

Pemanfaatan Recurrent Neural Network (RNN) Untuk Meningkatkan Akurasi Prediksi Mata Uang Pada Forex Trading

Abdillah Baradja*¹, Sukoco²

^{1,2}Fakultas Teknik Elektro dan Informatika, Universitas Surakarta, Indonesia

*dillahbaraja@gmail.com; ²sukoco@unsa.ac.id

Abstrak

Penelitian ini berfokus pada penerapan teknologi machine learning, khususnya Recurrent Neural Network (RNN) dan variasinya seperti Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU), untuk meningkatkan akurasi dalam prediksi pergerakan mata uang di pasar Forex. Dalam konteks di mana machine learning mendominasi analisis data cepat dalam trading forex, risiko keuangan yang tinggi, ketidakpastian pasar yang kompleks, dan tantangan ketidakakuratan prediksi model menjadi kendala utama. Penelitian ini membahas kemampuan RNN dalam memproses data berurutan dan mempertahankan informasi jangka pendek, yang menawarkan solusi terhadap keterbatasan neural network konvensional dalam analisis data berurutan. Dengan menggunakan data harga EUR/USD dari dua tahun terakhir, kami menemukan bahwa implementasi RNN, khususnya melalui LSTM dan GRU, menghasilkan penurunan signifikan dalam kesalahan prediksi, menunjukkan peningkatan kinerja model. Hasil ini menegaskan efektivitas RNN dalam memprediksi dinamika pasar Forex dan menjanjikan pendekatan yang lebih akurat dan andal dalam pemodelan dan peramalan harga mata uang, yang merupakan kunci untuk praktik trading Forex yang sukses. Ini memperkuat potensi machine learning dalam menyediakan alat yang kuat untuk trader dan analis dalam menghadapi pasar yang sangat tidak pasti dan fluktuatif.

Kata Kunci: Machine Learning, Recurrent Neural Network, Long Short-Term Memory, Prediksi Mata Uang, Forex Trading.

1. PENDAHULUAN

Tren perkembangan machine learning dalam trading forex semakin mendominasi, karena teknologi ini dapat digunakan untuk menganalisis data pasar dengan cepat dan akurat, mengidentifikasi pola-pola yang sulit dideteksi oleh manusia, dan mengambil keputusan perdagangan berdasarkan analisis data yang mendalam. Namun, beberapa masalah yang dihadapi dalam penerapan machine learning dalam trading forex termasuk risiko keuangan yang tinggi, model yang tidak selalu dapat diprediksi dengan baik, dan

ketidakpastian pasar yang tetap ada. Risiko keuangan dapat meningkat karena algoritma machine learning dapat membuat keputusan perdagangan yang berisiko jika tidak dikontrol dengan baik [1]. Selain itu, model machine learning tidak selalu dapat memprediksi pergerakan pasar yang kompleks dengan tepat, dan ketidakpastian pasar seperti berita ekonomi atau peristiwa geopolitik dapat membuat algoritma menjadi tidak efektif.

Telah diketahui bahwa neural network (NN) adalah model pembelajaran mesin yang sangat efektif, yang menghasilkan hasil yang canggih untuk berbagai macam aplikasi pembelajaran mesin. Dari sudut pandang algoritma kecerdasan buatan, neural network (NN) disebut sebagai model koneksionis karena terdiri dari neuron buatan, yang merupakan unit dasar yang terhubung yang secara kolektif dapat digabungkan secara berlapis-lapis untuk mengembangkan representasi hirarkis [2]. Berkat peningkatan eksponensial dalam kekuatan pemrosesan selama beberapa tahun terakhir, kecerdasan buatan telah mengalami kemajuan yang signifikan. Munculnya apa yang disebut aplikasi pembelajaran mendalam adalah bukti yang terakhir. Perceptron multilayer adalah contoh model pembelajaran mendalam, yang pada dasarnya adalah jaringan saraf dengan hirarki representasi yang sangat luas (MLP).

