

BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Pada penelitian sebelumnya, Ju-Jie Wang, Jian-Zhou Wang, Zhe-George Zhang, dan Shu-Po Guo, mencoba untuk melakukan peramalan dengan menggunakan data historis bulanan dari indeks pasar saham penutupan pada Shenzhen Integrated Index (SZII) dari Bulan Januari tahun 1993 hingga Bulan Desember tahun 2010 serta data historis bulanan indeks penutupan dari Dow Jones Industrial Average Index (DJIAI) mulai Bulan January tahun 1991 hingga Bulan Desember tahun 2010 menggunakan Matlab sebagai *software* pengolahan data. Dengan menggunakan model seperti *Exponential Smoothing Model* (ESM), *Autoregressive Integrated Moving Average Model* (ARIMA), *Proposed Hybrid Model* (PHM) yang ditentukan oleh *Genetic Algorithm* (GA), *Equal Weight Hybrid* (EWH), dan *Random Walk Model* (RWM). Indeks saham SZII berhasil di prediksi menggunakan model prediksi regresi dengan semua hasil model diuji dengan uji galat RMSE dan dievaluasi dengan *Robustness Evaluation* dengan rasio 70, untuk model ESM dengan galat 4391,3 kemudian jumlah galat 4046,8 untuk model ARIMA, lalu 4537,7 untuk galat pada model BPNN, pada model EWH bernilai 4181,6, selanjutnya diikuti oleh model PHM dan RWM dengan galat secara berurut 3983,2 dan 4880,4. Begitu juga dengan indeks DJIAI dengan model, uji galat dan rasio *Robustness Evaluation* yang sama, model ESM bernilai 6784,4 kemudian jumlah galat 6547,2 untuk model ARIMA, lalu 6125,4 untuk galat pada model BPNN, pada model EWH dengan galat 5424,6, selanjutnya model PHM dengan 4734,3 dan RWM dengan galat 6991,5 (Wang, et al., 2012).

Metode berbeda digunakan oleh Susanti, dimana Susanti menggunakan metode RNN jenis *Backpropagation Through Time*, dengan parameter pengujian dibatasi berdasarkan jumlah data pertahun dan peramalan perhari dari unit saham spesifik seperti ASII, ISAT, BUMI, dan BNII. Pada penelitian ini, rata rata Mean Square Error terendah dari 4 data saham yang diteliti oleh Susanti adalah sebesar

0,005 pada saham ASII dengan parameter inputan data indeks harian selama setahun (Susanti, et al., 2011).

Lain halnya dengan Baohua Wang, Hejiao Huang, dan Xiaolong Wang yang melakukan peramalan dengan menggunakan 2 model sebagai pembandingan model yang diajukan oleh peneliti, dengan model pertama menggunakan ARIMA dengan EVIEW system, model kedua menggunakan SVM, dan model yang diajukan peneliti merupakan hybrid dari ARIMA dan SVM. Dengan data penelitian diambil dari indeks 6 perusahaan dengan jangka waktu berbeda, yakni Merchants Bank dari Bulan Januari tahun 2002 hingga Bulan Desember tahun 2009, VANKE dari Bulan Januari tahun 2002 hingga Bulan Desember tahun 2009, SUNING APPLIANCE dari Bulan Januari tahun 2004 hingga Bulan Maret tahun 2010, GE dari Bulan Januari tahun 1994 hingga Bulan Desember tahun 2010, Johnson&Johnson dari Bulan Januari tahun 1994 hingga Bulan Desember tahun 2010, dan McDonald's dari Bulan Januari tahun 1994 hingga Bulan Desember tahun 2010. Pada penelitian tersebut secara keseluruhan model yang diajukan Wang, dkk berhasil memperoleh galat RMSE terendah sebesar 1,34 dibandingkan kedua model bandingan dengan model pertama sebesar 1,51 dan model kedua sebesar 1,63 (Baohua, et al., 2012).

