

Pemilihan Arsitektur Terbaik pada Model *Deep Learning* Melalui Pendekatan Desain Eksperimen untuk Peramalan Deret Waktu Nonlinier

NOVRI SUHERMI¹, SUHARTONO², I MADE GDE MERANGGI DANA³, DEDY DWI PRASTYO⁴

Departemen Statistika, Institut Teknologi Sepuluh Nopember

ABSTRAK

Penentuan arsitektur model *deep learning* yang tepat merupakan hal yang sangat esensial untuk mendapatkan hasil ramalan dengan tingkat kesalahan minimum. Arsitektur *deep learning* meliputi jumlah input dan variabel apa saja yang digunakan, jumlah *hidden layer*, jumlah neuron pada setiap *hidden layer*, dan fungsi aktivasi. Pada penelitian ini dilakukan studi simulasi pada salah satu model *deep learning*, yaitu *deep feedforward network*, dengan berbagai kombinasi arsitektur untuk mendapatkan arsitektur paling optimum. Data yang digunakan merupakan data bangkitan yang mengikuti model nonlinier *Exponential Smoothing Transition Auto-regressive* (ESTAR) sebanyak 1000 data, di mana 900 data digunakan sebagai data training dan 100 data digunakan sebagai data testing. Ukuran evaluasi model yang digunakan adalah *root mean square error of prediction* (RMSEP). Hasil empiris yang didapatkan di antaranya, pemilihan input yang tepat dapat meningkatkan akurasi peramalan, serta pemilihan fungsi aktivasi dan kedalaman arsitektur sangat diperlukan untuk mendapatkan hasil ramalan yang semakin optimum.

Kata Kunci: *deep learning*, *deep feedforward network*, arsitektur, desain eksperimen.

1. PENDAHULUAN

Deret waktu adalah pengamatan berurutan yang dikumpulkan dari waktu ke waktu dengan periode waktu yang sama, misalnya dalam jam, harian, mingguan, bulanan, dan tahunan (Hyndman dan Athanasopoulos, 2014). Berdasarkan pola data, deret waktu dibedakan menjadi dua, yaitu deret waktu linier dan deret waktu nonlinier. Salah satu metode yang sangat fleksibel dalam peramalan data deret waktu yang mengandung pola linier maupun nonlinier adalah *neural network*. Keunggulan menggunakan *neural network* adalah fleksibilitas model yang dapat dibentuk secara adaptif berdasarkan fitur yang disajikan dari data (Zhang dan Berardi, 1998).

Neural network mengadopsi cara kerja syaraf biologis yang terdiri dari neuron sebagai pemrosesan input, kemudian nilai input yang ada akan dijumlahkan oleh suatu fungsi perambatan (*summing function*), dan memberikan output berdasarkan bobot yang ada. Model *neural network* banyak diaplikasikan dalam berbagai bidang peramalan, seperti harga saham (Haefke dan Helmenstein (1995), Gencay dan Stengos (1998), Qi dan Maddala (1999)), nilai tukar (Kuan dan Liu (1995), Franses dan Van Griensven (1998), Gencay (1999)), dan suku bunga (Swanson dan White, 1995).

Suatu *neural network* memerlukan arsitektur yang tepat untuk mendapatkan ramalan yang menghasilkan kesalahan minimum. Arsitektur di dalam *neural network* meliputi jumlah input dan variabel apa saja yang digunakan, jumlah *hidden layer*, jumlah neuron pada setiap *hidden layer*, dan fungsi aktivasi. Oleh karena itu, diperlukan suatu rancangan percobaan atau desain eksperimen untuk menentukan arsitektur terbaik pada *neural network*. Pada penelitian ini dilakukan studi simulasi dengan berbagai macam kombinasi arsitektur pada *neural network*.