Namun demikian, model-model ini tetap memiliki beberapa kelemahan yang signifikan. Sebenarnya tidak mungkin untuk menganalisis deret waktu yang terhubung pada setiap langkah waktu dan menyimpan status lengkap dari urutan tersebut saat bekerja dengan data berurutan. Untuk alasan ini, opsi RNN bisa sangat membantu dalam situasi ini. RNN memproses satu elemen pada satu waktu karena mereka mentransfer data input di sekitar jaringan di seluruh langkah waktu, meskipun mereka masih merupakan model koneksionis [3]. Hidden Markov Models (HMMs), yang sering digunakan untuk memodelkan deret waktu sebagai realisasi dari urutan yang bergantung pada probabilitas dari kondisi yang tidak diketahui, dapat digunakan untuk memberikan opsi yang berbeda untuk mewakili data empiris.

Pendekatan pemrograman dinamis Viterbi biasanya digunakan dalam situasi ini karena memberikan skalabilitas inferensi yang efisien dalam waktu kuadratik. Jika dibandingkan dengan pendekatan HMM, implementasi RNN dapat mempercepat pekerjaan karena hanya membutuhkan satu input dalam satu waktu. Penting untuk dicatat bahwa deret waktu juga dapat dimodelkan dan diramalkan dengan menggunakan teknik lain yang lebih tradisional, seperti ARMA, ARIMA, GARCH, dan lain-lain, atau dengan menggunakan filter stokastik, seperti filter Kalman, dan pendekatan model peralihan, seperti pada, misalnya

[4], [5]. Namun, teknik kotak hitam, seperti NN, sangat menarik karena tidak memerlukan asumsi tentang karakter stokastik dari dinamika yang mendasarinya (lihat, misalnya, [6] dan referensi di dalamnya).

Kendala yang sama berlaku untuk alat filter stokastik seperti halnya pada contoh Kalman, terutama dalam konteks analisis data keuangan. Faktanya, seperti yang bisa kita lihat dari, misalnya, [7] dan referensinya, filter Kalman tidak memiliki sifat yang diperlukan untuk menangkap fluktuasi nilai saham yang cepat, terutama ketika terjadi gejolak keuangan, volatilitas tinggi, dan jaringan keuangan yang terjalin secara rumit.

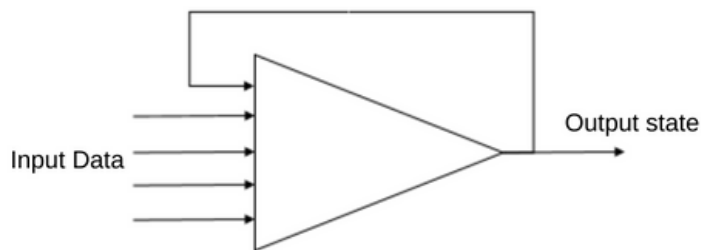
Di bidang keuangan, model pembelajaran mesin pada umumnya dan jaringan saraf pada khususnya telah digunakan secara efektif untuk lindung nilai dan peramalan. Pertimbangan masalah optimasi portofolio [8], di mana penerapan pembelajaran penguatan, jaringan saraf, dan algoritma genetik menghasilkan hasil yang sangat menggembirakan. Model-model seperti ini juga dapat digunakan dalam skenario manajemen risiko, di mana aset-aset yang berbahaya [9] dapat diidentifikasi secara terawasi dengan menggunakan deep neural network (DNN) atau metode pembelajaran mesin tradisional seperti random forest.

Kami meneliti masalah peramalan prediksi harga saham dalam penelitian ini. Mengenai hal yang terakhir, kami ingin menekankan bahwa beberapa strategi telah dikeluarkan. Bahkan jika kita membatasi pertimbangan kita pada aplikasi NNS, kita masih dapat menemukan aplikasi yang terkait dengan teknik MLP, seperti yang ditemukan di [10], convolutional neural network (CNN), [11], jaringan saraf Elman, [12], dan sebagainya. Kami membuat keputusan untuk berkonsentrasi pada analisis arsitektur RNN yang paling mutakhir, dengan perhatian khusus pada GRU dan LSTM.