2.2 Dasar Teori

2.2.1 Data Preprocessing

Preprocessing seringkali merujuk pada penambahan, pengurangan atau transformasi pada data. Akan tetapi perubahan pada data tersebut bisa menjadi perhatian penting, karena preprocessing bisa menciptakan maupun menghancurkan kemampuan model dalam melakukan prediksi. Model yang berbeda juga memiliki tingkat sensitifitas terhadap bentuk data, maka preprocessing pada data juga bergantung dari tipe model yang akan digunakan. Sebagai contoh model seperti pohon tidak peka pada karakteristik data, sedangkan linear regression peka terhadap karakteristik data (Kuhn & Johnson, 2013).

Sebenarnya, tujuan dari preprocessing data adalah untuk mengubah data real world ke dalam suatu format data yang dapat dibaca oleh komputer. Data yang diambil secara langsung dari database atau didapatkan secara real world merupakan data yang masih mentah, belum dilakukan transformasi, pembersihan atau perubahan. Sehingga kemungkinan data akan mengandung kesalahan seperti kesalahan input, data yang hilang, dan banyak lainnya (García, et al., 2014).

2.2.2 Penambangan Data

Secara analogi, sebutan yang tepat untuk penambangan data adalah penambangan pengetahuan. Akan tetapi banyak orang seringkali menyamakan penambangan data sebagai penemuan pengetahuan, sementara lainnya memandang penambangan data sebagai bagian dari penemuan pengetahuan. Proses pada penemuan pengetahuan terdiri atas beberapa langkah seperti pembersihan data, integrasi data, seleksi data, transformasi data, penambangan data, evaluasi pola dan presentasi pengetahuan.

Fungsi dari penambangan data adalah proses penemuan pola menarik dan pengetahuan dari suatu data berjumlah besar atau *Big Data*. Data tersebut bisa berasal dari berbagai macam sumber seperti databases, data warehouse, web, dan sebagainya. Secara umum, penambangan data bisa diaplikasikan pada berbagai macam data selama data tersebut memiliki arti pada target yang diinginkan. Ada beberapa fungsionalitas penambangan data termasuk karakterisasi dan diskriminasi, frequent pattern, asosiasi, korelasi, clustering, klasifikasi dan regresi. Fungsionalitas tersebut berguna untuk menemukan pola yang diinginkan secara spesifik. Akan tetapi walaupun penambangan data berpotensi untuk dapat menemukan ribuan bahkan jutaan pola atau aturan, tidak semua pola tersebut menarik. Hanya sebagian kecil saja yang dapat menarik perhatian. Pola yang dapat menarik perhatian adalah pola yang

mudah dipahami manusia, bisa divalidasi menggunakan data lain dan memiliki potensi yang berguna (Han, et al., 2011).

2.2.3 Supervised Learning

Prediksi sering dianggap sebagai supervised learning, karena metode supervised mencoba untuk menemukan relasi antara atribut masukan dan atribut target. Hubungan tersebut diwakili pada suatu struktur yang disebut model. Pada umumnya, model mendeskripsikan dan menjelaskan pengalaman, dimana hal tersebut tersembunyi di dalam data dan bisa digunakan untuk memprediksi nilai dari suatu target selama memiliki masukan yang diketahui. Supervised learning sering diterapkan pada berbagai macam bidang seperti keuangan, kesehatan, pembangunan dan sebagainya.

Terdapat 2 permasalahan klasik yang termasuk ke dalam supervised learning, yaitu klasifikasi dan regresi. Pada penelitian ini, nilai yang ingin diprediksi berupa angka pada suatu interval tertentu oleh karena itu model akan diselesaikan dengan model prediksi regresi. Namun, permasalahan regresi memberikan tingkat kesulitan, kebutuhan komputasi dan kompleksitas model yang lebih besar dibandingkan permasalahan klasifikasi (García, et al., 2014).

2.2.4 Penambangan Data Temporal

Merupakan teknik penambangan data yang memiliki suatu satuan waktu yang disebut data temporal atau data deret waktu. Hal penting pada penambangan data temporal adalah pencarian suatu kemiripan antara suatu data deret waktu atau deret kejadian. Sebagai contoh kesamaan fluktuasi harga saham dengan data yang lama bisa berguna untuk prediksi harga saham di masa mendatang. Akan tetapi hasil komputasi akan kesamaan data bergantung pada representasi waktu data yang digunakan (Mitsa, 2010).

