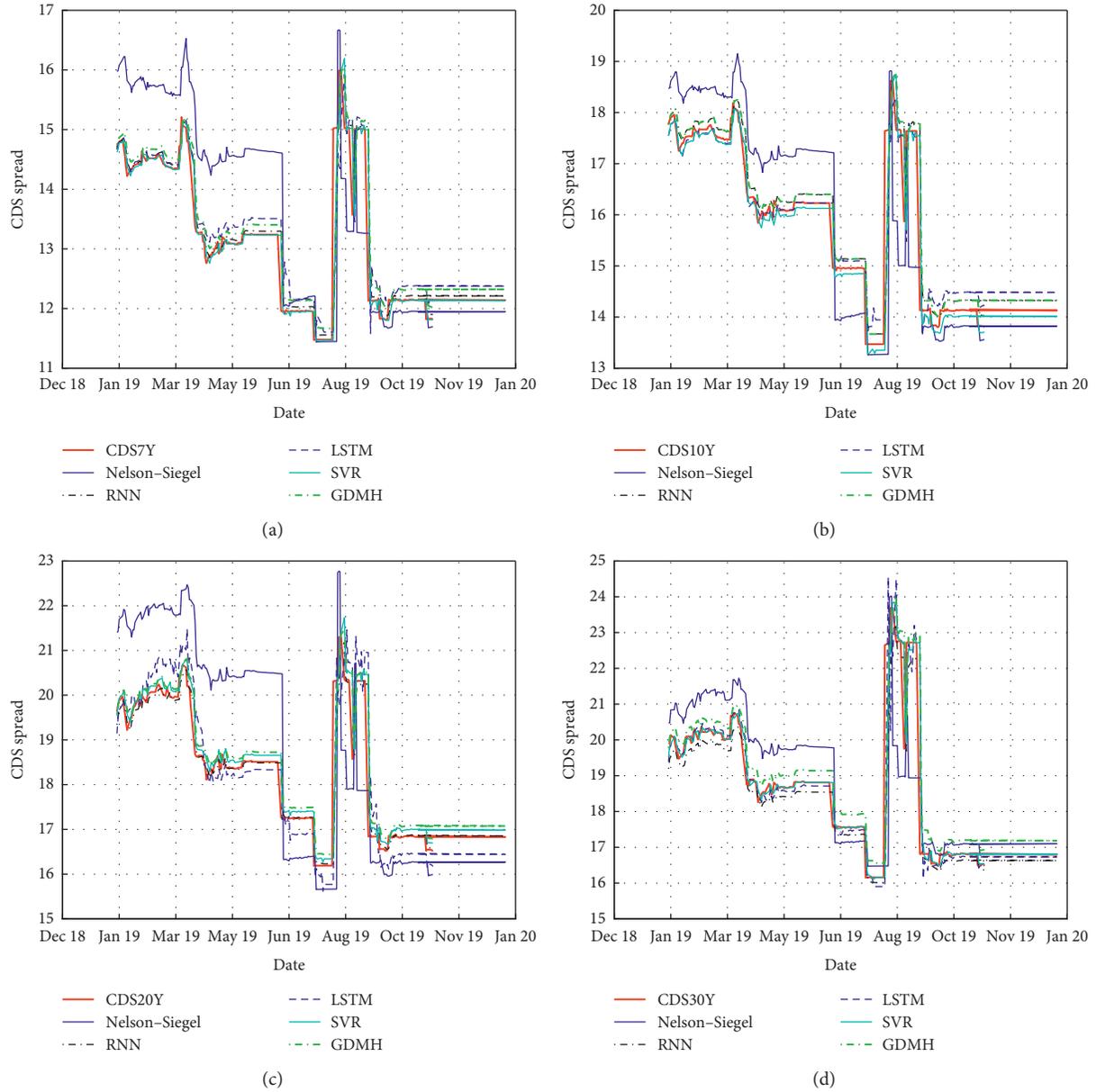


FIGURE 7: Predictions of each model and target CDS term structure from seven years to 30 years maturity for test dataset 1 with the full period training set (case 1).

outperform the model-driven Nelson–Siegel method, which is usually used in analyzing the CDS term structure. The performance of model-driven methods could decline if the data have a significant number of outliers because it is dependent on the assumption that the dataset can be formalized on a specific formula. In our dataset, the presence of outliers made it difficult to make predictions with model-driven methods. On the contrary, data-driven methods were not affected by outliers (see Solomatine et al. [92]), as these consider only datasets that include outliers. As most data available today have many outliers, it is not surprising that data-driven methods outperform model-driven ones.

Some studies show that linear models such as AR are better than ANNs [93–95] for forecasting time series. However, CDS series data are not persistent and volatile, as shown in Figure 1, so Nelson–Siegel based on the AR process performs more poorly than the machine-learning methods. In other words, because of the nonlinearity, machine-learning techniques can be successfully used for modeling and forecasting time series [96–100].

Based on the empirical findings given in Section 3, we have three implications. The first is that the data-driven method is more effective in predictive power than the theoretical model consisting of theoretical variables that influence a financial asset’s price. Of course, the data-driven

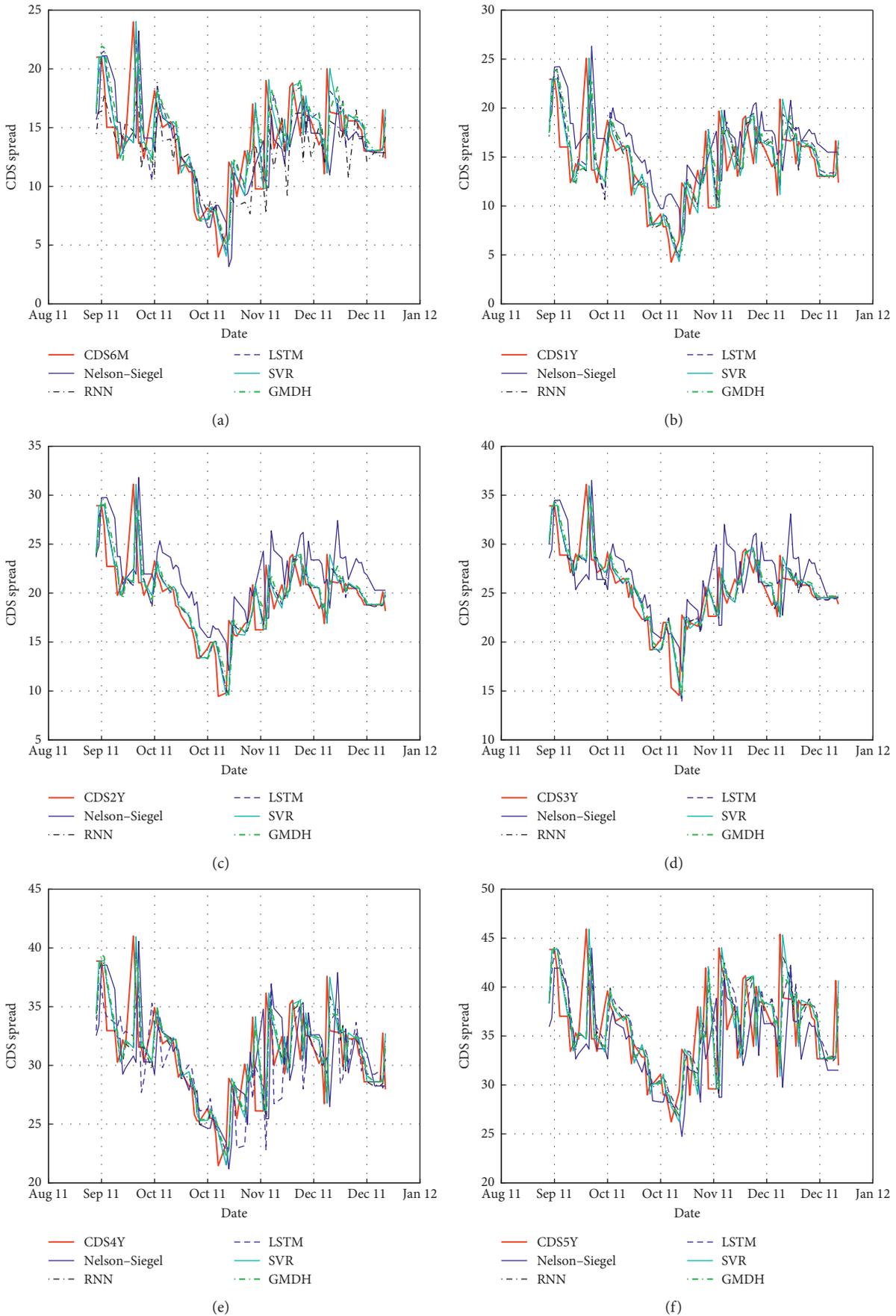


FIGURE 8: Predictions of each model and target CDS term structure from six months to five years maturity for test dataset 2 with the subperiod 1 training set (case 2).

