

Penerapan Gated Recurrent Unit untuk Prediksi Pergerakan Harga Saham pada Bursa Efek Indonesia

Joseph Axel Ripto^{#1}

Program Studi Informatika
Institut Teknologi Harapan Bangsa
Bandung, Indonesia
if-18032@students.ithb.ac.id

Hery Heryanto^{#2}

Program Studi Informatika
Institut Teknologi Harapan Bangsa
Bandung, Indonesia
herry_heryanto@ithb.ac.id

Abstract—Predicting stock prices which are time series data was not an easy process. This is due to the characteristics of the data that have high noise, high complexity, and nonlinear structure. Anomaly factors in the stock market also affect the results of stock price predictions. This study tested the GRU model to predict stock prices on the Bursa Efek Indonesia. This study uses four indicators, namely training period, unit, batch size, and epoch. This study shows the effect of each indicator on each dataset. The measuring instrument used to calculate accuracy in this study is the Root Mean Square Error (RMSE). This study was conducted on three main datasets, namely PT. Aneka Tambang, PT. Indofood, and PT. Telkom. Based on the results of the tests carried out in this study, the best accuracy value of 0.009977 was obtained through experiments on the dataset PT. Japfa Comfeed Indonesia. PT. Aneka Tambang produces the best accuracy of 0.012138. Experiments conducted on the PT. Vale Indonesia produced the best accuracy of 0.019060. An experiment at PT. XL Axiata produces the best accuracy of 0.024266. Experiments conducted at PT. Indofood produced the best accuracy of 0.029635. Finally, the experiment on PT. Telkom has the best accuracy of 0.031479. The test results prove that all the indicators used in the test have an influence on the accuracy value.

Keywords—Gated Recurrent Unit (GRU), Recurrent Neural Network, Deep learning, Stock prediction, Time series forecasting.

Abstrak—Prediksi harga saham yang merupakan time series data adalah proses yang tidak mudah. Hal ini dikarenakan karakteristik data saham yang mempunyai noise yang tinggi, kompleksitas yang tinggi, dan struktur nonlinier. Faktor anomali dalam pasar saham juga memengaruhi hasil prediksi harga saham. Penelitian ini menguji model GRU untuk memprediksi harga saham di Bursa Efek Indonesia. Penelitian ini menggunakan empat indikator, yaitu training period, unit, batch size, dan epoch. Penelitian ini memperlihatkan pengaruh dari setiap indikator terhadap masing-masing dataset. Alat ukur yang digunakan untuk menghitung akurasi dalam penelitian ini adalah Root Mean Square Error (RMSE). Penelitian ini dilakukan terhadap enam dataset, yaitu PT. Aneka Tambang, PT. Vale Indonesia, PT. Indofood, PT. Japfa Comfeed Indonesia, PT. XL Axiata, dan PT. Telkom. Berdasarkan hasil pengujian yang dilakukan pada penelitian ini, nilai akurasi terbaik sebesar 0.009977 didapatkan melalui percobaan terhadap dataset PT. Japfa Comfeed Indonesia. PT. Aneka Tambang menghasilkan akurasi terbaik sebesar 0.012138. Percobaan yang dilakukan pada dataset PT. Vale Indonesia menghasilkan akurasi terbaik sebesar 0.019060. Percobaan pada PT. XL Axiata menghasilkan akurasi terbaik sebesar 0.024266. Percobaan yang dilakukan pada PT. Indofood menghasilkan akurasi terbaik sebesar 0.029635. Terakhir, percobaan pada PT. Telkom memiliki akurasi terbaik sebesar 0.031479. Hasil

pengujian membuktikan bahwa semua indikator yang digunakan dalam pengujian memiliki pengaruh terhadap nilai akurasi.

Kata Kunci—Gated Recurrent Unit (GRU), Recurrent Neural Network, Deep learning, Stock prediction, Time series forecasting.

I. PENDAHULUAN

Terdapat dua jenis sekuritas yang diperdagangkan di bursa efek, yaitu saham dan obligasi. Saham dan obligasi adalah instrumen keuangan yang disebut sekuriti. Perbedaan saham dan obligasi terletak pada perlakuan sekuritas terhadap pemilik sekuriti. Pemegang saham dianggap sebagai pemilik perusahaan penerbit saham tersebut, sedangkan pemegang obligasi dianggap sebagai pemberi pinjaman kepada penerbit obligasi [1].

Menurut artikel yang ditulis oleh Setiani Widiarti, seorang investor harus memutuskan kapan dan ke mana dirinya menginvestasikan pendapatannya. Tujuan berinvestasi adalah memperoleh keuntungan. Keuntungan yang didapat investor bergantung pada nilai harga saham [1]. Proses jual beli saham di Indonesia dilakukan di Bursa Efek Indonesia untuk perusahaan-perusahaan yang telah terdaftar [2].

Prediksi harga saham yang merupakan data *time series*, data yang bergantung pada waktu, adalah proses yang tidak mudah. Hal ini dikarenakan karakteristik data saham yang mempunyai *noise* yang tinggi, kompleksitas yang tinggi, dan struktur nonlinier [3]. Faktor anomali dalam pasar saham juga memengaruhi hasil prediksi harga saham. Pada beberapa tahun terakhir, studi yang sudah ada telah menunjukkan bahwa teknologi *Machine Learning* (ML) mampu secara efektif menangkap struktur nonlinier dalam data pasar saham yang kompleks [4].

Hingga saat ini, banyak penelitian yang mengangkat topik tentang memprediksi pergerakan harga saham. Tidak sedikit juga metode yang digunakan untuk membuat model prediksi pergerakan harga saham. Salah satu model yang digunakan untuk memprediksi pergerakan saham adalah *Support Vector Machine* (SVM). Model SVM yang diuji dengan menggunakan data SSE 50 Index menghasilkan akurasi sebesar 71,33%. Nilai akurasi tersebut meningkat menjadi 89,93% setelah menambahkan *sentiment variables* kedalam data yang diuji [5].

Selain SVM, metode yang digunakan untuk model prediksi pergerakan harga saham adalah *Long Short-Term Memory* (LSTM). Hasil penelitian yang dilakukan oleh S. Chen menunjukkan bahwa LSTM mempunyai hasil *Mean Squared Error* (MSE) sebesar 0.0072. MSE yang dihasilkan oleh model LSTM lebih kecil dibandingkan dengan *Principal Component Analysis - Support Vector Machine* (PCA-SVM) dan Random Forest. MSE yang lebih kecil tersebut menandakan bahwa LSTM mempunyai kesalahan yang lebih kecil dalam prediksi harga saham. Selain menggunakan LSTM, S. Chen mencoba untuk melakukan seleksi fitur dengan metode *Genetic Algorithm* (GA). Percobaan tersebut menghasilkan MSE yang lebih kecil dibandingkan dengan LSTM [6].