Antunes berpendapat bahwa tujuan utama dari penambahan data temporal adalah untuk menemukan relasi antara sequence dan subsequence pada suatu event. Dikarenakan data yang bersifat temporal, maka pendekatan yang perlu dilakukan pun juga berbeda. Seringkali, aplikasi yang berhubungan dengan data temporal digunakan untuk melakukan diagnosa dan prediksi perilaku di masa mendatang (Antunes & Oliveir, 2001).

2.2.5 Deep Learning

Deep Learning adalah cabang dari bidang machine learning yang dibuat menggunakan algoritma yang meniru struktur dan fungsi dari otak manusia. Deep learning berkaitan dengan pembuatan jaringan saraf tiruan yang terdiri dari banyak lapisan yang ditumpuk di atas satu sama lain. Lapisan-lapisan ini terinspirasi oleh lapisan otak manusia dan interaksi neuron antara yang melewati informasi di antara lapisan-lapisan ini. Dalam jaringan syaraf tiruan, beberapa lapisan ini memungkinkan neuron menyimpan informasi dan meneruskannya ke lapisan untuk mencapai versi yang lebih abstrak dari data asli. Versi abstrak dari data ini membantu untuk memahami data asli dengan melatih algoritma pada mesin untuk mengenali data tersebut (Nahal, et al., 2017).

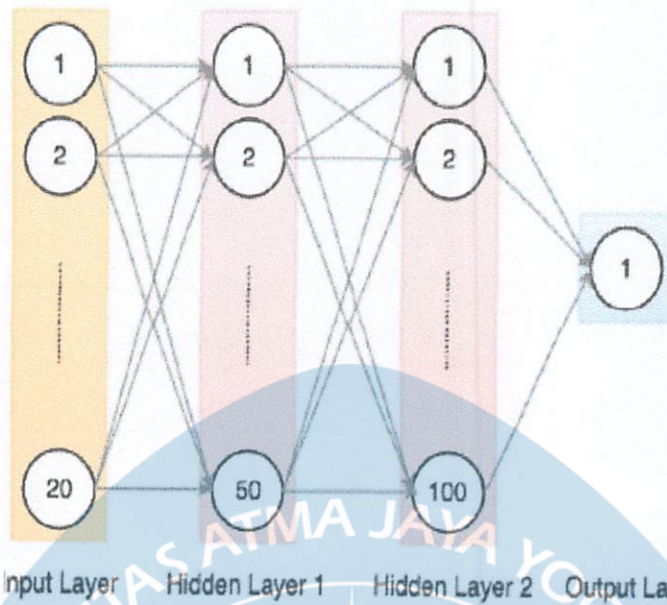
2.2.6 Recurrent Neural Network

Recurrent Neural Network (RNN) adalah generalisasi alami dari jaringan saraf model *feedforward*. Given a sequence of inputs (x_1, \dots, x_T), jenis standar dari RNN menghitung urutan keluaran (y_1, \dots, y_T) dengan melakukan iterasi dengan persamaan berikut (Sutskever, et al., 2014):

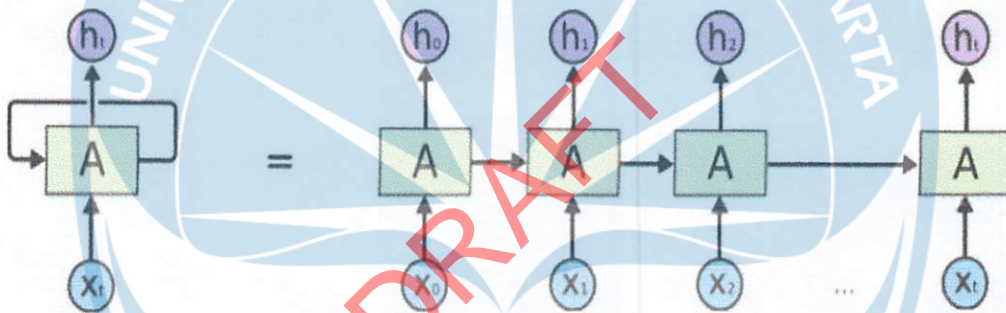
$$h_t = \text{sigm}(W^{hx}x_t + W^{hh}h_{t-1}) \quad (2.1)$$