Beberapa penelitian juga menunjukkan bahwa model *deep neural network*, atau disebut juga *deep learning*, yaitu *neural network* yang memiliki arsitektur *hidden layer* yang lebih satu, telah

banyak diterapkan untuk kasus peramalan dan menghasilkan ramalan dengan akurasi tinggi (Chen dkk., (2017), Liu dan Chen (2017), Qin dkk., (2017), Qiu dkk., (2017), Voyant dkk., (2017), Zhao dkk., (2017)). Kemudian, pada penelitian ini akan dibandingkan model *deep feedforward network* dengan model *single hidden layer feedforward network*. Tujuan lainnya dari penelitian ini adalah untuk mengetahui efek dari input dan variabel yang digunakan, efek dari jumlah *hidden layer*, efek dari jumlah neuron pada setiap *hidden layer*, dan efek dari fungsi aktivasi untuk meminimumkan nilai *root mean square error of prediction* (RMSEP). Metode yang digunakan adalah desain eksperimen dari data simulasi model nonlinier *Exponential Smoothing Transition Auto-regressive* (ESTAR) yang dimodelkan dengan berbagai macam arsitektur *neural network*.

2. MODEL NEURAL NETWORK

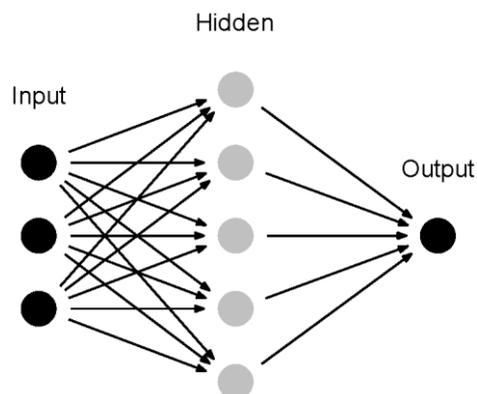
ARTIFICIAL NEURAL NETWORK

Artificial Neural Network (ANN) atau jaringan saraf tiruan merupakan suatu sistem proses informasi yang memiliki karakteristik performa tertentu dalam jaringan saraf biologis, dalam penelitian ini *neural network* dipandang sebagai suatu objek matematika atau secara khusus sebagai suatu model *time series*, dengan asumsi sebagai berikut.

- i. Pemrosesan informasi terjadi pada banyak elemen sederhana yang disebut *neuron*.
- ii. Sinyal dilewatkan di antara neuron di atas *connection links*.
- iii. Masing-masing *connection link* memiliki bobot yang dikalikan dengan sinyal yang ditransmisi.
- iv. Masing-masing *neuron* menggunakan fungsi aktivasi (biasanya nonlinier) pada net input (jumlahan sinyal input terboboti) untuk menentukan sinyal output.

Suatu *neural network* digolongkan berdasarkan pola koneksi di antara neuron yang disebut juga sebagai arsitektur, metode dalam menentukan bobot dari koneksi yang disebut sebagai *training*, *learning*, atau algoritma, dan fungsi aktivasi (Fausett, 1994). Ada banyak jenis arsitektur *neural network*, beberapa diantaranya yaitu *feedforward network*, *recurrent network*, dan *radial basis function network*. Model *feedforward neural network* (FFNN) merupakan salah satu model yang sering digunakan untuk tujuan peramalan.

Pada FFNN, proses dimulai dari input yang diterima oleh neuron, dimana neuron-neuron ini dikelompokkan dalam *layer*. Informasi yang diterima dari *input layer* dilanjutkan ke *layer* dalam FFNN secara berurutan hingga mencapai *output layer*. *Layer* yang berbeda diantara input dan output disebut *hidden layer*. Input yang digunakan dalam *neural network* adalah lag dari observasi sebelumnya dan output merupakan hasil peramalan. Contoh arsitektur model FFNN dengan terdiri dari p input, satu *hidden layer* yang terdiri dari m unit dan dihubungkan ke output, disajikan pada Gambar 1 (El-Telbany, 2014).