Penelitian tahun 2019 menunjukkan bahwa *Gated Recurrent Unit* (GRU) dan LSTM memiliki performa yang lebih baik dibandingkan *Multilayer Perceptron* (MLP) dan SVM. Model GRU memiliki *return ratio* (keuntungan) paling besar, sebesar 5.722242 yang menunjukkan bahwa model tersebut memprediksi pergerakan harga saham dengan baik. GRU memiliki nilai RMSE sebesar 0.000511. RMSE pada GRU memiliki perbedaan 0.0001 dengan hasil RMSE pada LSTM [7].

Penelitian ini menguji metode GRU untuk memprediksi harga saham. GRU adalah model LSTM yang telah dimodifikasi. GRU menggunakan dua *gate* dalam prosesnya, yaitu *Reset gate* dan *update gate* yang digunakan untuk *hidden state*. *Reset gate* menentukan memperhitungkan kombinasi *input* saat ini dengan *historic memory* (data masa lalu yang disimpan). *Update gate* bertanggung jawab untuk memutuskan tingkatan *historic memory* mana yang harus dipertahankan dalam *node* [7].

II. METODOLOGI

A. Teori Prediksi Saham

Terdapat tiga tipe cara manusia memprediksi harga saham, yaitu *technical analysis*, *fundamental analysis*, dan *gambler* (asal tebak). Mental atau emosi dari manusia itu sendiri juga termasuk salah satu faktor yang menentukan keputusan dalam menjual atau membeli saham. Jika seorang investor mempunyai ekspektasi bahwa suatu saham akan naik, maka investor itu akan membeli saham tersebut. Sebaliknya jika seorang investor mempunyai ekspektasi bahwa suatu saham akan turun, maka investor itu akan menjual saham tersebut [8].

Technical analysis adalah pengetahuan tentang harga dengan grafik sebagai alat utamanya. *Technical analysis* menganalisis pola data atau pola grafik dalam memprediksi harga saham. *Technical analysis* berbicara tentang sifat tren harga, konfirmasi dan divergensi, volume yang mencerminkan perubahan harga, dan istilah *support/resistance* [8].

Buku yang ditulis oleh Steven B. Achelis menggambarkan bahwa harga saham adalah hasil dari pertarungan satu lawan satu antara pembeli dan penjual. Pembeli menekan harga menjadi lebih tinggi dan penjual menekan harga lebih rendah (mengikuti prinsip *supply and demand*). Arah harga naik atau turun adalah hasil dari siapa yang menang dalam pertarungan

tersebut. Arah naik atau turunnya pergerakan saham dalam periode tertentu biasa disebut dengan tren [8].

Support adalah nilai batas atau area di mana harga saham berada di bawah. Area *support* tercipta karena adanya dukungan dari pembeli sehingga harga saham tidak jatuh terlalu jauh. Selain karena dukungan dari pembeli, area *support* juga tercipta karena penjual tidak mau menjual sahamnya di bawah harga tersebut. Pada area *support*, jumlah pembeli lebih banyak dari jumlah penjual sehingga nilai saham mengalami peningkatan. Area *support* yang kuat menyebabkan nilai saham sulit untuk melewati batas atau turun lebih jauh [8].

Resistance adalah nilai batas atau area di mana harga saham berada di atas. Area *resistance* tercipta karena adanya pengaruh dari penjual terhadap saham tersebut. Penjual mencegah harga saham menjadi terlalu mahal yang menyebabkan pembeli tidak mau membeli saham tersebut. Pada area *resistance*, jumlah penjual lebih banyak dari jumlah pembeli sehingga nilai saham mengalami penurunan. Area *resistance* yang kuat menyebabkan nilai saham sulit untuk melewati batas atau naik lebih jauh [8].

Area *support* dan *resistance* menahan harga saham agar tidak melewati batas area tersebut. Akan tetapi, tidak menutup kemungkinan harga saham untuk menembus batas area *support* atau *resistance*. Peristiwa di mana harga saham menembus batas area *support* atau *resistance* disebut *breakout*. Ketika terjadi *breakout*, area *support* yang lama menjadi area *resistance* yang baru, berlaku sebaliknya [8].

Fundamental analysis adalah cara memprediksi saham berdasarkan laporan keuangan triwulan perusahaan terkait atau berita yang berhubungan dengan perusahaan terkait. Jika ada berita yang negatif tentang perusahaan terkait, maka investor tidak membeli saham perusahaan tersebut. Jika terjadi kenaikan keuntungan pada laporan keuangan perusahaan terkait, maka investor membeli saham perusahaan tersebut [8].

Gambler adalah salah satu cara manusia untuk memprediksi harga saham akan naik atau turun. Tidak sedikit investor yang menggunakan cara ini. Ketika investor membeli sebuah saham atau sekuriti, investor tersebut tidak tahu harga saham tersebut akan naik atau turun. Tetapi jika investor tersebut membeli saham ketika tren harga sedang naik, maka kemungkinan investor tersebut mendapat keuntungan akan meningkat [8].

Berdasarkan teori Dow market dibagi menjadi tiga tren. Tren tersebut, yaitu *Primary trend*, *Secondary trends*, dan *Minor trends*. *Primary trend* bertahan satu tahun sampai beberapa tahun. Artinya, harga saham konsisten naik atau turun dalam satu tahun sampai beberapa tahun. *Secondary trends* bertahan satu bulan sampai tiga bulan. Artinya, harga saham konsisten naik atau turun dalam satu bulan sampai tiga bulan. *Secondary trends* merupakan bentuk koreksi dari *Primary trend*. Jika *Primary trend* merupakan tren naik maka *Secondary trends* merupakan tren turun. Terakhir *Minor trends* adalah tren yang bertahan hanya satu hari sampai tiga minggu [8]. Penelitian ini menggunakan *Primary trend* selama satu tahun dan lima tahun untuk memprediksi *Secondary trends* selama 100 hari.