$$y_t = W^{yh}h_t$$

RNN memproses urutan input satu elemen pada satu waktu, mempertahankan sekuens tersebut dalam *Hidden Units* atau *Hidden Layer* dalam bentuk *State Vector* yang secara implisit berisi informasi tentang rekam semua elemen sebelumnya dari urutan sebelumnya. Seperti sebagian kuadrat terkecil, hasilnya dimodelkan oleh seperangkat variabel tak teramati (disebut *Hidden Variable* atau *Hidden Units*). Unit-unit tersembunyi ini adalah kombinasi linear dari prediktor asli, tetapi RNN tidak diestimasi secara hirarkis. Ketika kita mempertimbangkan output dari unit tersembunyi pada langkah waktu diskrit yang berbeda seolah-olah mereka adalah hasil dari neuron yang berbeda dalam jaringan *multilayer* yang dalam.

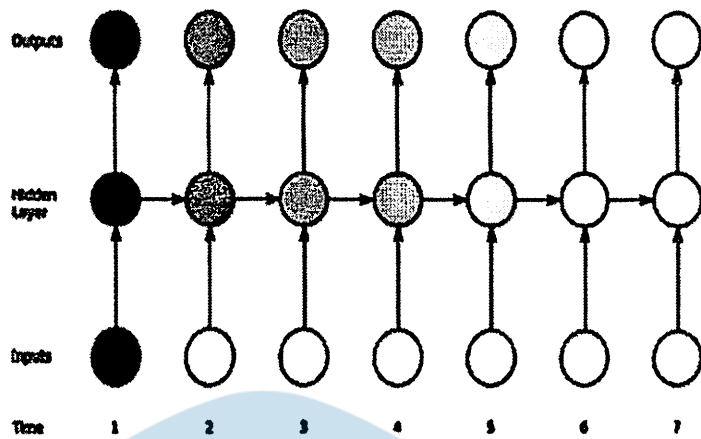


Gambar 2.1. Arsitektur Recurrent Neural Network



Gambar 2.2. Alur Recurrent Neural Network

RNN memperlakukan setiap indeks berurutan dan secara rekursif menyusun setiap indeks dengan memori sebelumnya hingga seluruh sekuens indeks telah diturunkan. Namun, dalam praktiknya, terdapat masalah terkait pelatihan model yang salah satunya *Vanishing Gradient* (Hochreiter & Schmidhuber, 1997) (Bengio, Simard, & Frasconi, 1994).



Gambar 2.3. Vanishing Gradient

Ada dua faktor yang mempengaruhi besarnya gradien - bobot dan fungsi aktivasi (atau lebih tepatnya, turunannya) yang dilalui oleh gradien. Jika salah satu dari faktor ini lebih kecil dari 1, maka gradien dapat lenyap pada waktunya; jika lebih besar dari 1, gradien akan meledak, gradien yang meledak dapat menghasilkan jaringan tidak stabil yang tidak dapat belajar dari data pelatihan dan paling baik jaringan yang tidak dapat belajar melalui sekuens data input yang panjang (Pascanu, Mikolov, & Bengio, 2013).

2.2.7 Long Short Term Memory

LSTM diajukan pada tahun 1997 oleh Sepp Hochreiter dan Jürgen Schmidhuber kemudian dikembangkan kembali pada tahun 2000 oleh tim Felix Gers. *Long Short Term Memory* (LSTM) merupakan jenis *Recurrent Neural Network* (RNN) yang melakukan proses menggantikan sel RNN normal dan menggunakan *input gate*, *forget gate*, dan *output gate*.