Gambar 1. Contoh Arsitektur FFNN

Berikut ini adalah persamaan dalam pencarian output dari arsitektur FFNN dengan satu hidden layer seperti pada Gambar 1 (Taylor, 2000).

$$f(\mathbf{x}_t, \mathbf{v}, \mathbf{w}) = g_2 \left\{ \sum_{j=1}^m v_j g_1 \left[\sum_{i=1}^n w_{ji} x_{it} \right] \right\}, \tag{1}$$

dengan \mathbf{w} merupakan pembobot yang menghubungkan *input layer* dengan *hidden layer*, \mathbf{v} merupakan pembobot yang menghubungkan *hidden layer* dengan *output layer*, $g_1(\cdot)$ dan $g_2(\cdot)$ merupakan fungsi aktivasi, sedangkan w_{ji} dan v_j merupakan bobot yang akan diestimasi. Pada umumnya, fungsi aktivasi yang banyak digunakan adalah *sigmoid* dan *hyperbolic tangent* (tanh), dirumuskan pada persamaan (2) dan (3).

$$g(x) = \frac{1}{1 + e^{-x}}, \tag{2}$$

$$g(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{3}$$

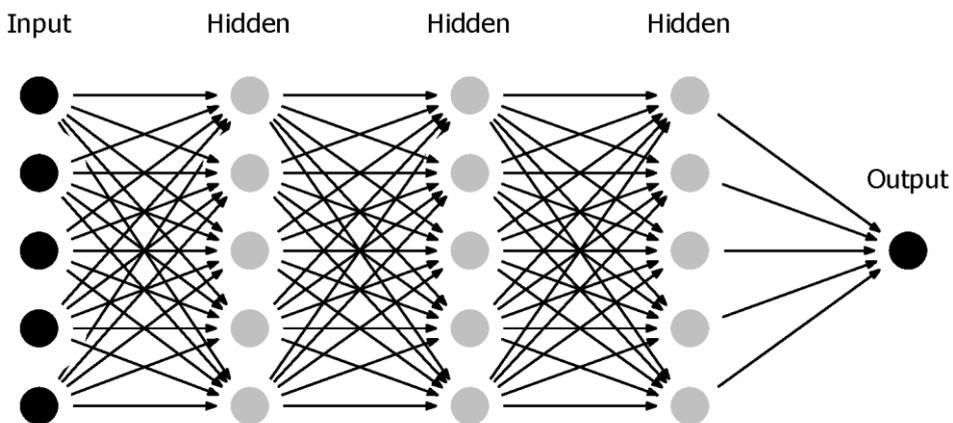
Fungsi sigmoid memiliki *range* pada interval (0,1), sementara fungsi tanh memiliki range pada interval (-1,1).

DEEP FEEDFORWARD NETWORK

Deep feedforward network atau *deep neural network* (DNN) adalah model FFNN dengan jumlah *hidden layer* lebih dari satu. DNN merupakan salah satu model dasar *deep learning* (Goodfellow dkk., 2016). DNN bertujuan untuk mengaproksimasi suatu fungsi f . DNN akan mencari fungsi aproksimasi terbaik dengan cara mempelajari (*learning*) parameter θ dari pemetaan fungsi $y = f(x; \theta)$, di mana arsitektur dari DNN ditunjukkan pada gambar 2. Di dalam konteks model deret waktu, hubungan antara output Y_t dan input-input lag Y_t yaitu $Y_{t-1}, Y_{t-2}, \dots, Y_{t-p}$ di dalam model DNN dengan 3 *hidden layer* dijelaskan pada persamaan berikut.

$$Y_t = \sum_{i=1}^s \alpha_i g \left(\sum_{j=1}^r \beta_{ij} g \left(\sum_{k=1}^q \gamma_{jk} g \left(\sum_{l=1}^p \theta_{kl} Y_{t-l} \right) \right) \right) + \varepsilon_t, \tag{4}$$

di mana ε_t adalah komponen error, $\alpha_i (i = 1, 2, \dots, s)$, $\beta_{ij} (i = 1, 2, \dots, s; j = 1, 2, \dots, r)$, $\gamma_{jk} (j = 1, 2, \dots, r; k = 1, 2, \dots, q)$, dan $\theta_{kl} (k = 1, 2, \dots, q; l = 1, 2, \dots, p)$ adalah parameter model yang disebut juga sebagai *connection weights*, p menyatakan jumlah input, q, r, s, berturut-turut menyatakan jumlah jumlah *node* di dalam *hidden layer* 1, 2, dan 3. Fungsi $g(\cdot)$ menyatakan fungsi aktivasi.