B. Pembelajaran Mesin

Pembelajaran mesin atau *machine learning* merupakan percabangan dari kecerdasan buatan. Pembelajaran mesin juga mempunyai bagian lebih kompleks yang disebut *Deep Learning* [9]. Pembelajaran mesin adalah ilmu dan seni dari pemrograman komputer sehingga komputer mampu belajar dari data. Komputer seolah diberi kemampuan untuk belajar tanpa harus diprogram secara eksplisit. Jadi, Pembelajaran mesin memberikan komputer kemampuan belajar untuk menyelesaikan masalah berdasarkan pola data.

Pembelajaran mesin menyederhanakan kode dalam program dan memiliki kinerja lebih baik daripada pendekatan tradisional. Pembelajaran mesin menyelesaikan masalah rumit yang tidak bisa diselesaikan dengan pendekatan tradisional. Selain itu, Pembelajaran mesin mampu menyelesaikan masalah yang melibatkan data dalam jumlah besar. Pembelajaran mesin juga beradaptasi dengan data baru [10].

Pembelajaran mesin dibagi menjadi empat tipe, yaitu *supervised learning*, *unsupervised learning*, *semisupervised learning*, dan *reinforcement learning*. Pada *Supervised Learning*, data latihan yang diberikan pada model disertakan dengan solusi yang diinginkan, biasa disebut label. *Supervised learning* biasa digunakan untuk menyelesaikan masalah klasifikasi atau regresi. *Unsupervised learning* tidak membutuhkan label pada data latihannya. *Unsupervised learning* biasanya digunakan untuk menyelesaikan masalah pengelompokan, deteksi anomali, *association rule learning*, visualisasi, dan pengurangan dimensi. *Semisupervised learning* adalah gabungan dari *supervised learning* dan *unsupervised learning*. *Reinforcement learning* sangat berbeda dari tipe pembelajaran mesin yang lain. *Reinforcement learning* menggunakan sistem pembelajaran yang disebut agen yang melakukan aksi dan mengamati lingkungan. Setelah berhasil melakukan aksi, agen akan mendapatkan *reward* atau *penalty*. *Reinforcement learning* bertujuan untuk mencari strategi terbaik agar agennya mendapatkan poin *reward* sebanyak mungkin [10].

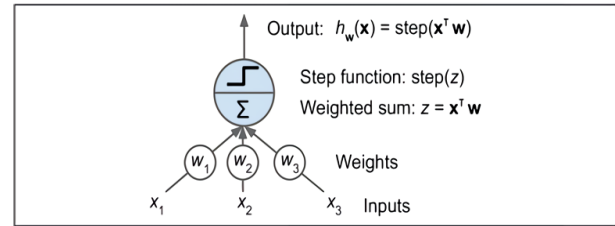
C. Artificial Neural Network (ANN)

Artificial Neural Network (ANN) adalah inti dari *Deep Learning*. ANN terinspirasi dari studi tentang sistem saraf pusat atau neuron pada otak manusia. ANN terdiri dari kumpulan neuron yang saling berhubungan dan terdiri dari beberapa lapisan. Cara kerja dasar ANN adalah neuron pada suatu lapisan memberikan pesan kepada neuron pada lapisan selanjutnya. Dalam ANN terdapat beberapa arsitektur, yaitu *Perceptron* dan *Multi Layer Perceptron* (MLP) [9] [10].

1) Perceptron

Perceptron adalah salah satu arsitektur ANN yang paling simpel. Gambar ?? merupakan gambar dari sebuah proses komputasi neuron buatan yang disebut *Threshold Logic Unit* (TLU), atau terkadang disebut *Linear Threshold Unit* (LTU). Masukan dan keluaran yang dihasilkan dari TLU berupa angka dan setiap masukan diasosiasikan dengan sebuah bobot. TLU menghitung jumlah bobot dari setiap masukan dengan rumus ($z = w_1x_1 + w_2x_2 + \dots + w_nx_n = x^T w$), lalu jumlah

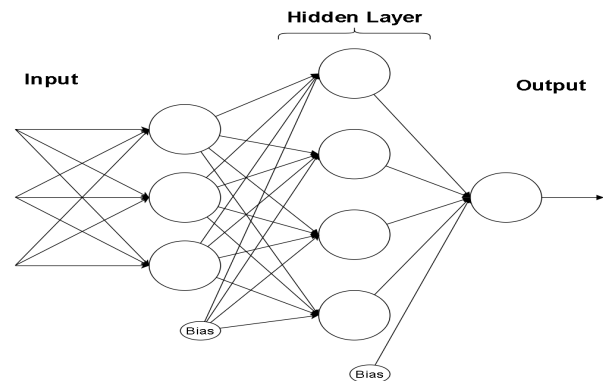
bobot tersebut dimasukkan ke dalam *step function* sehingga menghasilkan keluaran seperti pada gambar berikut [10].



gambar 1. *Threshold logic unit*: Sebuah neuron buatan yang menghitung jumlah bobot dari setiap masukan lalu diterapkan ke dalam *step function* [10]

2) Multi Layer Perceptron (MLP)

Multi Layer Perceptron (MLP) adalah salah satu contoh arsitektur ANN lainnya yang memiliki banyak lapis. MLP memiliki lapisan yang tersembunyi, terletak diantara lapisan masukan (*input layer*) dan lapisan keluaran (*output layer*), disebut *hidden layer*. Proses yang terjadi pada lapisan masukan dan lapisan keluaran terlihat dari luar arsitektur, di sisi lain proses yang terjadi pada lapisan tersembunyi tidak bisa dilihat dari luar. Arsitektur MLP dapat dilihat pada gambar di bawah [9].



gambar 2. Contoh arsitektur *Multi Layer Perceptron* (MLP)

Lingkaran yang terdapat pada gambar di atas disebut node atau neuron buatan. Konsep neuron buatan pada MLP sama dengan konsep pada *perceptron*. Neuron tersebut menghitung jumlah bobot dari setiap masukan lalu diterapkan ke dalam *step function*. Setelah ada penelitian tentang algoritme *back-propagation*, *step function* diganti menjadi fungsi *sigmoid*. *step function* dan fungsi *sigmoid* disebut dengan fungsi aktivasi [9].