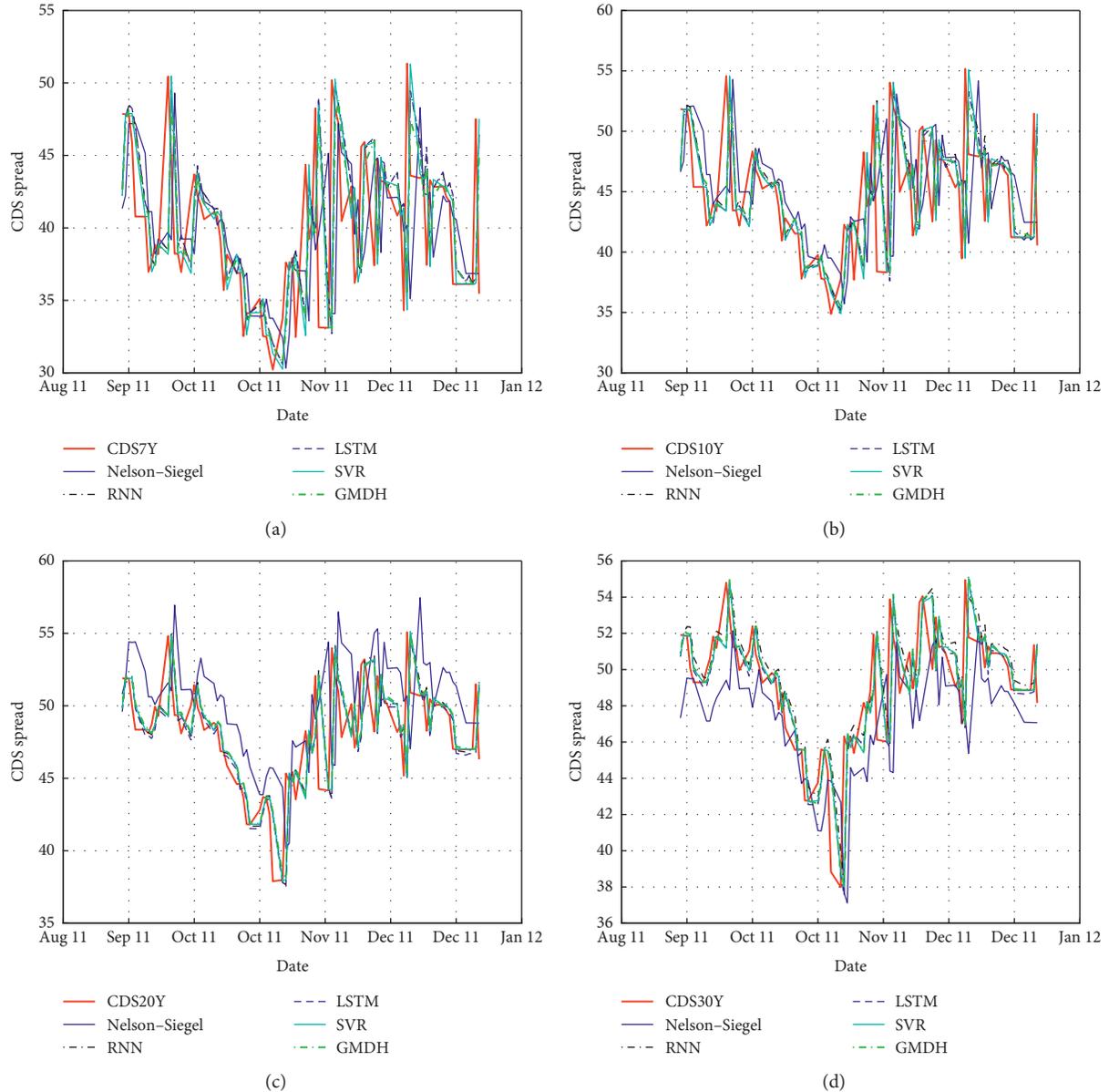


FIGURE 9: Predictions of each model and target CDS term structure from seven years to 30 years maturity for test dataset 2 with the subperiod 1 training set (case 2).

method has a much larger number of parameters than the model-driven method and a much slower implementation speed. However, it is acceptable to use a machine-learning algorithm without the need for prior knowledge, such as interest rate period structure, to predict CDS term structure more accurately. Second, we need to improve the existing Nelson-Siegel model. We showed that the machine-learning models outperform the Nelson-Siegel model for all three cases, which implies both that the machine-learning methodologies excel at this task and that there is a factor in the CDS term structure that the Nelson-Siegel model does not reflect. Nelson-Siegel still has room for improvement in its performance, especially in forecasting applications. Third, the performance of all models was inconsistent depending

on the data period. In the highly volatile data period (subperiod 1), all models were less predictable than in the less volatile data period (subperiod 2). In both approaches, the model performance is not stable when the data are highly volatile. Figure 1 shows that the CDS term structure from 2012 to 2019 seems regular but has some unpredictable points related to the financial turbulence from 2008 to 2011. This unusual volatility is one of the things that reduced the forecasting performance of all models. Therefore, it is necessary to consider a new approach that can achieve solid forecasting performance regardless of the volatility of the data.

Our findings can help investors and policymakers analyze the risk of companies or countries. The CDS spread is

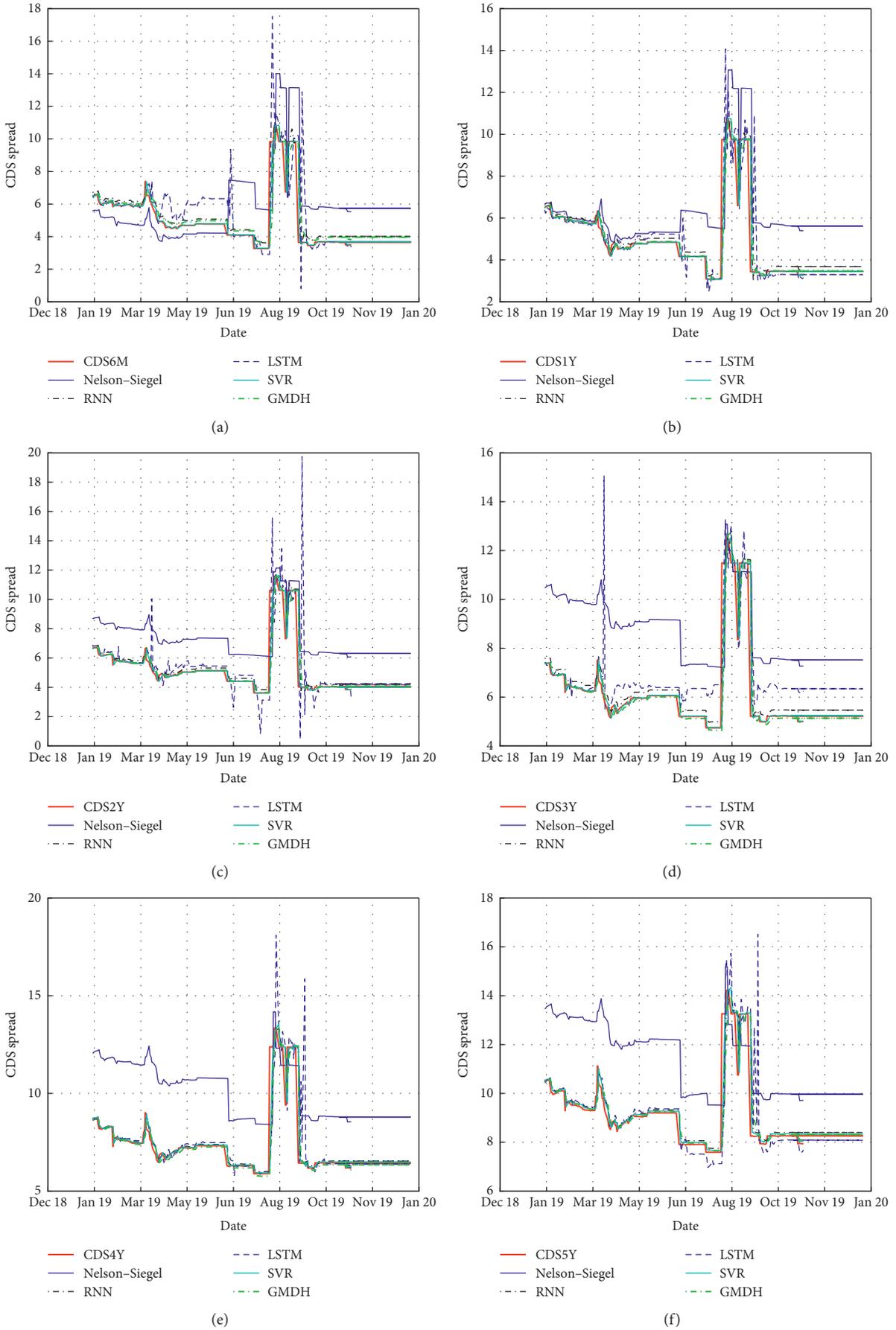


FIGURE 10: Predictions of each model and target CDS term structure from six months to five years maturity for test dataset 1 with the subperiod 2 training set (case 3).