Gambar 2. Contoh Arsitektur DNN

3. DESAIN EKSPERIMEN

Eksperimen diawali dengan membangkitkan data dari suatu model deret waktu nonlinier, yaitu *Exponential Smoothing Transition Auto-regressive* (ESTAR) (Haefke dan Helmenstein, 1995), yang didefinisikan pada persamaan berikut (Terasvirta, 1994).

$$Y_t = 6.5Y_{t-3} \exp(-0.25Y_{t-3}) + a_t \quad (5)$$

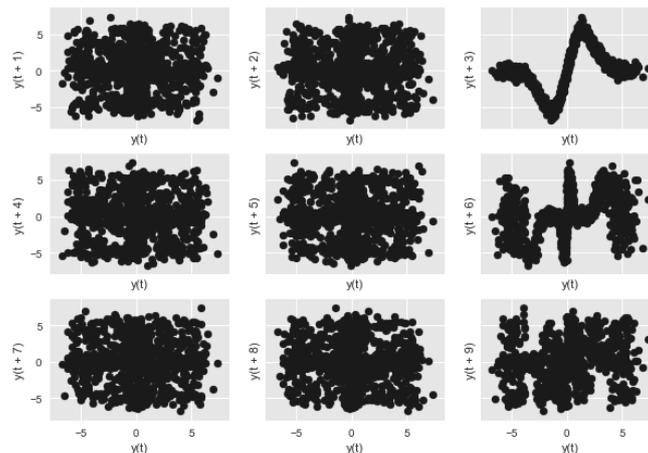
Beberapa faktor yang digunakan di dalam eksperimen ini di antaranya pemilihan input, jumlah input, jumlah *hidden layer*, jumlah *node* di dalam *hidden layer*, dan fungsi aktivasi. Input yang digunakan adalah variabel lag dari data deret waktu, yaitu lag 1, lag 2, lag 3, lag 4, dan lag 5. Jumlah *hidden layer* yang digunakan adalah sebanyak 1, 2, dan 3. Jumlah *node* di dalam *hidden layer* yang digunakan dibatasi dari 1 sampai 5 *node*. Fungsi aktivasi yang digunakan adalah fungsi *sigmoid* dan fungsi *tanh*. Setiap kombinasi model dilakukan replikasi sebanyak 3 kali.

Data dibagi menjadi dua bagian, yaitu data *training* dan *testing*. Model dilatih menggunakan data *training* yang selanjutnya digunakan untuk memprediksi data *testing*. Teknik peramalan yang digunakan adalah peramalan satu tahap. Keabikan ramalan diukur menggunakan *root mean square error of prediction* (RMSEP). RMSEP dihitung menggunakan persamaan berikut (De Gooijer dan Hyndman, 2006).

$$\text{RMSEP} = \sqrt{\frac{1}{L} \sum_{l=1}^L (Y_{n+l} - \hat{Y}_n(l))^2} \quad (6)$$

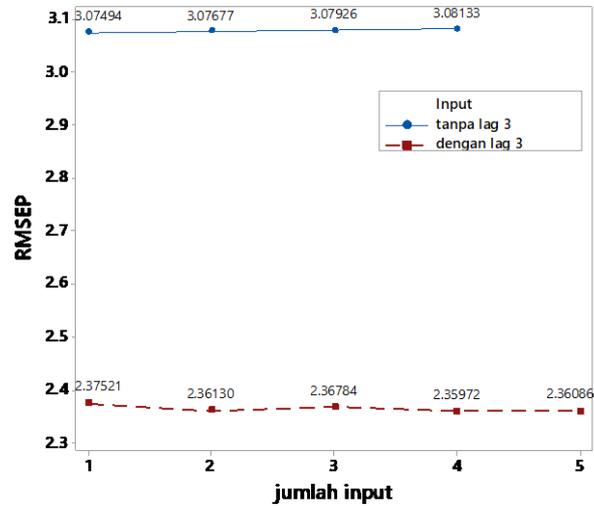
4. ANALISIS DAN PEMBAHASAN

Analisis awal dilakukan dengan menggambarkan secara visual hubungan antara data dengan variabel-variabel lagnya. Berdasarkan gambar 3, dapat dilihat bahwa plot pada lag 3 menunjukkan adanya pola nonlinier yang jelas, sehingga dapat disimpulkan bahwa data bangkitan sudah sesuai dengan model ESTAR (Haefke dan Helmenstein, 1995).