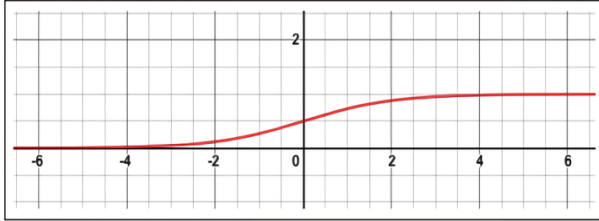
D. Fungsi Aktivasi

Fungsi aktivasi adalah rangkaian rumus matematika yang mengubah suatu nilai. Fungsi aktivasi dalam ANN dipakai ketika jumlah bobot yang sudah ditambahkan bias diteruskan ke lapisan selanjutnya. Fungsi aktivasi memastikan adanya nonlinearitas di antara lapisan *neural network*. Jika pada setiap lapisan *neural network* tidak mempunyai nonlinieritas, maka sebanyak apa pun lapisan tersembunyi yang digunakan setara dengan hanya memakai satu lapisan tersembunyi. Fungsi aktivasi pada *neural network* yang digunakan dalam arsitektur

GRU adalah fungsi *sigmoid* dan fungsi *hyperbolic tangent* (tanh) [10].

1) Fungsi Sigmoid

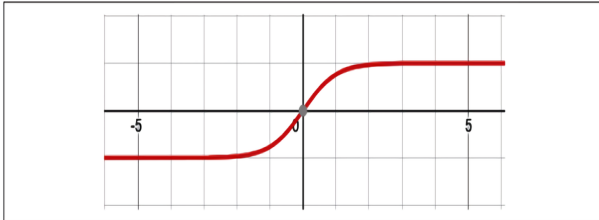
Fungsi *Sigmoid* adalah fungsi aktivasi yang menghasilkan nilai dengan skala 0 sampai 1. Fungsi *Sigmoid* menerima masukan dari skala negatif tak hingga sampai tak hingga. Fungsi *sigmoid* didefinisikan dengan persamaan $\sigma(x) = \frac{1}{1+e^{-x}}$. Fungsi *sigmoid* merupakan fungsi aktivasi nonlinier. Tipikal fungsi *Sigmoid* direpresentasikan dalam gambar berikut [9].



gambar 3. Fungsi *sigmoid* dengan keluaran dalam skala 0 sampai 1 [9]

2) Fungsi Hyperbolic Tangent (tanh)

Fungsi *Hyperbolic Tangent* (tanh) adalah fungsi aktivasi nonlinier yang menghasilkan nilai dengan skala -1 sampai 1. Skala -1 sampai 1 membuat keluaran setiap lapisan kurang lebih berpusat di sekitar 0 di awal pelatihan, yang sering membantu mempercepat konvergensi. Fungsi tanh menerima masukan dari skala negatif tak hingga sampai tak hingga. Fungsi tanh memiliki persamaan $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$. Fungsi tanh direpresentasikan dalam gambar II-D2 [9] [10].



gambar 4. Fungsi *Hyperbolic Tangent* (tanh) dengan keluaran dalam skala -1 sampai 1 [9]

E. Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) merupakan salah satu kelas dari *neural network*. RNN lebih populer digunakan untuk masukan berupa teks. RNN memanfaatkan masukan yang memiliki karakteristik berurutan. Masukan yang memiliki karakteristik berurutan memiliki beberapa bentuk, yaitu teks, ucapan, dan deret waktu. Masukan yang memiliki karakteristik berurutan adalah data berurutan yang bergantung atau memiliki hubungan dengan data sebelumnya [9].

Tradisional MLP membuat asumsi bahwa semua masukan tidak memiliki hubungan dengan masukan lainnya. Asumsi ini tidak berlaku pada banyak tipe data berurutan. Contohnya, kata dalam kalimat, notasi musik dalam komposisi, harga saham dari waktu ke waktu, atau bahkan molekul dalam suatu senyawa. Oleh karena itu, RNN membuat *hidden state* atau memori yang menyimpan esensial dari rentetan masukan yang

terjadi sampai suatu waktu. Perhitungan dasar yang terjadi pada *hidden state* atau memori diperlihatkan pada persamaan berikut [9] [10].

$$h_t = f(h_{t-1}, X_t) \quad (2.1)$$

Keterangan :

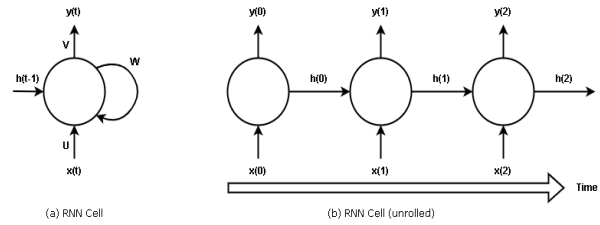
h_t : Nilai *hidden state* pada waktu t

h_{t-1} : Nilai *hidden state* pada waktu $t-1$

X_t : Nilai masukan pada waktu t

f : Fungsi untuk menghitung Nilai *hidden state* pada waktu t

RNN menggunakan tiga matriks bobot U , V , dan W sebagai parameter. Bobot U adalah bobot dari masukan, bobot V adalah bobot dari keluaran, dan bobot W adalah bobot dari *hidden state* atau memori.



gambar 5. (a) sistematis dari sebuah sel RNN; (b) sel RNN (unrolled)

Gambar 5(b) adalah bentuk lengkap dari gambar 5(a) jika bagian pengulangannya dijabarkan. Gambar 5(b) mempunyai tiga langkah waktu. Bobot U , V , dan W dibagikan pada setiap langkah waktu tersebut. $y(t)$ merupakan keluaran pada waktu t yang diwakili oleh nilai $h(t)$ atau *hidden state* pada waktu t . Perhitungan $y(t)$ dapat dilihat pada persamaan ???. *Hidden state* pada waktu t digunakan pada perhitungan *hidden state* pada waktu $t+1$. Perhitungan tersebut dapat dilihat pada persamaan 2.2 [9] [10].

$$h_t = \tanh(W h_{t-1} + U x_t) \quad (2.2)$$

$$y_t = \text{softmax}(V h_t) \quad (2.3)$$

Keterangan :

h_t : Nilai *hidden state* pada waktu t

h_{t-1} : Nilai *hidden state* pada waktu $t-1$

W : Nilai bobot dari *hidden state* pada waktu $t-1$

U : Nilai bobot dari masukan pada waktu t

x_t : Nilai masukan pada waktu t

$\tanh$: Fungsi aktivasi *Hyperbolic Tangent*

y_t : Nilai Keluaran pada waktu t

V : Nilai bobot dari *hidden state* pada waktu t

softmax : Fungsi aktivasi *softmax*

F. Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) adalah salah satu alat ukur kinerja sebuah kasus regresi. RMSE memberi gambaran tentang bagaimana banyak kesalahan yang dibuat sistem dalam hasil prediksinya, dengan memberi bobot yang lebih tinggi

untuk kesalahan besar. Persamaan 2.4 memperlihatkan rumus matematika untuk menghitung nilai RMSE [10].