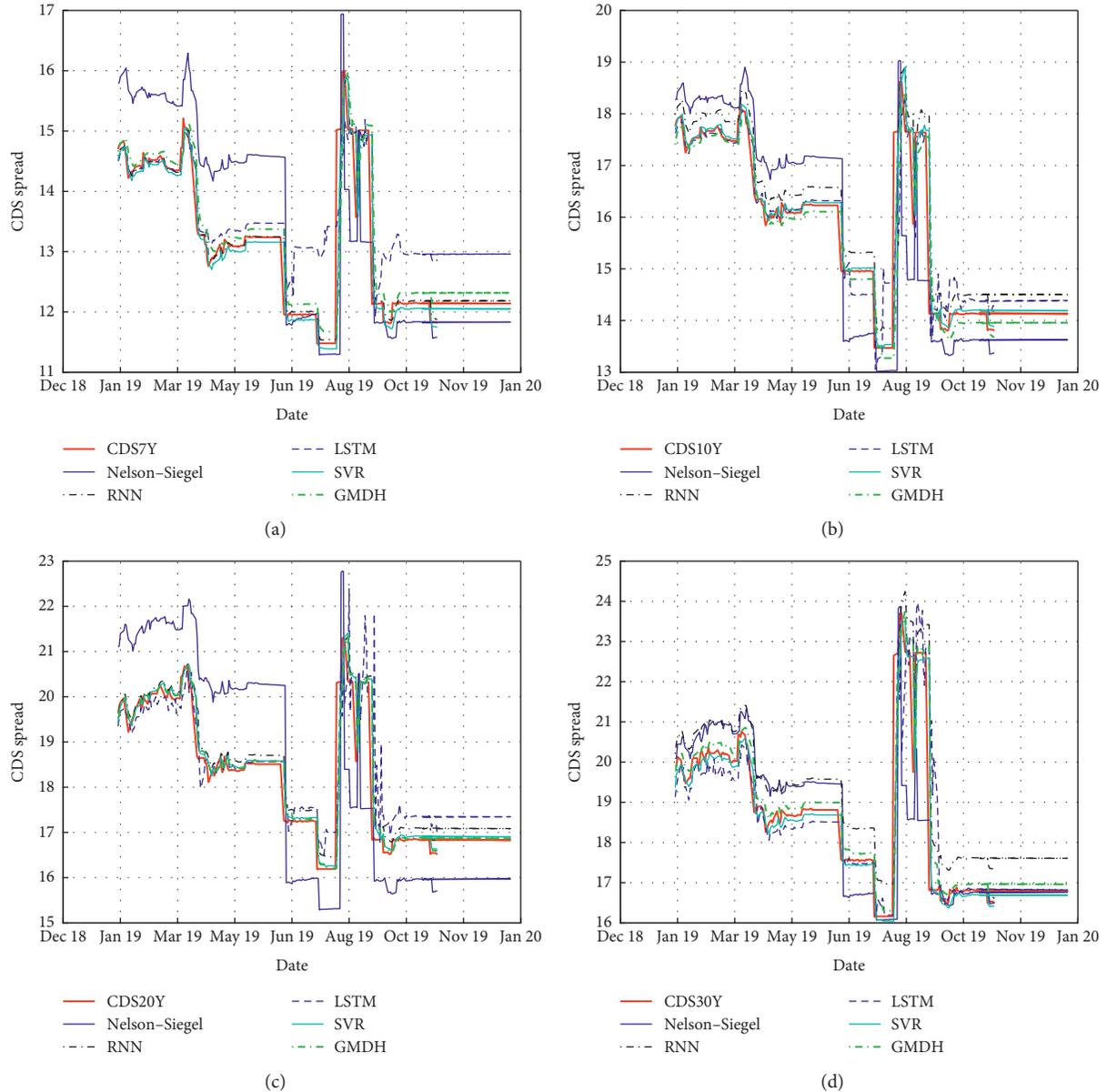


FIGURE 11: Predictions of each model and target CDS term structure from seven years to 30 years maturity for test dataset 1 with the subperiod 2 training set (case 3).

an index that represents the probability of credit default; thus, this study offers a measure to predict future risk. For instance, Zghal et al. [101] showed that CDS can function as a strong hedging mechanism against European stock market fluctuations, and Ratner and Chiu [19] confirmed the hedging and safe-haven characteristics of CDS against stock risks in the U.S. Researchers can also apply machine-learning models to forecast financial risk time-series data.

Future studies should apply this same experiment to datasets other than CDS data for comparing the forecasting performance of model-driven and data-driven methods, such as the implied volatility surface. The implied volatility surface is a fundamental concept for pricing various financial derivatives. Therefore, for a long time, many

researchers have been working on it, and various models have been developed [102–106]. Because it is a key part of the evaluation of financial derivatives, comparisons of performance between existing volatility models and data-driven models in predicting implied volatility should draw attention from academics and practitioners. GMDH showed the best predictive performance for the CDS term structure used in this study. It is now necessary to ensure that GMDH performs best for other term structures as well, such as for volatility term structures and yield curves, or other CDS contracts, for example, corporate CDS and CDS index. As a possible future study, extended Nelson–Siegel models can be used, such as regime-switching [107] and the Nelson–Siegel–Svensson model [108], to forecast CDS term

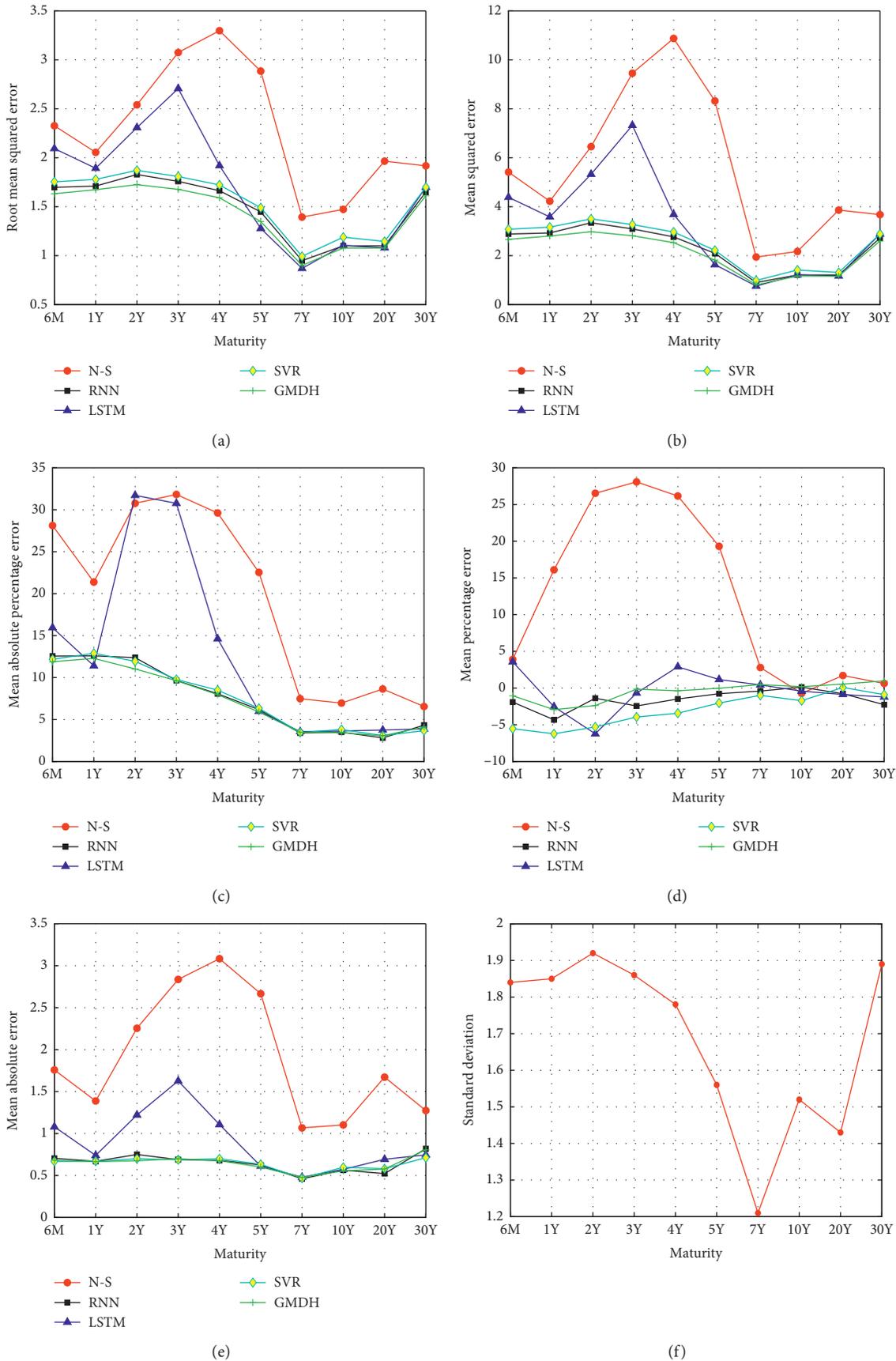


FIGURE 12: Error of each method for all maturities from Table 6 and standard deviations for all maturities (case 1): (a) RMSE (full period training set); (b) MSE (full period training set); (c) MAPE (full period training set); (d) MPE (full period training set); (e) MAE (full period training set); (f) test dataset 1.