Masalahnya adalah, sulit untuk menentukan jumlah data yang ideal untuk dianalisis dalam grafik harga. Pola yang berbeda bisa muncul pada interval waktu yang berbeda juga. Bahkan intervalnya sendiri tidak selalu tetap dan bisa berubah tergantung pada situasi saat ini. Beberapa kejadian mungkin jarang terjadi di pasar, tetapi memiliki kemungkinan berhasil yang tinggi. Hal ini bagus jika kejadian tersebut berada dalam jendela data yang dianalisis. Jika kejadian itu berada di luar seri data yang dianalisis, jaringan saraf akan mengabaikannya, meskipun pasar mungkin sedang bereaksi terhadap kejadian itu pada saat itu juga. Meningkatkan jendela analisis akan menyebabkan peningkatan konsumsi

sumber daya komputasi dan memerlukan lebih banyak waktu untuk membuat keputusan.

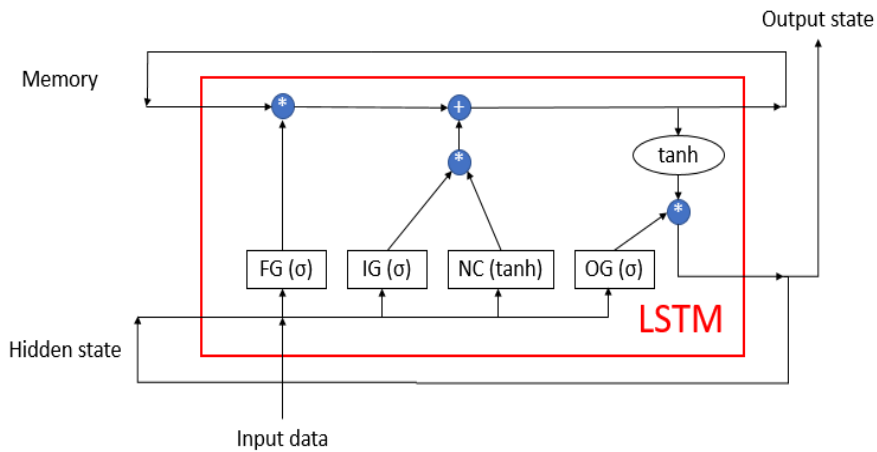
Neuron rekuren telah direkomendasikan sebagai solusi untuk mengatasi masalah analisis seri waktu dalam jaringan saraf. Pendekatan ini bertujuan untuk mengintegrasikan memori jangka pendek dalam jaringan saraf, dimana keadaan saat ini dari sistem diinputkan ke neuron bersamaan dengan keadaan sebelumnya dari neuron yang sama [13]. Proses ini didasarkan pada asumsi bahwa output neuron mencerminkan pengaruh dari semua faktor relevan, termasuk keadaan sebelumnya, memungkinkan transfer pengetahuan secara menyeluruh ke keadaan berikutnya. Ini serupa dengan bagaimana manusia bertindak berdasarkan pengalaman sebelumnya dan tindakan yang telah dilakukan sebelumnya, di mana durasi dan pengaruh memori tergantung pada bobotnya.



Gambar 1. Kerangka kerja recurrent neuron pada jaringan

Namun, solusi yang tampaknya sederhana ini memiliki kelemahan. Pendekatan ini hanya memungkinkan penyimpanan "memori" untuk interval waktu yang singkat. Perkalian siklis sinyal dengan faktor kurang dari satu dan aplikasi fungsi aktivasi neuron menyebabkan penurunan bertahap kekuatan sinyal seiring bertambahnya jumlah siklus. Untuk mengatasi masalah ini, [14] mengusulkan arsitektur Memori Jangka Pendek Panjang (LSTM) pada tahun 1997, yang dianggap sebagai solusi terbaik untuk masalah klasifikasi dan peramalan seri waktu, di mana peristiwa penting terpisah secara waktu dan terentang dalam interval waktu.