Gambar 3. Lag Plot Data Simulasi Model ESTAR (Haefke dan Helmenstein, 1995)

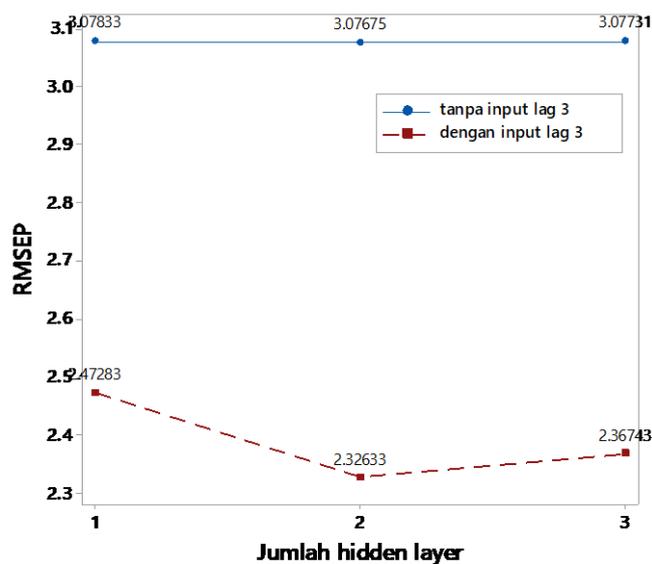
Gambar 4 menunjukkan efek dari penambahan jumlah input pada model *neural network* dengan input melibatkan lag 3 dan tanpa melibatkan lag 3 berdasarkan rata-rata RMSEP data *testing*. Penambahan jumlah input terlihat tidak memberikan efek yang signifikan terhadap hasil ramalan. Sementara pelibatan lag 3 sebagai input model memberikan efek yang sangat drastis pada penurunan nilai RMSEP. Ini mengindikasikan pemilihan input memiliki peran yang sangat penting terhadap hasil ramalan, dibandingkan jumlah input. Pada saat model menggunakan input yang salah, maka hasil ramalannya akan buruk sebanyak apapun input yang digunakan.



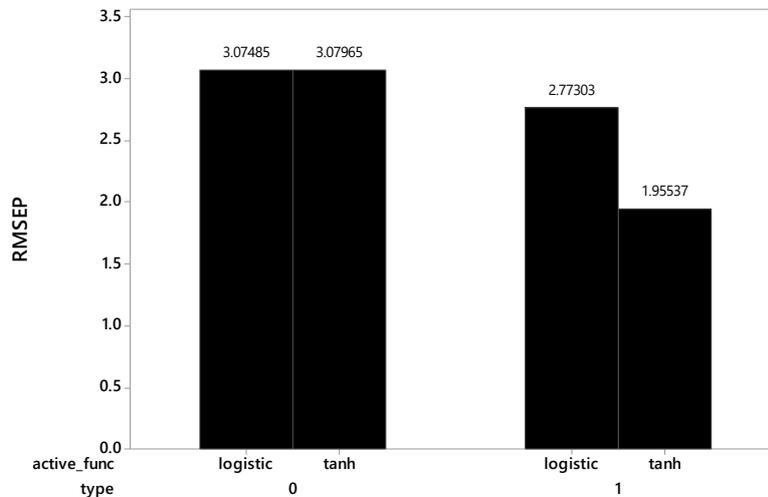
Gambar 4. Efek Jumlah Input pada Model FFNN dan DNN

Kemudian dilanjutkan dengan analisis efek jumlah hidden layer. Gambar 5 menunjukkan bahwa model DNN yang memiliki arsitektur lebih dalam berhasil meningkatkan akurasi ramalan. Hal ini ditunjukkan dengan dengan penurunan rata-rata RMSEP dengan penambahan *hidden layer*. Model dengan 2 *hidden layer* dan input melibatkan lag 3 memberikan rata-rata RMSEP terendah dibandingkan model lain. Sementara model yang tidak melibatkan input lag 3 sama sekali tidak terdapat perubahan yang signifikan dari efek penambahan jumlah *hidden layer*.