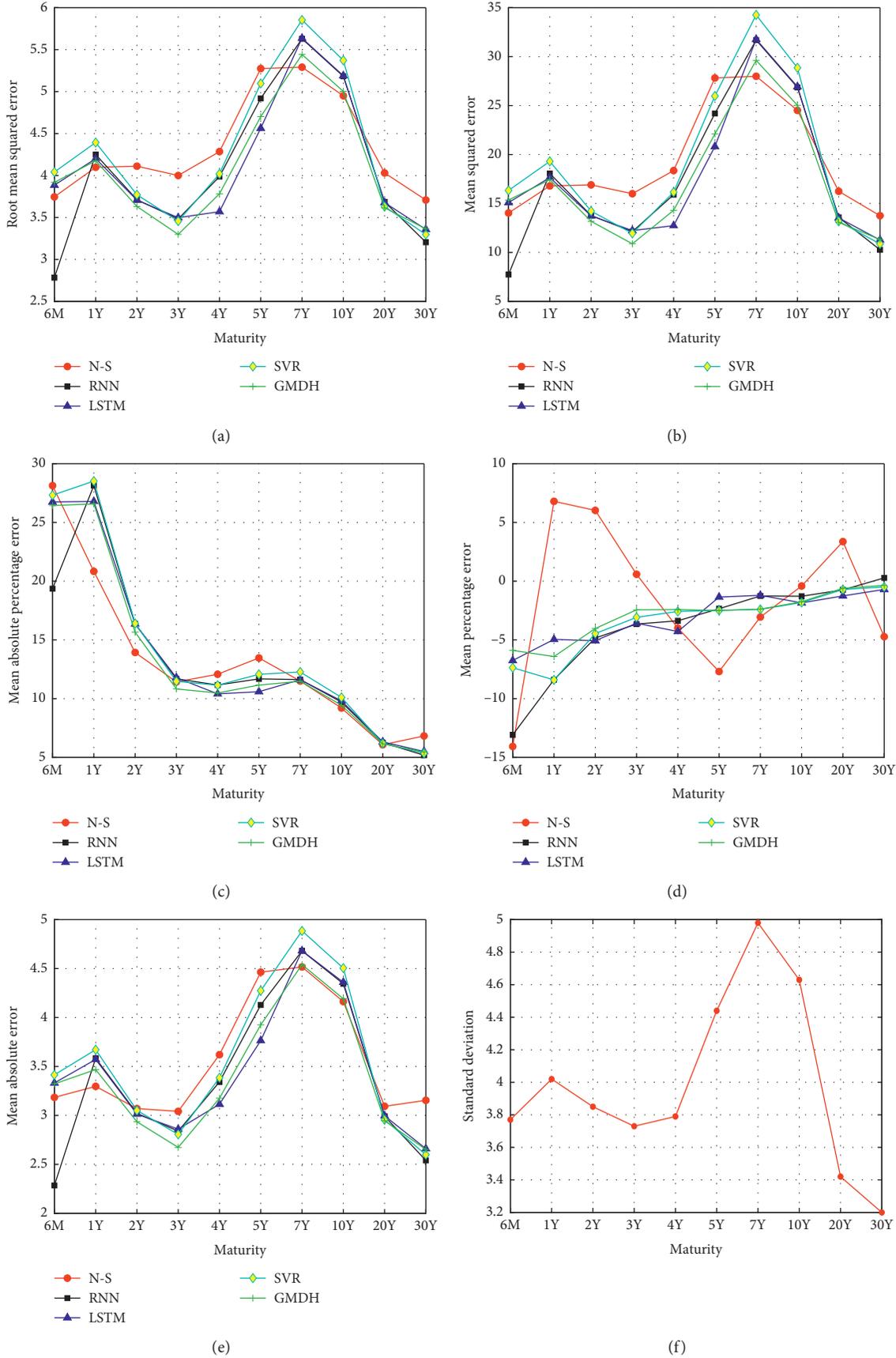


FIGURE 13: Error of each method for all maturities from Table 7 and standard deviations for all maturities (case 2): (a) RMSE (subperiod 1 training set); (b) MSE (subperiod 1 training set); (c) MAPE (subperiod 1 training set); (d) MPE (subperiod 1 training set); (e) MAE (subperiod 1 training set); (f) test dataset 2.

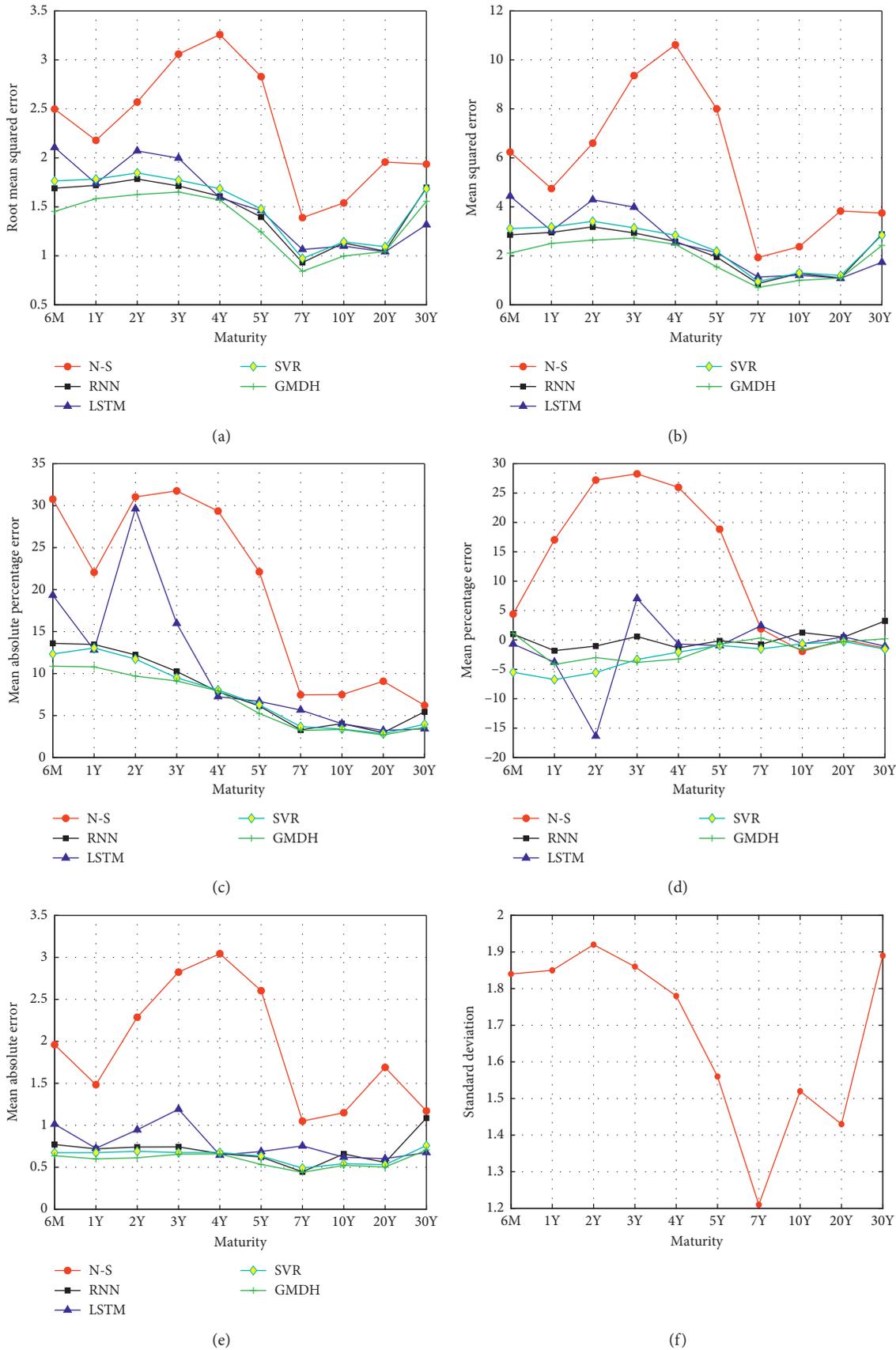


FIGURE 14: Error of each method for all maturities from Table 8 and standard deviations for all maturities (case 3): (a) RMSE (subperiod 2 training set); (b) MSE (subperiod 2 training set); (c) MAPE (subperiod 2 training set); (d) MPE (subperiod 2 training set); (e) MAE (subperiod 2 training set); (f) test dataset 1.