$$RMSE(X, h) = \sqrt{\frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})^2} \quad (2.4)$$

Keterangan :

- m : Jumlah *instance* atau baris data dalam kumpulan data.
 $x^{(i)}$: Vektor dari semua nilai fitur (tidak termasuk label) dari *instance* atau baris data ke-i dalam kumpulan data
 $y^{(i)}$: Vektor berisi label (nilai keluaran yang diinginkan untuk baris ke-i)
X : Matriks yang berisi semua nilai fitur (tidak termasuk label) dari semua baris dalam kumpulan data.
h : Fungsi prediksi sistem atau hipotesis.
RMSE(X,h) : Fungsi biaya yang diukur pada kumpulan data menggunakan hipotesis h

G. Normalisasi Min-max

Normalisasi *Min-max* adalah salah satu teknik untuk mengubah nilai dari fitur yang dimiliki. Pada Normalisasi *Min-max*, nilai digeser dan dihitung ulang sehingga menghasilkan skala nol hingga satu. Normalisasi *Min-max* mengurangi nilai x dengan nilai minimum dan membaginya dengan nilai maksimum dikurangi dengan nilai minimum seperti pada persamaan 2.5. *Library Scikit-Learn* menyediakan fungsi untuk menghitung normalisasi *Min-max* yang disebut *MinMaxScaler*. Namun, penelitian ini membuat normalisasi *Min-max* dari awal [10].

$$x_{scaled} = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (2.5)$$

Keterangan :

- x_{scaled} : Hasil normalisasi sebuah nilai x
 x_i : Nilai ke-i dari kumpulan data yang ingin dinormalisasi.
 $\min(x)$: nilai minimum dari kumpulan data
 $\max(x)$: nilai maksimum dari kumpulan data

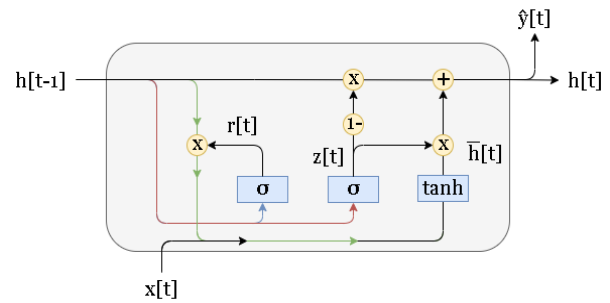
H. Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) adalah varian *gated RNN* selain dari *Long Short-Term Memory* (LSTM). GRU pertama kali diperkenalkan oleh Kyunghyun Cho pada tahun 2014. Perbedaan utama GRU dengan LSTM adalah bahwa satu unit gerbang secara bersamaan mengontrol faktor lupa dan keputusan untuk memperbarui unit dalam *hidden state*. Dengan kata lain GRU menggabungkan proses untuk menentukan faktor unit yang ingin dibuang dan unit mana yang ingin diperbarui ke dalam satu *gate* yang disebut *update gate* [9] [10].

Buku yang ditulis oleh Antonio Gulli menyatakan bahwa GRU dan LSTM menyelesaikan masalah *Vanishing gradient* tapi GRU lebih cepat saat dilatih karena komputasinya lebih sedikit [9]. Menurut buku yang ditulis Aurelien Geron, GRU mempunyai performa yang sama bagusnya dengan LSTM

[10]. Tujuan utama GRU adalah untuk menangkap dependensi atau faktor yang berpengaruh terhadap hasil prediksi di waktu yang berbeda dengan perubahan yang adaptif. Sebagai contoh, manusia tidak membutuhkan semua informasi masa lalu untuk membuat keputusan. Jika seseorang merencanakan untuk pergi liburan, maka informasi masa lalu seperti nilai ujian kemarin tidak banyak memengaruhi keputusan untuk tempat tujuan pergi liburan.

GRU menggunakan tipe *hidden unit* baru yang terinspirasi dari LSTM. Di dalam struktur GRU, terdapat komponen yang disebut *gate* berfungsi untuk mengatur alur informasi model GRU. GRU hanya mempunyai dua *gate*, yaitu *reset gate* dan *update gate*. *Reset gate* pada GRU menggabungkan *input* baru dengan informasi masa lalu. *Update gate* menentukan berapa banyak informasi masa lalu yang harus tetap disimpan [9].



gambar 6. Struktur Gated Recurrent Unit

Keterangan :

- $r[t]$: Reset gate
 $z[t]$: Update gate
 $\tilde{h}[t]$: New remember atau hidden state
 $h[t]$: Hidden state akhir
 $y[t]$: output pada waktu t
 $x[t]$: Input baru
 $h[t - 1]$: Informasi masa lalu

Dalam GRU hal pertama yang terjadi adalah *reset gate* dikomputasi untuk menggabungkan informasi masa lalu dan input baru dengan rumus sebagai berikut:

$$r_t = \sigma(W_r h_{t-1} + U_r x_t + b_r) \quad (2.6)$$

Di mana σ adalah fungsi aktivasi *sigmoid*, b_r adalah bias dan W_r, U_r adalah weight dari persamaan tersebut. Setelah *reset gate*, komputasi dilanjutkan ke *new remember* atau *hidden state* dengan rumus sebagai berikut:

$$\tilde{h}[t] = \tanh(W \cdot r_t * h_{t-1} + U x_t + b) \quad (2.7)$$

Setelah itu masuk ke dalam *update gate* untuk menentukan berapa banyak informasi masa lalu yang harus tetap disimpan dengan rumus sebagai berikut:

$$z_t = \sigma(W_z h_{t-1} + U_z x_t + b_z) \quad (2.8)$$

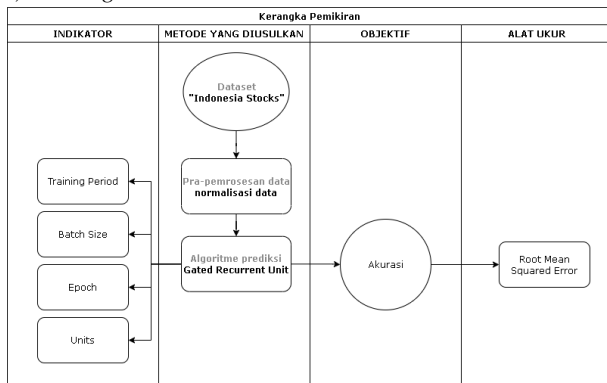
Terakhir, *hidden state* final dihitung dengan rumus sebagai berikut:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}[t] \quad (2.9)$$

Dari *hidden state* diakhiri dengan penentuan hasil prediksi [9].