Definisi variabel input, output dan forget gate :

$$f_t = \text{sigm}(W_f[h_{t-1}, x_t] + b_t) \quad (2.2)$$

$$i_t = \text{sigm}(W_i[h_{t-1}, x_t] + b_i)$$

$$o_t = \text{sigm}(W_o[h_{t-1}, x_t] + b_o)$$

Definisi variabel RNN Cell

$$c_t = \tanh (W_c[h_{t-1}, x_t] + b_c) \quad (2.3)$$

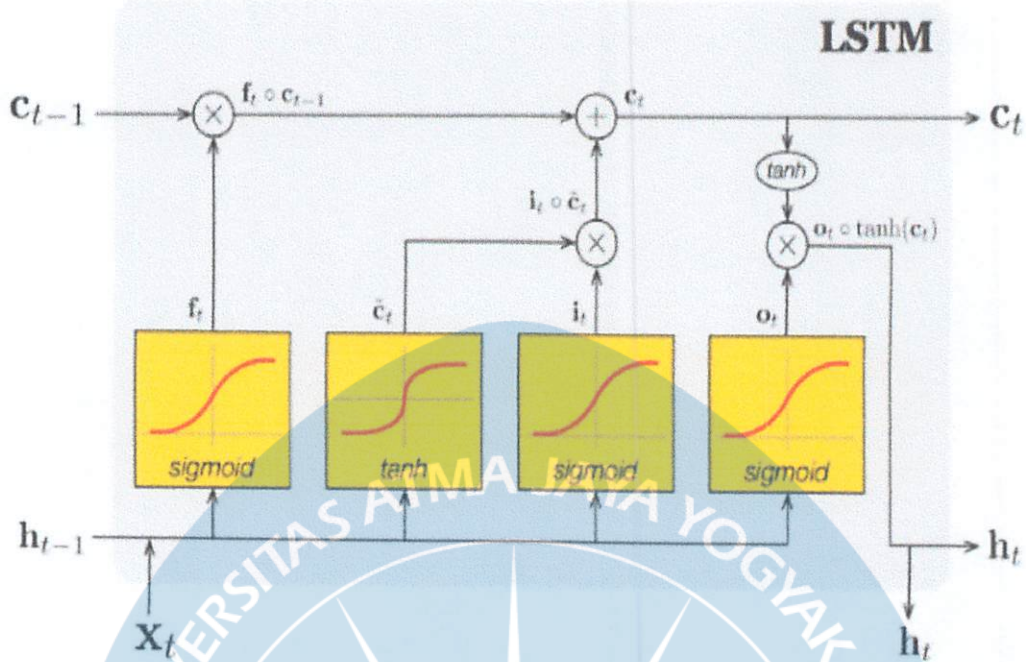
Definisi Variabel Hidden State

$$c_t = f_t o \ c_{t-1} + i_t o \ c_t \quad (2.4)$$

$$h_t = o_t o \ \tanh (c_t)$$

Pada sebuah sekuens dari panjang variabel dengan $x = (x_1, x_2, \dots, x_n)$ dengan secara bertahap menambahkan konten baru pada sebuah slot memory, dengan gerbang yang mengontrol sejauh mana konten baru harus diingat, konten lama harus dihapus, dan konten yang saat ini harus diproses.

Ide LSTM dasar sangat sederhana. Beberapa unit disebut *Constant Error Carousels* (CECs). Setiap CEC digunakan sebagai fungsi aktivasi f , yakni fungsi identitas, dan memiliki koneksi ke dirinya sendiri dengan bobot tetap 1,0. Karena turunan konstan dari 1,0 maka kesalahan backpropagated melalui CEC tidak dapat menghilang atau meledak namun tetap atau tidak berubah. CECs terhubung ke beberapa unit adaptif nonlinier (beberapa dengan fungsi aktivasi multiplikatif) yang diperlukan untuk mempelajari perilaku nonlinear. Perubahan berat unit-unit ini sering menguntungkan dari sinyal kesalahan yang disebar jauh ke masa lalu melalui CECs. CECs adalah alasan utama mengapa jaring LSTM dapat belajar untuk menemukan pentingnya (dan menghafal) peristiwa yang terjadi ribuan langkah waktu diskrit lalu. Berikut penjabaran persamaannya:



Gambar 2.4. Alur Gerbang Recurrent Neural Network (Hochreiter & Schmidhuber, 1997)

Model LSTM dapat mengetahui apa yang harus dilupakan, apa yang harus diingat. Jadi ketika sebuah masukan baru masuk, model itu terlebih dahulu melupakan informasi jangka panjang yang diputuskan tidak lagi dibutuhkan. Kemudian ia belajar bagian mana dari input baru yang layak digunakan, dan menyimpannya ke dalam memori jangka panjangnya.