TABLE 6: Error statistics of each method for all maturities for test dataset 1 with the full period training set (case 1).

Type	Method	6M	1Y	2Y	3Y	4Y	5Y	7Y	10Y	20Y	30Y	Average
RMSE	N-S	2.33	2.06	2.54	3.07	3.30	2.88	1.39	1.47	1.97	1.92	2.29
	RNN	1.70	1.71	1.83	1.76	1.66	1.45	0.95	1.10	1.10	1.64	1.49
	LSTM	2.09	1.89	2.31	2.71	1.91	1.28	0.87	1.10	1.08	1.70	1.69
	SVR	1.75	1.78	1.87	1.81	1.72	1.49	0.99	1.19	1.15	1.70	1.55
	GMDH	1.63	1.67	1.73	1.68	1.59	1.35	0.90	1.08	1.08	1.61	1.43
MSE	N-S	5.41	4.22	6.45	9.45	10.88	8.32	1.94	2.17	3.86	3.68	5.64
	RNN	2.88	2.93	3.34	3.09	2.77	2.09	0.90	1.21	1.21	2.71	2.31
	LSTM	4.38	3.58	5.33	7.33	3.68	1.63	0.76	1.22	1.17	2.89	3.20
	SVR	3.08	3.17	3.50	3.27	2.97	2.22	0.99	1.41	1.31	2.89	2.48
	GMDH	2.67	2.80	2.98	2.81	2.53	1.82	0.81	1.16	1.16	2.60	2.13
MAPE (%)	N-S	37.30	32.47	46.24	48.08	43.09	29.50	7.89	6.82	8.95	6.51	26.69
	RNN	12.56	12.59	12.38	9.65	8.07	6.18	3.40	3.50	2.81	4.33	7.55
	LSTM	15.94	11.40	31.72	30.77	14.62	5.98	3.56	3.64	3.75	3.89	12.53
	SVR	12.18	12.89	11.93	9.78	8.50	6.34	3.47	3.81	3.09	3.67	7.57
	GMDH	11.88	12.28	11.03	9.63	7.96	5.90	3.48	3.44	3.06	4.11	7.28
MPE (%)	N-S	-19.03	-29.26	-43.62	-45.40	-40.38	-26.86	-3.79	-0.13	-2.76	-1.55	-21.28
	RNN	-1.91	-4.32	-1.39	-2.44	-1.49	-0.77	-0.41	0.13	-0.77	-2.26	-1.56
	LSTM	3.57	-2.54	-6.23	-0.68	2.90	1.16	0.43	-0.42	-0.87	-1.22	-0.39
	SVR	-5.53	-6.23	-5.29	-3.94	-3.43	-2.04	-0.99	-1.72	0.08	-0.90	-3.00
	GMDH	-1.05	-2.94	-2.38	-0.15	-0.37	-0.02	0.47	0.19	0.54	0.98	-0.47
MAE	N-S	1.76	1.39	2.26	2.84	3.08	2.67	1.07	1.10	1.67	1.27	1.91
	RNN	0.70	0.67	0.75	0.69	0.68	0.62	0.46	0.56	0.52	0.82	0.65
	LSTM	1.08	0.74	1.22	1.63	1.11	0.61	0.48	0.57	0.69	0.75	0.89
	SVR	0.67	0.67	0.70	0.69	0.70	0.63	0.47	0.60	0.58	0.71	0.64
	GMDH	0.68	0.66	0.68	0.70	0.68	0.60	0.48	0.56	0.58	0.81	0.64

N-S: Nelson-Siegel.

TABLE 7: Error statistics of each method for all maturities for test dataset 2 with the subperiod 1 training set (case 2).

Type	Method	6M	1Y	2Y	3Y	4Y	5Y	7Y	10Y	20Y	30Y	Average
RMSE	N-S	3.75	4.10	4.11	4.00	4.29	5.27	5.29	4.95	4.03	3.71	4.35
	RNN	2.78	4.25	3.71	3.48	3.99	4.92	5.63	5.18	3.69	3.20	4.08
	LSTM	3.89	4.20	3.71	3.50	3.57	4.56	5.64	5.19	3.68	3.36	4.13
	SVR	4.04	4.39	3.77	3.46	4.02	5.10	5.85	5.37	3.63	3.30	4.29
	GMDH	3.92	4.18	3.63	3.30	3.78	4.70	5.44	5.00	3.62	3.36	4.09
MSE	N-S	14.03	16.80	16.90	16.00	18.36	27.82	27.99	24.51	16.25	13.75	19.24
	RNN	7.74	18.07	13.78	12.13	15.90	24.20	31.69	26.86	13.59	10.27	17.42
	LSTM	15.10	17.65	13.74	12.24	12.74	20.82	31.77	26.93	13.53	11.27	17.58
	SVR	16.33	19.30	14.22	11.95	16.16	25.98	34.26	28.86	13.18	10.87	19.11
	GMDH	15.36	17.43	13.19	10.88	14.30	22.11	29.62	25.03	13.07	11.32	17.23
MAPE (%)	N-S	28.13	20.83	13.93	11.42	12.08	13.47	11.50	9.20	6.08	6.83	13.35
	RNN	19.36	28.14	16.36	11.69	11.16	11.68	11.62	9.71	6.26	5.19	13.11
	LSTM	26.75	26.79	16.36	11.79	10.41	10.59	11.61	9.82	6.32	5.51	13.60
	SVR	27.32	28.52	16.40	11.48	11.14	12.07	12.27	10.11	6.19	5.36	14.09
	GMDH	26.44	26.58	15.65	10.83	10.48	11.16	11.48	9.44	6.16	5.46	13.37
MPE (%)	N-S	-14.07	6.80	6.03	0.59	-3.99	-7.69	-3.05	-0.41	3.37	-4.72	-1.72
	RNN	-13.08	-8.39	-4.84	-3.65	-3.37	-2.35	-1.26	-1.28	-0.74	0.27	-3.87
	LSTM	-6.75	-4.96	-5.07	-3.58	-4.29	-1.35	-1.20	-1.85	-1.26	-0.71	-3.10
	SVR	-7.37	-8.40	-4.46	-3.08	-2.58	-2.49	-2.40	-1.82	-0.71	-0.49	-3.38
	GMDH	-5.89	-6.42	-4.03	-2.44	-2.42	-2.49	-2.38	-1.75	-0.60	-0.36	-2.88
MAE	N-S	3.18	3.30	3.07	3.04	3.62	4.46	4.52	4.16	3.09	3.15	3.56
	RNN	2.28	3.58	3.02	2.84	3.34	4.13	4.68	4.34	2.99	2.54	3.37
	LSTM	3.33	3.57	3.01	2.86	3.11	3.76	4.68	4.36	3.00	2.66	3.43
	SVR	3.41	3.67	3.05	2.81	3.39	4.27	4.88	4.50	2.96	2.60	3.55
	GMDH	3.32	3.47	2.93	2.67	3.18	3.92	4.54	4.19	2.95	2.65	3.38

TABLE 8: Error statistics of each method for all maturities for test dataset 1 with the subperiod 2 training set (case 3).