I. Perancangan Sistem

1) Kerangka Pemikiran



gambar 6. Diagram Kerangka Pemikiran

Pertimbangan parameter pada bagian indikator:
 $\text{nolistsep, leftmargin=0.5cm}$

- 1) *Training Period* adalah perlakuan berbeda pada saat pelatihan model khususnya pada periode yang digunakan. Data latihan dibagi menjadi dua yaitu, data dengan periode satu tahun dan data dengan periode lima tahun [16].
- 2) *Batch size* adalah hyperparameter yang digunakan untuk menentukan jumlah batch pada saat melatih model GRU. Menurut penelitian tahun 2019, *batch size* optimal berada pada angka 32 [16]. Penelitian ini menguji *batch size* sebesar 32, 64, dan 128.
- 3) *Epoch* adalah hyperparameter yang digunakan untuk menentukan jumlah iterasi pada saat melatih model GRU. Menurut penelitian tahun 2019, *epoch* memiliki hasil terbaik pada angka 100 [16]. Penelitian ini menguji *epoch* sebesar 10, 100, dan 1000.
- 4) *Units* adalah parameter yang digunakan saat membentuk model GRU. *Units* digunakan untuk menentukan jumlah *perceptron* pada satu layer. Penelitian ini menguji *units* sebesar 16, 32, 64, dan 128.

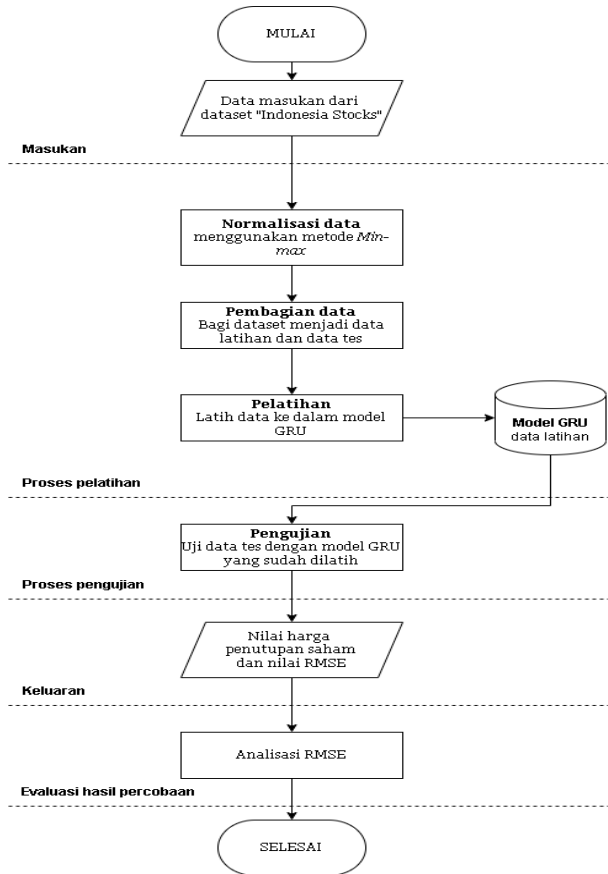
Bagian metode yang diusulkan menjelaskan metode-metode yang diterapkan untuk melakukan prediksi harga saham dalam penelitian ini. *Dataset* yang digunakan adalah *dataset* "Indonesia Stocks" dan "Dataset Saham Indonesia / Indonesia Stock Dataset" yang didapat dari situs Kaggle. Data dinormalisasi sehingga skala data tidak terlalu jauh. Proses normalisasi data dalam penelitian ini menggunakan normalisasi *Min-Max*. Setelah itu, data dibagi menjadi dua kelompok, yaitu data latihan dan data tes. Data latihan yang digunakan dibagi menjadi dua periode, yaitu periode satu tahun dan periode dua tahun. Selanjutnya, data latihan dilatih menggunakan model GRU. Setelah dilatih, model GRU siap untuk melakukan prediksi terhadap data tes.

Objektif adalah acuan pengukuran evaluasi model. Penelitian ini berfokus untuk menguji nilai akurasi prediksi harga saham. Bagian pengukuran adalah alat yang dipakai untuk mengukur suatu Objektif. Penelitian ini menggunakan *Root Mean Squared Error* (RMSE) untuk mengukur akurasi yang dihasilkan oleh model prediksi harga saham.

2) Flowchart Global

Bagian ini menjelaskan urutan proses yang dilalui dalam penelitian secara global. Urutan proses global dapat dilihat pada gambar 7. Dalam penelitian ini hal pertama yang dilakukan adalah memasukkan *dataset* ke dalam program. Selanjutnya, data yang sudah dimuat ke dalam sistem dinormalisasi menggunakan normalisasi *Min-max*. Setelah data sudah diolah, data tersebut dibagi menjadi data latihan untuk proses pelatihan model dan data tes untuk proses pengujian model.

Penelitian ini menyiapkan dua periode data latihan. data latihan pertama menggunakan data dengan periode satu tahun dan data latihan kedua menggunakan data dengan periode lima tahun. Selanjutnya, data latihan dilatih menggunakan model GRU. Lalu, Model tersebut dipakai untuk memprediksi data yang digolongkan ke dalam data tes. Hasil prediksi tersebut dievaluasi dengan menganalisis nilai RMSE. Nilai RMSE yang semakin kecil menandakan nilai prediksi yang dihasilkan model mendekati nilai aslinya.



gambar 7. Flowchart Urutan Proses Global

3) Dataset

Penelitian ini menggunakan tiga dataset perusahaan, yaitu PT. Aneka Tambang, PT. Indofood, dan PT. Telkom sebagai dataset utama. Dataset perusahaan PT. XL Axiata, PT. Vale Indonesia, dan PT. Japfa Comfeed Indonesia dijadikan pembandingan untuk dataset utama. Semua data yang ada pada dataset tersebut digunakan secara keseluruhan. Tidak ada fitur yang dibuang kecuali tanggal. Dataset PT. Aneka Tambang memiliki data lebih sedikit dengan selisih dua hari. Dalam dua hari tersebut, PT. Aneka Tambang tidak mempunyai volume. Dengan kata lain, dalam dua hari tersebut tidak terjadi transaksi pada PT. Aneka Tambang. Hal ini dikonfirmasi pada dataset "Dataset Saham Indonesia / Indonesia Stock Dataset". Hal serupa juga terjadi pada PT. Japfa Comfeed Indonesia dan PT. XL Axiata. Namun, selisih tersebut tidak berpengaruh terhadap proses pelatihan karena selisih tersebut dimasukkan ke dalam data pengujian. Pembagian dataset dapat dilihat pada tabel berikut.