Selanjutnya, dilakukan perbandingan pada efek fungsi aktivasi terhadap nilai RMSEP yang ditunjukkan pada gambar 6. Angka 0 pada gambar 6 mengacu pada model-model tanpa input lag 3. Sementara angka 1 mengacu pada model-model yang mengandung input lag 3. Berdasarkan gambar 6, dapat dilihat bahwa model dengan fungsi aktivasi tanh berhasil mengungguli fungsi *sigmoid* dengan perbedaan rata-rata RMSEP yang sangat signifikan. Maka, pemilihan fungsi aktivasi yang tepat menjadi sangat diperlukan supaya hasil ramalan menjadikan semakin optimum. Namun, jika input model yang digunakan tidak tepat, pemilihan fungsi aktivasi tidak akan memberikan pengaruh apapun terhadap hasil ramalan.



Gambar 5. Efek Jumlah *Hidden Layer* pada Model FFNN dan DNN



Gambar 6. Efek Fungsi Aktivasi pada Model FFNN dan DNN

Berdasarkan hasil eksperimen secara keseluruhan, didapatkan 10 model terbaik yang ditampilkan pada tabel 1 yang diperingkat berdasarkan nilai RMSEP. Semua model yang termasuk dalam 10 besar akurasi terbaik mengandung input lag 3 pada arsitekturnya. Model dengan hanya 1 input, yaitu lag 3, ternyata berhasil mengungguli model-model yang lain, dengan menggunakan fungsi aktivasi tanh dan 3 *hidden layer*. Kemudian, peringkat ini juga menunjukkan bahwa model DNN dengan fungsi aktivasi tanh dan jumlah *hidden layer* sebanyak 3 memberikan hasil ramalan yang paling akurat dibandingkan model-model lainnya.

Tabel 1. Peringkat 10 Model Terbaik Berdasarkan Nilai RMSEP

No	Input	Jumlah Hidden Layer	Jumlah Node pada Hidden Layer	Fungsi Aktivasi	RMSEP
1	Y_{t-3}	3	(4, 5, 5)	tanh	0,685
2	$Y_{t-1}, Y_{t-2}, Y_{t-3}, Y_{t-4}$	3	(4, 4, 5)	tanh	0,694
3	Y_{t-3}	3	(5, 2, 5)	tanh	0,698
4	Y_{t-3}	3	(5, 5, 5)	tanh	0,698
5	Y_{t-3}	3	(4, 5, 5)	tanh	0,716
7	Y_{t-3}	3	(5, 2, 5)	tanh	0,718
8	$Y_{t-1}, Y_{t-2}, Y_{t-3}, Y_{t-4}$	3	(4, 4, 5)	tanh	0,719
9	$Y_{t-1}, Y_{t-2}, Y_{t-3}$	3	(4, 5, 5)	tanh	0,726
10	$Y_{t-1}, Y_{t-2}, Y_{t-3}, Y_{t-5}$	3	(4, 4, 5)	tanh	0,734

5. KESIMPULAN

Eksperimen menghasilkan beberapa penemuan yang menarik untuk mendapatkan arsitektur model *deep learning* terbaik. Pada kasus ini jumlah input tidak memberikan efek yang signifikan terhadap hasil ramalan. Namun, pemilihan input yang tepat adalah faktor utama penentu kebaikan model. Selanjutnya, terdapat beberapa faktor lain yang bisa dioptimalkan untuk meningkatkan akurasi ramalan, di antaranya jumlah *hidden layer* dan fungsi aktivasi.