Type	Method	6M	1Y	2Y	3Y	4Y	5Y	7Y	10Y	20Y	30Y	Average
RMSE	N-S	2.50	2.18	2.57	3.06	3.26	2.83	1.39	1.54	1.96	1.94	2.32
	RNN	1.69	1.72	1.78	1.71	1.61	1.40	0.93	1.13	1.05	1.70	1.47
	LSTM	2.11	1.74	2.07	2.00	1.59	1.46	1.06	1.10	1.04	1.32	1.55
	SVR	1.76	1.78	1.84	1.77	1.69	1.48	0.97	1.14	1.09	1.68	1.52
	GMDH	1.45	1.58	1.63	1.65	1.57	1.24	0.84	1.00	1.04	1.56	1.36
MSE	N-S	6.24	4.75	6.60	9.36	10.62	8.00	1.93	2.37	3.83	3.75	5.74
	RNN	2.86	2.96	3.18	2.94	2.59	1.95	0.87	1.28	1.10	2.88	2.26
	LSTM	4.44	3.02	4.29	3.99	2.54	2.13	1.13	1.21	1.08	1.74	2.56
	SVR	3.11	3.18	3.41	3.14	2.84	2.19	0.95	1.31	1.20	2.84	2.42
	GMDH	2.11	2.51	2.64	2.73	2.47	1.55	0.71	1.00	1.09	2.42	1.92
MAPE (%)	N-S	30.77	22.06	31.04	31.76	29.36	22.11	7.47	7.49	9.08	6.22	19.74
	RNN	13.60	13.46	12.23	10.25	7.84	6.14	3.29	4.04	2.96	5.45	7.93
	LSTM	19.34	12.80	29.63	15.97	7.25	6.69	5.65	4.03	3.23	3.42	10.80
	SVR	12.32	13.05	11.73	9.49	8.05	6.28	3.68	3.39	2.84	3.97	7.48
	GMDH	10.88	10.80	9.69	9.11	7.94	5.25	3.22	3.34	2.69	3.60	6.65
MPE (%)	N-S	4.40	17.04	27.21	28.27	25.98	18.84	1.88	-1.95	0.03	-1.39	12.03
	RNN	1.00	-1.82	-1.04	0.59	-1.36	-0.15	-0.75	1.24	0.47	3.23	0.14
	LSTM	-0.68	-3.77	-16.35	7.06	-0.69	-0.96	2.42	-0.66	0.54	-1.07	-1.42
	SVR	-5.49	-6.74	-5.55	-3.33	-2.07	-0.92	-1.52	-0.66	-0.28	-1.56	-2.81
	GMDH	1.14	-4.17	-3.01	-3.82	-3.24	-0.73	0.34	-1.66	-0.29	0.19	-1.53
MAE	N-S	1.96	1.48	2.29	2.82	3.04	2.60	1.05	1.15	1.69	1.17	1.93
	RNN	0.77	0.72	0.74	0.74	0.66	0.62	0.45	0.66	0.56	1.09	0.70
	LSTM	1.01	0.73	0.95	1.19	0.64	0.69	0.75	0.62	0.60	0.68	0.79
	SVR	0.67	0.67	0.69	0.68	0.68	0.63	0.49	0.54	0.53	0.76	0.63
	GMDH	0.64	0.60	0.61	0.66	0.66	0.53	0.44	0.52	0.50	0.71	0.59

structure. Optimized through grid search for machine-learning algorithms, we expect to increase the forecasting power of the Nelson–Siegel model using extended models rather than by optimizing parameters for the Nelson–Siegel model.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors are grateful to the editor Baogui Xin for the valuable comments which helped to significantly improve this paper. This work was supported by the Gachon University Research Fund of 2018 (GCU-2018-0295) and by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (no. 2019R1G1A1010278).

References

- [1] Z. Li and V. Tam, “A comparative study of a recurrent neural network and support vector machine for predicting price movements of stocks of different volatilities,” in *Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, Paris, France, pp. 1–8, June 2017.
- [2] K. Chen, Y. Zhou, and F. Dai, “A LSTM-based method for stock returns prediction: a case study of China stock market,” in *Proceedings of the 2015 IEEE international conference on big data (big data)*, IEEE, Santa Clara, CA, USA, pp. 2823–2824, 2015.
- [3] T. Gao, Y. Chai, and Yi Liu, “Applying long short term memory neural networks for predicting stock closing price,” in *Proceedings of the 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, IEEE, Beijing, China, pp. 575–578, November 2017.
- [4] F. Shaw, F. Murphy, and F. O’Brien, “The forecasting efficiency of the dynamic Nelson-Siegel model on credit default swaps,” *Research in International Business and Finance*, vol. 30, pp. 348–368, 2014.
- [5] D. Avino and O. Nneji, “Are CDS spreads predictable? an analysis of linear and non-linear forecasting models,” *International Review of Financial Analysis*, vol. 34, pp. 262–274, 2014.
- [6] A. Sensoy, F. J. Fabozzi, and V. Eraslan, “Predictability dynamics of emerging sovereign CDS markets,” *Economics Letters*, vol. 161, pp. 5–9, 2017.
- [7] S. Neftci, A. Oliveira Santos, and Y. Lu, *Credit Default Swaps and Financial Crisis Prediction*, Technical Report, Working Paper Series, 2005.
- [8] J. Duyvesteyn and M. Martens, “Forecasting sovereign default risk with Merton’s model,” *The Journal of Fixed Income*, vol. 25, no. 2, pp. 58–71, 2015.
- [9] D. F. Gray, R. C. Merton, and Z. Bodie, “Contingent claims approach to measuring and managing sovereign credit risk,” *Journal of Investment Management*, vol. 5, no. 4, p. 5, 2007.

- [10] H. Blommestein, S. Eijffinger, and Z. Qian, "Regime-dependent determinants of euro area sovereign CDS spreads," *Journal of Financial Stability*, vol. 22, pp. 10–21, 2016.
- [11] J. Pan and K. J. Singleton, "Default and recovery implicit in the term structure of sovereign CDS spreads," *The Journal of Finance*, vol. 63, no. 5, pp. 2345–2384, 2008.
- [12] F. A. Longstaff, J. Pan, L. H. Pedersen, and K. J. Singleton, "How sovereign is sovereign credit risk?" *American Economic Journal: Macroeconomics*, vol. 3, no. 2, pp. 75–103, 2011.
- [13] E. C. Galariotis, P. Makrchoriti, and S. Spyrou, "Sovereign CDS spread determinants and spill-over effects during financial crisis: a panel var approach," *Journal of Financial Stability*, vol. 26, pp. 62–77, 2016.
- [14] S. Srivastava, H. Lin, I. M. Premachandra, and H. Roberts, "Global risk spillover and the predictability of sovereign CDS spread: International evidence," *International Review of Economics & Finance*, vol. 41, pp. 371–390, 2016.
- [15] S. H. Ho, "Long and short-runs determinants of the sovereign CDS spread in emerging countries," *Research in International Business and Finance*, vol. 36, pp. 579–590, 2016.
- [16] P. Augustin, "The term structure of CDS spreads and sovereign credit risk," *Journal of Monetary Economics*, vol. 96, pp. 53–76, 2018.
- [17] E. Bouri, S. J. H. Shahzad, N. Raza, and D. Roubaud, "Oil volatility and sovereign risk of BRICS," *Energy Economics*, vol. 70, pp. 258–269, 2018.
- [18] E. Bouri, N. Jalkh, and D. Roubaud, "Commodity volatility shocks and BRIC sovereign risk: a GARCH-quantile approach," *Resources Policy*, vol. 61, pp. 385–392, 2019.
- [19] M. Ratner and C.-C. Chiu, "Hedging stock sector risk with credit default swaps," *International Review of Financial Analysis*, vol. 30, pp. 18–25, 2013.
- [20] S. Hammoudeh, T. Liu, C.-L. Chang, and M. McAleer, "Risk spillovers in oil-related cds, stock and credit markets," *Energy Economics*, vol. 36, pp. 526–535, 2013.
- [21] S. J. Hussain Shahzad, E. Bouri, J. Arreola-Hernandez, D. Roubaud, and S. Bekiros, "Spillover across eurozone credit market sectors and determinants," *Applied Economics*, vol. 51, no. 59, pp. 6333–6349, 2019.
- [22] S. J. H. Shahzad, S. M. Nor, R. Ferrer, and S. Hammoudeh, "Asymmetric determinants of CDS spreads: U.S. industry-level evidence through the NARDL approach," *Economic Modelling*, vol. 60, pp. 211–230, 2017.
- [23] B. Han and Y. Zhou, "Understanding the term structure of credit default swap spreads," *Journal of Empirical Finance*, vol. 31, pp. 18–35, 2015.
- [24] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Computers & Operations Research*, vol. 32, no. 10, pp. 2513–2522, 2005.
- [25] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.
- [26] C.-J. Lu, T.-S. Lee, and C.-C. Chiu, "Financial time series forecasting using independent component analysis and support vector regression," *Decision Support Systems*, vol. 47, no. 2, pp. 115–125, 2009.
- [27] P.-F. Pai and C.-S. Lin, "A hybrid arima and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497–505, 2005.
- [28] N. Amanifard, N. Nariman-Zadeh, M. H. Farahani, and A. Khalkhali, "Modelling of multiple short-length-scale stall cells in an axial compressor using evolved gmdh neural networks," *Energy Conversion and Management*, vol. 49, no. 10, pp. 2588–2594, 2008.
- [29] M. Najafzadeh and G.-A. Barani, "Comparison of group method of data handling based genetic programming and back propagation systems to predict scour depth around bridge piers," *Scientia Iranica*, vol. 18, no. 6, pp. 1207–1213, 2011.
- [30] M. Witczak, J. Korbicz, M. Mrugalski, and R. J. Patton, "A GMDH neural network-based approach to robust fault diagnosis: application to the damadics benchmark problem," *Control Engineering Practice*, vol. 14, no. 6, pp. 671–683, 2006.
- [31] H. Yan and H. Ouyang, "Financial time series prediction based on deep learning," *Wireless Personal Communications*, vol. 102, no. 2, pp. 683–700, 2018.
- [32] Y. Baek and H. Y. Kim, "Modaugnet: a new forecasting framework for stock market index value with an overfitting prevention lstm module and a prediction lstm module," *Expert Systems with Applications*, vol. 113, pp. 457–480, 2018.
- [33] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on ceemdan and lstm," *Physica A: Statistical Mechanics and its Applications*, vol. 519, pp. 127–139, 2019.
- [34] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.
- [35] P. Thottakkara, T. Ozrazgat-Baslanti, B. B. Hupf et al., "Application of machine learning techniques to high-dimensional clinical data to forecast postoperative complications," *PLoS One*, vol. 11, no. 5, Article ID e0155705, 2016.
- [36] R. Motka, V. Parmar, B. Kumar, and A. R. Verma, "Diabetes mellitus forecast using different data mining techniques," in *Proceedings of the 2013 4th International Conference on Computer and Communication Technology (ICCCCT)*, IEEE, Allahabad, India, pp. 99–103, 2013.
- [37] N. Boyko, T. Sviridova, and N. Shakhovska, "Use of machine learning in the forecast of clinical consequences of cancer diseases," in *Proceedings of the 2018 7th Mediterranean Conference on Embedded Computing (MECO)*, IEEE, Budva, Montenegro, pp. 1–6, 2018.
- [38] P. J. Tighe, C. A. Harle, R. W. Hurley, H. Aytug, A. P. Boezaart, and R. B. Fillingim, "Teaching a machine to feel postoperative pain: combining high-dimensional clinical data with machine learning algorithms to forecast acute postoperative pain," *Pain Medicine*, vol. 16, no. 7, pp. 1386–1401, 2015.
- [39] S. Choi, Y. J. Kim, B. Simon, and D. Mavris, "Prediction of weather-induced airline delays based on machine learning algorithms," in *Proceedings of the 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, IEEE, Sacramento, CA, USA, pp. 1–6, September 2016.
- [40] S. E. Haupt and B. Kosovic, "Big data and machine learning for applied weather forecasts: forecasting solar power for utility operations," in *Proceedings of the 2015 IEEE Symposium Series on Computational Intelligence*, IEEE, Cape Town, South Africa, pp. 496–501, December 2015.
- [41] J. Rhee and J. Im, "Meteorological drought forecasting for ungauged areas based on machine learning: using long-range climate forecast and remote sensing data," *Agricultural and Forest Meteorology*, vol. 237–238, pp. 105–122, 2017.
- [42] S. C. James, Y. Zhang, and F. O'Donncha, "A machine learning framework to forecast wave conditions," *Coastal Engineering*, vol. 137, pp. 1–10, 2018.
- [43] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for