No	Dataset	Training period	Total Data	Pembagian Data	
				Pelatihan	Pengujian
1	ANTM	1 tahun	1501	235	100
2		5 tahun	1501	1191	100
3	INDF	1 tahun	1503	235	100
4		5 tahun	1503	1191	100
5	TLKM	1 tahun	1503	235	100
6		5 tahun	1503	1191	100
7	INCO	1 tahun	1503	235	100
8		5 tahun	1503	1191	100
9	JPFA	1 tahun	1502	235	100
10		5 tahun	1502	1191	100
11	EXCL	1 tahun	1504	235	100
12		5 tahun	1504	1191	100

III. HASIL DAN PEMBAHASAN

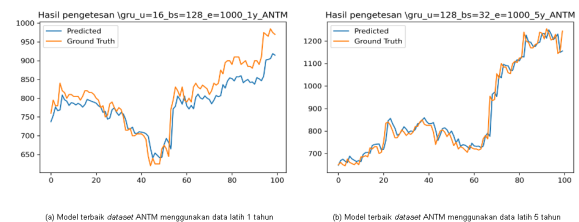
A. Hasil Pengujian pada Dataset Utama

Pengujian ini dilakukan sebanyak kombinasi training period, unit, batch size, dan epoch yang dijelaskan pada bagian kerangka pemikiran. Tabel Berikut memperlihatkan kombinasi parameter masing-masing dataset yang menghasilkan nilai RMSE paling kecil.

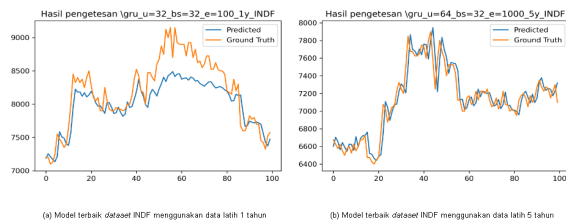
No	Dataset	Training period	Parameter			RMSE
			Unit	Batch Size	Epoch	
1	ANTM	1 tahun	16	128	1000	0.013073
2		5 tahun	128	32	1000	0.012138
3	INDF	1 tahun	32	32	100	0.065499
4		5 tahun	64	32	1000	0.029635
5	TLKM	1 tahun	64	128	100	0.031611
6		5 tahun	64	32	1000	0.031479

Training periode lima tahun menghasilkan nilai RMSE terbaik dari semua percobaan. Percobaan pada dataset PT. Aneka Tambang menunjukkan model terbaik menggunakan training period lima tahun. Model tersebut menggunakan parameter unit sebesar 128, batch size sebesar 32 dan epoch sebesar 1000. Percobaan pada dataset PT. INDF menunjukkan model terbaik menggunakan training period lima tahun. Model tersebut menggunakan parameter unit sebesar 64, batch size sebesar 32 dan epoch sebesar 1000.

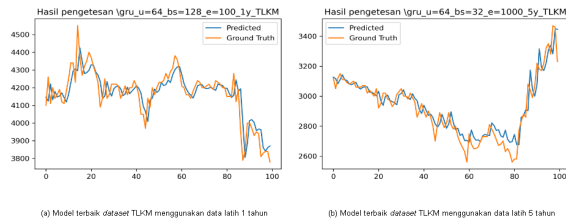
Percobaan pada dataset PT. Telkom menunjukkan model terbaik menggunakan training period lima tahun. Model tersebut menggunakan parameter unit sebesar 64, batch size sebesar 32 dan epoch sebesar 1000. Perbandingan hasil prediksi harga saham yang dihasilkan oleh model pada tabel di atas dapat dilihat pada gambar berikut.



gambar 8. Model Terbaik ANTM



gambar 9. Model Terbaik INDF



gambar 10. Model Terbaik TLKM

B. Hasil Pengujian Dataset Pembanding Menggunakan Parameter Terbaik pada Dataset Utama

Pengujian ini dilakukan sebanyak kombinasi *training period*, *unit*, *batch size*, dan *epoch* yang dijelaskan pada bagian kerangka pemikiran. Pengujian ini bertujuan untuk menguji apakah parameter terbaik pada *dataset* utama dapat menghasilkan nilai akurasi terbaik pada *dataset* pembanding dengan catatan perusahaan memiliki sektor yang sama. Berikut adalah hasil percobaan menggunakan parameter terbaik pada *dataset* utama.

No	Dataset	Training period	Parameter			RMSE
			Unit	Batch Size	Epoch	
1	ANTM	5 tahun	128	32	1000	0.012138
2	INDF	5 tahun	64	32	1000	0.029635
3	TLKM	5 tahun	64	32	1000	0.031479

Tabel Hasil Terbaik pada *Dataset* Utama

No	Dataset	Training period	Parameter			RMSE
			Unit	Batch Size	Epoch	
1	INCO	5 tahun	128	32	1000	0.020385
2	JPFA	5 tahun	64	32	1000	0.010212
3	EXCL	5 tahun	64	32	1000	0.024334

Tabel Pengujian *Dataset* Pembanding Menggunakan Parameter Terbaik pada *Dataset* Utama

Dari tabel di atas menunjukkan bahwa hasil pengujian *dataset* pembanding menggunakan parameter terbaik pada *dataset* utama menghasilkan nilai RMSE yang cukup baik. Setelah menguji setiap kombinasi parameter pada *dataset* pembanding, kombinasi parameter yang menghasilkan nilai RMSE terbaik ditunjukkan pada tabel berikut.

No	Dataset	Training period	Parameter			RMSE
			Unit	Batch Size	Epoch	
1	INCO	1 tahun	16	128	1000	0.035606
2		5 tahun	128	64	10	0.019060
3	JPFA	1 tahun	128	64	1000	0.025999
4		5 tahun	32	32	1000	0.009977
5	EXCL	1 tahun	128	32	1000	0.041279
6		5 tahun	128	128	1000	0.024266

Tabel Model dengan RMSE Terbaik pada *Dataset* Pembanding

Tabel di atas menjelaskan bahwa parameter terbaik pada *dataset* utama masih bisa diterapkan pada *dataset* pembanding dengan sektor yang sama. Akan tetapi, parameter tersebut tidak menghasilkan nilai RMSE terbaik. *Dataset* pembanding harus dilakukan *hyperparameter tuning*, penyetelan terhadap nilai parameter, guna menghasilkan nilai RMSE rendah pada arsitektur tersebut.