Penelitian ini telah menunjukkan bahwa model *deep learning* merupakan model yang menjanjikan untuk diteliti lebih jauh berkat keberhasilannya mengungguli model klasik *neural network* seperti FFNN. Oleh karena itu, penelitian ini dapat dilanjutkan dengan mengeksplorasi

arsitektur model *deep learning* lainnya seperti *recurrent neural network* (RNN) dan *long short term memory network* (LSTM).

DAFTAR PUSTAKA

- Hyndman, R.J. and Athanasopoulos, G., 2014. *Forecasting: principles and practice*. OTexts.
- Zhang, G.P. and Berardi, V.L., 1998. An investigation of neural networks in thyroid function diagnosis. *Health Care Management Science*, 1(1), pp.29-37.
- Haefke, C. and Helmenstein, C., 1995, April. A neural network model to exploit the econometric properties of austrian IPOs. In *Computational Intelligence for Financial Engineering, 1995.*, Proceedings of the IEEE/IAFE 1995 (pp. 128-135). IEEE.
- Gencay, A.Y.R. and Stengos, T., 1998. Moving average rules, volume and the predictability of security returns with feedforward networks. *J. Forecast*, 17, pp.401-414.
- Qi, M. and Maddala, G.S., 1999. Economic factors and the stock market: a new perspective. *Journal of Forecasting*, 18(3), pp.151-166.
- Kuan, C.M. and Liu, T., 1995. Forecasting exchange rates using feedforward and recurrent neural networks. *Journal of applied econometrics*, 10(4), pp.347-364.
- Franses, P.H. and Van Griensven, K., 1998. Forecasting exchange rates using neural networks for technical trading rules. *Studies in Nonlinear Dynamics & Econometrics*, 2(4).
- Gencay, R., 1999. Linear, non-linear and essential foreign exchange rate prediction with simple technical trading rules. *Journal of International Economics*, 47(1), pp.91-107.
- Swanson, N.R. and White, H., 1995. A model-selection approach to assessing the information in the term structure using linear models and artificial neural networks. *Journal of Business & Economic Statistics*, 13(3), pp.265-275.
- Chen, Y., He, K. and Tso, G.K., 2017. Forecasting Crude Oil Prices: a Deep Learning based Model. *Procedia Computer Science*, 122, pp.300-307.
- Liu, L. and Chen, R.C., 2017. A novel passenger flow prediction model using deep learning methods. *Transportation Research Part C: Emerging Technologies*, 84, pp.74-91.
- Qin, M., Li, Z. and Du, Z., 2017. Red tide time series forecasting by combining ARIMA and deep belief network. *Knowledge-Based Systems*, 125, pp.39-52.
- Qiu, X., Ren, Y., Suganthan, P.N. and Amaratunga, G.A., 2017. Empirical mode decomposition based ensemble deep learning for load demand time series forecasting. *Applied Soft Computing*, 54, pp.246-255.
- Voyant, C., Notton, G., Kalogirou, S., Nivet, M.L., Paoli, C., Motte, F. and Fouilloy, A., 2017. Machine learning methods for solar radiation forecasting: A review. *Renewable Energy*, 105, pp.569-582.
- Zhao, Y., Li, J. and Yu, L., 2017. A deep learning ensemble approach for crude oil price forecasting. *Energy Economics*, 66, pp.9-16.
- Fausett, L., 1994. *Fundamentals of neural networks: architectures, algorithms, and applications*. Prentice-Hall, Inc..
- Taylor, J.W., 2000. A quantile regression neural network approach to estimating the conditional density of multiperiod returns. *Journal of Forecasting*, 19(4), pp.299-311.
- El-Telbany, M.E., 2014, December. What quantile regression neural networks tell us about prediction of drug activities. In *Computer Engineering Conference (ICENCO), 2014 10th International* (pp. 76-80). IEEE.
- Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y., 2016. *Deep learning* (Vol. 1). Cambridge: MIT press.
- Teräsvirta, T., 1994. Specification, estimation, and evaluation of smooth transition autoregressive models. *Journal of the American Statistical Association*, 89(425), pp.208-218.
- De Gooijer, J.G. and Hyndman, R.J., 2006. 25 years of time series forecasting. *International Journal of Forecasting*, 22(3), pp.443-473.