- large-scale transportation network speed prediction,” *Sensors*, vol. 17, no. 4, p. 818, 2017.
- [44] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: data-driven traffic forecasting,” 2017, <http://arxiv.org/abs/1707.01926>.
- [45] S. J. Farlow, *Self-Organizing Methods in Modeling: GMDH Type Algorithms*, Vol. 54, CRC Press, Boca Raton, FL, USA, 1984.
- [46] M. Najafzadeh and H. M. Azamathulla, “Group method of data handling to predict scour depth around bridge piers,” *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2107–2112, 2013.
- [47] N. Nariman-Zadeh, A. Darvizeh, and G. R. Ahmad-Zadeh, “Hybrid genetic design of GMDH-type neural networks using singular value decomposition for modelling and prediction of the explosive cutting process,” *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 217, no. 6, pp. 779–790, 2003.
- [48] S.-Y. Choi and C. Hong, “Relationship between uncertainty in the oil and stock markets before and after the shale gas revolution: evidence from the OVX, VIX, and VKOSPI volatility indices,” *PloS One*, vol. 15, no. 5, Article ID e0232508, 2020.
- [49] R. Y. M. Li, S. Fong, and K. W. S. Chong, “Forecasting the reits and stock indices: group method of data handling neural network approach,” *Pacific Rim Property Research Journal*, vol. 23, no. 2, pp. 123–160, 2017.
- [50] I. Pavlova, M. E. de Boyrie, and A. M. Parhizgari, “A dynamic spillover analysis of crude oil effects on the sovereign credit risk of exporting countries,” *The Quarterly Review of Economics and Finance*, vol. 68, pp. 10–22, 2018.
- [51] R. Ramezani, A. Peymanfar, and S. B. Ebrahimi, “An integrated framework of genetic network programming and multi-layer perceptron neural network for prediction of daily stock return: an application in tehran stock exchange market,” *Applied Soft Computing*, vol. 82, Article ID 105551, 2019.
- [52] S. Siami-Namini and A. S. Namin, *Forecasting Economics and Financial Time Series: ARIMA vs. LSTM*, <http://arxiv.org/abs/1803.06386>, 2018.
- [53] S. McNally, J. Roche, and S. Caton, “Predicting the price of bitcoin using machine learning,” in *Proceedings of the 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pp. 339–343, Cambridge, UK, 2018.
- [54] B. Cortez, B. Carrera, Y.-J. Kim, and J.-Y. Jung, “An architecture for emergency event prediction using LSTM recurrent neural networks,” *Expert Systems with Applications*, vol. 97, pp. 315–324, 2018.
- [55] J. Samarawickrama and T. G. I. Fernando, *A Recurrent Neural Network Approach in Predicting Daily Stock Prices: An Application to the Sri Lankan Stock Market*, IEEE, Peradeniya, Sri Lanka, 2017.
- [56] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, “Stock price prediction using LSTM, RNN and CNN-sliding window model,” in *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, Udipi, India, pp. 1643–1647, September 2017.
- [57] C. R. Nelson and A. F. Siegel, “Parsimonious modeling of yield curves,” *Journal of Business*, vol. 60, no. 4, pp. 473–489, 1987.
- [58] B. Guo, Q. Han, and B. Zhao, “The nelson-siegel model of the term structure of option implied volatility and volatility components,” *Journal of Futures Markets*, vol. 34, no. 8, pp. 788–806, 2014.
- [59] N. S. GrØnborg and A. Lunde, “Analyzing oil futures with a dynamic Nelson-Siegel model,” *Journal of Futures Markets*, vol. 36, no. 2, pp. 153–173, 2016.
- [60] J. West, “Long-dated agricultural futures price estimates using the seasonal Nelson-Siegel model,” *International Journal of Business and Management*, vol. 7, no. 3, pp. 78–93, 2012.
- [61] R.-R. Chen, X. Cheng, and L. Wu, “Dynamic interactions between interest-rate and credit risk: theory and evidence on the credit default swap term structure,” *Review of Finance*, vol. 17, no. 1, pp. 403–441, 2011.
- [62] M. Tsuruta, “Decomposing the term structures of local currency sovereign bond yields and sovereign credit default swap spreads,” *The North American Journal of Economics and Finance*, vol. 51, Article ID 101072, 2020.
- [63] S. R. Gunn, “Support vector machines for classification and regression,” *ISIS Technical Report*, vol. 14, no. 1, pp. 5–16, 1998.
- [64] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1994.
- [65] F. Pérez-cruz, J. A. Afonso-rodriguez, J. Giner et al., “Estimating GARCH models using support vector machines,” *Quantitative Finance*, vol. 3, no. 3, pp. 163–172, 2003.
- [66] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*, MIT Press, Cambridge, MA, USA, 2018.
- [67] L. Cao and F. E. H. Tay, “Financial forecasting using support vector machines,” *Neural Computing & Applications*, vol. 10, no. 2, pp. 184–192, 2001.
- [68] S. Hochreiter and J. Schmidhuber, “LSTM can solve hard long time lag problems,” in *Proceedings of the 10th Annual Conference on Neural Information Processing Systems (NIPS 1996)*, pp. 473–479, Denver, CO, USA, December 1996.
- [69] F. A. Gers and J. . Schmidhuber, “Recurrent nets that time and count,” in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN 2000)*, IEEE, Como, Italy, pp. 189–194, July 2000.
- [70] H.-G. Zimmermann, C. Tietz, and R. Grothmann, “Forecasting with recurrent neural networks: 12 tricks,” in *Neural Networks: Tricks of the Trade*, Springer, Berlin, Germany, 2012.
- [71] J. Chung, C. Gulcehre, K.H. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” 2014, <http://arxiv.org/abs/1412.3555>.
- [72] A. G. Ivakhnenko, “Polynomial theory of complex systems,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-1, no. 4, pp. 364–378, 1971.
- [73] M. Najafzadeh, G.-A. Barani, and H. M. Azamathulla, “GMDH to predict scour depth around a pier in cohesive soils,” *Applied Ocean Research*, vol. 40, pp. 35–41, 2013.
- [74] G. P. Liu and V. Kadirkamanathan, “Multiobjective criteria for neural network structure selection and identification of nonlinear systems using genetic algorithms,” *IEE Proceedings-Control Theory and Applications*, vol. 146, no. 5, pp. 373–382, 1999.
- [75] E. Sanchez, T. Shibata, and L. Asker Zadeh, *Genetic Algorithms and Fuzzy Logic Systems: Soft Computing Perspectives*, World Scientific, Singapore, 1997.
- [76] A. M. Ghaedi, M. M. Baneshi, A. Vafaei et al., “Comparison of multiple linear regression and group method of data handling models for predicting sunset yellow dye removal