IV. SIMPULAN

Percobaan menghasilkan akurasi terbaik sebesar 0.009977. Akurasi tersebut didapatkan melalui percobaan yang dilakukan pada *dataset* pembanding PT. Japfa Comfeed Indonesia. Nilai tersebut dihasilkan dengan menggunakan *training period* lima tahun, *unit* sebesar 32, *batch size* sebesar 32 dan *epoch* sebesar 1000. Percobaan yang dilakukan pada *dataset* PT. Aneka Tambang memiliki akurasi terbaik sebesar 0.012138. Percobaan yang dilakukan pada *dataset* PT. Vale Indonesia menghasilkan akurasi terbaik sebesar 0.019060. Pengujian terhadap *dataset* PT. XL Axiata menghasilkan akurasi terbaik sebesar 0.024266. Percobaan pada *dataset* PT. Indofood menghasilkan akurasi terbaik sebesar 0.029635. Terakhir, PT. Telkom menghasilkan akurasi terbaik sebesar 0.031479.

Indikator *training period*, *unit*, *batch size*, dan *epoch* berpengaruh terhadap akurasi model prediksi harga saham. Setiap *dataset* memiliki kombinasi nilai indikator yang berbeda untuk menghasilkan performa yang baik. Menggunakan *dataset* untuk proses pelatihan dalam jumlah besar belum tentu menghasilkan performa yang lebih baik dibandingkan menggunakan *dataset* dalam jumlah kecil. Nilai parameter *unit* tidak berhubungan dengan panjang data latih yang digunakan. Nilai *batch size* dan *epoch* optimal bergantung pada *dataset* yang digunakan.

Parameter pada *dataset* utama dapat digunakan untuk *dataset* pembanding yang memiliki sektor yang sama. Namun, parameter tersebut tidak menghasilkan nilai akurasi terbaik.

REFERENCES

- [1] Setiani Widiarti, ANALISA PENILAIAN DAN PERHITUNGAN HARGA SAHAM. April 2020. Available https://www.researchgate.net/publication/340595378_ANALISA_PENILAIAN_DAN_PERHITUNGAN_HARGA_SAHAM [Accessed: 11 May 2022]

- [2] Yuneita Anisma, Maret 2012. Faktor-faktor yang mempengaruhi harga saham perusahaan perbankan yang listing di Bursa Efek Indonesia, Vol 2, No. 5. [Accessed: 11 October 2021]
- [3] Fama, E. F., Efficient capital markets: A review of theory and empirical work. *The Journal of Finance* 25 (2), 383-417. 1970. [Accessed: 11 October 2021]
- [4] Guizhu Shen, Qingping Tan, Haoyu Zhang, Ping Zeng, Jianjun Xu, Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions, *Procedia Computer Science*, Volume 131, Pages 895-903, May 2018, <https://doi.org/10.1016/j.procs.2018.04.298>. [Accessed: 27 September 2021]
- [5] R. Ren, D. D. Wu and T. Liu, "Forecasting Stock Market Movement Direction Using Sentiment Analysis and Support Vector Machine," in *IEEE Systems Journal*, vol. 13, no. 1, pp. 760-770, March 2019, doi: 10.1109/JSYST.2018.2794462. [Accessed: 4 October 2021]
- [6] S. Chen and C. Zhou, "Stock Prediction Based on Genetic Algorithm Feature Selection and Long Short-Term Memory Neural Network," in *IEEE Access*, vol. 9, pp. 9066-9072, 2021, doi: 10.1109/ACCESS.2020.3047109. [Accessed: 4 October 2021]
- [7] Sethia A., Raut P. Application of LSTM, GRU and ICA for Stock Price Prediction. In: Satapathy S., Joshi A. (eds) *Information and Communication Technology for Intelligent Systems. Smart Innovation, Systems and Technologies*, vol 107. Springer, Singapore. 2019. https://doi.org/10.1007/978-981-13-1747-7_46 [Accessed: 4 October 2021]
- [8] Achelis, S.B., *Technical analysis from A to Z: covers every trading tool from the absolute breath index to the zig zag* (No. E70 70). McGraw-Hill. 2001.[Accessed: 12 May 2022]
- [9] A. Gulli, A. Kapoor, and S. Pal, *Deep Learning with TensorFlow 2 and Keras: Regression, ConvNets, GANs, RNNs, NLP, and more with TensorFlow 2 and the Keras API*, 2nd Edition, 2nd ed. Birmingham, England: Packt Publishing, 2019. [Accessed: 12 March 2022]
- [10] A. Geron, *Hands-on machine learning with scikit-learn, keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2019. [Accessed: 12 March 2022]
- [11] Pandas Development Team, "Getting started - pandas 1.4.1 documentation," Pandas documentation, Feb 12, 2022. [Online]. Available: https://pandas.pydata.org/docs/getting_started/index.html. [Accessed : 26 March 2022]
- [12] Keras team on Google, "Keras API Reference," Keras documentation. [Online]. Available: <https://keras.io/api/>. [Accessed: 26 March 2022]
- [13] NumPy community, "NumPy: the absolute basics for beginners," NumPy documentation, Jan 14, 2022. [Online]. Available: https://numpy.org/doc/1.22/user/absolute_beginners.html. [Accessed : 26 March 2022]
- [14] John Hunter, Darren Dale, Eric Firing, Michael Droettoom and the Matplotlib development team, "Pyplot function overview," matplotlib.pyplot. [Online]. Available: https://matplotlib.org/stable/api/pyplot_summary.html. [Accessed : 26 March 2022]
- [15] Y. Ji, A. W. -C. Liew and L. Yang, "A Novel Improved Particle Swarm Optimization With Long-Short Term Memory Hybrid Model for Stock Indices Forecast," in *IEEE Access*, vol. 9, pp. 23660-23671, 2021, doi: 10.1109/ACCESS.2021.3056713. [Accessed: 4 October 2021]
- [16] Rahman, M.O., Hossain, M.S., Junaid, T.S., Forhad, M.S.A. and Hossen, M.K., "Predicting prices of stock market using gated recurrent units (GRUs) neural networks," *Int. J. Comput. Sci. Netw. Secur.*, 19(1), pp.213-222, 2019 [Accessed: 12 March 2022]