2.2.8 Adaptive Moment Estimation

ADAM merupakan metode untuk optimalisasi stokastik yang efisien yang hanya membutuhkan gradien orde pertama dengan kebutuhan memori yang minimal. Metode ini menghitung tingkat pembelajaran adaptif individu untuk parameter yang berbeda dari perkiraan momen pertama dan kedua dari gradien; nama Adam berasal dari adaptive moment estimation (Kingma & Ba, 2014). Metode dirancang untuk menggabungkan keuntungan dari dua metode yang populer yakni: AdaGrad (Duchi, Hazan, & Singer, 2011), yang bekerja dengan baik dengan gradien yang tersebar, dan RMSProp (Tieleman & Hinton, 2012) yang bekerja dengan baik di pengaturan data non-stasioner.

AdaGrad (Duchi, et al., 2011), yang bekerja dengan baik dengan gradien yang tersebar, dan RMSProp (Tieleman & Hinton, 2012) yang bekerja dengan baik di pengaturan data non-stasioner.

Beberapa keuntungan ADAM adalah bahwa besarnya pembaruan parameter tidak berubah terhadap pengecilan ulang gradien, ukurannya perkiraan dibatasi oleh parameter kompleks, dan tidak memerlukan konsistensi data secara objektif, selain itu Adam juga berkerja dengan baik pada jenis *sparse gradien*, dan secara alami melakukan penguatan terhadap *step-size* (Kingma & Ba, 2014).

2.2.9 Model Prediksi Regresi

Untuk mengukur data kuantitatif dimana model akan memprediksi hasil numerik, beberapa ukuran akurasi biasanya digunakan untuk mengevaluasi keefektifan model. Namun, ada berbagai cara untuk mengukur akurasi, masing-masing dengan nuansa tersendiri. Metode yang paling umum untuk mengkarakterisasi kemampuan prediktif model adalah menggunakan Root Mean Square Error (RMSE). Visualisasi dari model yang sesuai, terutama plot telah ada, sangat penting untuk memahami apakah model sesuai untuk tujuan penelitian (Kuhn & Johnson, 2013).

2.2.10 Peramalan Indeks Saham

Prediksi pasar saham, biasanya dianggap sebagai tugas yang menantang karena karakteristiknya yang bergejolak dan kacau (Baohua, et al., 2012). Hal ini terjadi karena harga saham dipengaruhi oleh banyak faktor ekonomi makro seperti peristiwa politik, kebijakan perusahaan, kondisi ekonomi umum, indeks harga komoditas, suku bunga dan nilai tukar, dan ekspektasi investor dan faktor psikologis (Wang, et al., 2012).

2.2.11 Python

Bahasa pemrograman Python yang merupakan salah satu bahasa yang paling populer untuk komputasi ilmiah. Python hadir dengan pustaka standar luas yang berisi dukungan untuk hampir setiap bidang ilmu komputer. Berkat sifat interaktif tingkat tinggi dan ekosistem perpustakaan

ilmiahnya yang semakin matang, ini adalah pilihan yang menarik untuk pengembangan algoritmis dan analisis data eksplorasi (Dubois, 2007; Milmann & Avaizis, 2011).

Dalam penelitian ini, peneliti menggunakan API Keras yang dimana merupakan API yang menyediakan fitur jaringan jaringan saraf tingkat tinggi, yang ditulis dengan Python dan mampu berjalan di atas TensorFlow, CNTK, atau Theano. Ini dikembangkan dengan fokus pada memungkinkan eksperimen cepat. Hal ini sangat dibutuhkan dimana pada penelitian ini membutuhkan dukungan fitur jaringan saraf, mengingat LSTM merupakan algoritma dari model *Recurrent Neural Network*.