- onto activated carbon from oak tree wood,” *Environmental Technology & Innovation*, vol. 11, pp. 262–275, 2018.
- [77] I. Ebtehaj, H. Bonakdari, F. Khoshbin, and H. Azimi, “Pareto genetic design of group method of data handling type neural network for prediction discharge coefficient in rectangular side orifices,” *Flow Measurement and Instrumentation*, vol. 41, pp. 67–74, 2015.
- [78] R. Shirmohammadi, B. Ghorbani, M. Hamed, M.-H. Hamed, and L. M. Romeo, “Optimization of mixed refrigerant systems in low temperature applications by means of group method of data handling (GMDH),” *Journal of Natural Gas Science and Engineering*, vol. 26, pp. 303–312, 2015.
- [79] M. Najafzadeh, G.-A. Barani, and H. M. Azamathulla, “Prediction of pipeline scour depth in clear-water and live-bed conditions using group method of data handling,” *Neural Computing and Applications*, vol. 24, no. 3-4, pp. 629–635, 2014.
- [80] A. Sakaguchi and T. Yamamoto, “A gmdh network using backpropagation and its application to a controller design,” in *Proceedings of the 2000 IEEE International Conference on Systems, Man and Cybernetics’ Cybernetics Evolving to Systems, Humans, Organizations, and their Complex Interactions*, IEEE, Nashville, TN, USA, pp. 2691–2696, 2000.
- [81] D. Srinivasan, “Energy demand prediction using GMDH networks,” *Neurocomputing*, vol. 72, no. 1–3, pp. 625–629, 2008.
- [82] C.-W. Hsu, C.-C. Chang, C.-J. Lin et al., *A Practical Guide to Support Vector Classification*, National Taiwan University, Taipei, Taiwan, 2003.
- [83] J. Bergstra and Y. Bengio, “Random search for hyperparameter optimization,” *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.
- [84] N. Schilling, M. Wistuba, L. Drumond, and L. Schmidt-Thieme, “Hyperparameter optimization with factorized multilayer perceptrons,” *Machine Learning and Knowledge Discovery in Databases*, Springer, Berlin, Germany, 2015.
- [85] F. Hutter, J. Lücke, and L. Schmidt-Thieme, “Beyond manual tuning of hyperparameters,” *KI-Künstliche Intelligenz*, vol. 29, no. 4, pp. 329–337, 2015.
- [86] J. Sun, C. Zheng, X. Li, and Y. Zhou, “Analysis of the distance between two classes for tuning SVM hyperparameters,” *IEEE Transactions on Neural Networks*, vol. 21, no. 2, pp. 305–318, 2010.
- [87] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, “Auto-weka: combined selection and hyperparameter optimization of classification algorithms,” in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 847–855, Chicago, IL, USA, August 2013.
- [88] N. K. Ahmed, A. F. Atiya, N. El Gayar, and H. El-Shishiny, “An empirical comparison of machine learning models for time series forecasting,” *Econometric Reviews*, vol. 29, no. 5-6, pp. 594–621, 2010.
- [89] A. Khosravi, L. Machado, and R. O. Nunes, “Time-series prediction of wind speed using machine learning algorithms: a case study osorio wind farm, Brazil,” *Applied Energy*, vol. 224, pp. 550–566, 2018.
- [90] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga, “Ensemble deep learning for regression and time series forecasting,” in *Proceedings of the 2014 IEEE symposium on computational intelligence in ensemble learning (CIEL)*, IEEE, Orlando, FL, USA, pp. 1–6, 2014.
- [91] A. V. Seliverstova, D. A. Pavlova, S. A. Tonoyan, and Y. E. Gapanyuk, “The time series forecasting of the company’s electric power consumption,” *Advances in Neural Computation, Machine Learning, and Cognitive Research II*, Springer, Berlin, Germany, 2018.
- [92] D. Solomatine, L. M. See, and R. J. Abrahart, “Data-driven modelling: concepts, approaches and experiences,” in *Practical Hydroinformatics*, Springer, Berlin, Germany, 2009.
- [93] S. Aras and İ. D. Kocakoç, “A new model selection strategy in time series forecasting with artificial neural networks: IHTS,” *Neurocomputing*, vol. 174, pp. 974–987, 2016.
- [94] M. C. Brace, J. Schmidt, and M. Hadlin, “Comparison of the forecasting accuracy of neural networks with other established techniques,” in *Proceedings of the First International Forum on Applications of Neural Networks to Power Systems*, IEEE, Seattle, WA, USA, pp. 31–35, 1991.
- [95] W. R. Foster, F. Collopy, and L. H. Ungar, “Neural network forecasting of short, noisy time series,” *Computers & Chemical Engineering*, vol. 16, no. 4, pp. 293–297, 1992.
- [96] Ü. Ç. Büyüksahin and Ş. Ertekin, “Improving forecasting accuracy of time series data using a new ARIMA-ANN hybrid method and empirical mode decomposition,” *Neurocomputing*, vol. 361, pp. 151–163, 2019.
- [97] A. Lapedes and R. Farber, “Nonlinear signal processing using neural networks: prediction and system modelling,” Technical Report, 1987.
- [98] M. C. Medeiros, A. Veiga, and C. E. Pedreira, “Modeling exchange rates: smooth transitions, neural networks, and linear models,” *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 755–764, 2001.
- [99] K. Rasouli, W. W. Hsieh, and A. J. Cannon, “Daily streamflow forecasting by machine learning methods with weather and climate inputs,” *Journal of Hydrology*, vol. 414-415, pp. 284–293, 2012.
- [100] J. C. Reboredo, J. M. Matias, and R. Garcia-Rubio, “Non-linearity in forecasting of high-frequency stock returns,” *Computational Economics*, vol. 40, no. 3, pp. 245–264, 2012.
- [101] R. Zghal, A. Ghorbel, and M. Triki, “Dynamic model for hedging of the european stock sector with credit default swaps and euro STOXX 50 volatility index futures,” *Borsa Istanbul Review*, vol. 18, no. 4, pp. 312–328, 2018.
- [102] S.-Y. Choi, J.-P. Fouque, and J.-H. Kim, “Option pricing under hybrid stochastic and local volatility,” *Quantitative Finance*, vol. 13, no. 8, pp. 1157–1165, 2013.
- [103] B. Dupire, “Pricing with a smile,” *Risk*, vol. 7, no. 1, pp. 18–20, 1994.
- [104] J. Gatheral and A. Jacquier, “Arbitrage-free SVI volatility surfaces,” *Quantitative Finance*, vol. 14, no. 1, pp. 59–71, 2014.
- [105] P. S. Hagan, Deep Kumar, A. S. Lesniewski, and D. E. Woodward, “Managing smile risk,” *The Best of Wilmott*, vol. 1, pp. 249–296, 2002.
- [106] S. L. Heston, “A closed-form solution for options with stochastic volatility with applications to bond and currency options,” *Review of Financial Studies*, vol. 6, no. 2, pp. 327–343, 1993.
- [107] J. Xiang and X. Zhu, “A regime-switching nelson-siegel term structure model and Interest rate forecasts,” *Journal of Financial Econometrics*, vol. 11, no. 3, pp. 522–555, 2013.
- [108] R. B. De Rezende and M. S. Ferreira, “Modeling and forecasting the yield curve by an extended nelson-siegel class of models: a quantile autoregression approach,” *Journal of Forecasting*, vol. 32, no. 2, pp. 111–123, 2013.