Struktur LSTM melampaui definisi neuron konvensional, mewakili sistem jaringan saraf yang lebih kompleks dengan kanal input dan output yang berbeda, serta kanal khusus untuk pertukaran informasi siklik internal, termasuk Memory dan Hidden state.



Gambar 2. Flowchart blok LSTM

Dalam blok LSTM, dua aliran data utama berinteraksi melalui empat lapisan saraf yang saling terhubung, masing-masing diisi dengan jumlah neuron yang sama. Neuron-neuron ini sesuai dengan jumlah aliran output dan memori, memungkinkan pemeriksaan mendetail terhadap dinamika operasional algoritma.

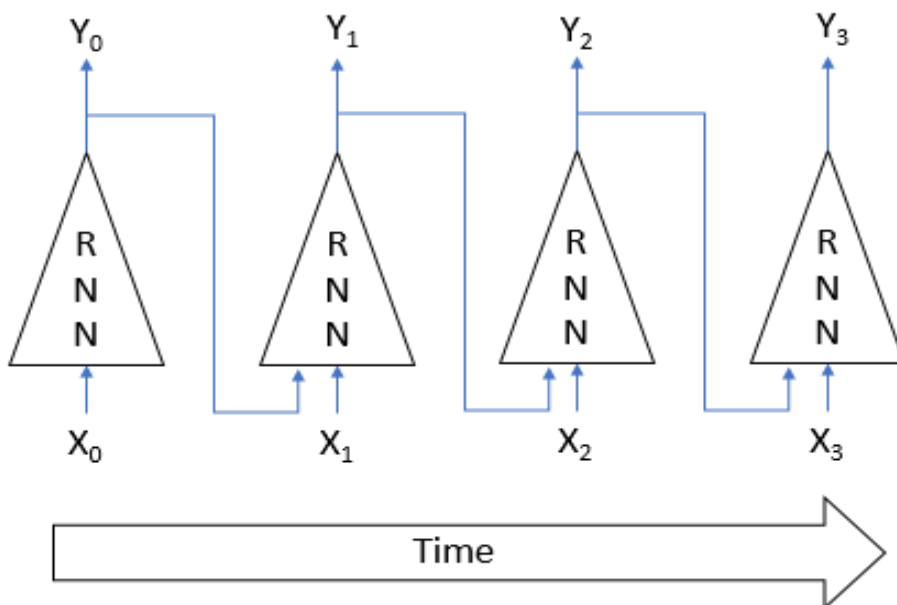
Algoritma dimulai dengan menggabungkan data input dengan informasi dari Hidden state menjadi satu larik data, yang kemudian diproses oleh empat lapisan saraf tersembunyi LSTM. Di antara ini, lapisan "Forget gate" menggunakan fungsi aktivasi sigmoid untuk menentukan elemen memori mana yang harus dipertahankan atau dibuang. Selanjutnya, lapisan "New Content" dan "Input gate" menentukan informasi baru yang akan ditambahkan ke memori, menggunakan fungsi tangen hiperbolik dan sigmoid secara berturut-turut. Memori yang diperbarui ini kemudian menginformasikan siklus iterasi berikutnya.

Akhirnya, generasi nilai output melibatkan proses serupa, dengan "Output gate" menormalisasi nilai memori saat ini, menghasilkan larik sinyal output. Sinyal ini tidak hanya keluar dari LSTM ke lingkungan eksternal tetapi juga masuk ke siklus iterasi berikutnya sebagai bagian dari aliran Hidden state, memastikan kontinuitas dan penyebaran pengetahuan sepanjang waktu.

2. METODE

Jaringan saraf berulang (RNN) dilatih menggunakan metode propagasi balik yang sudah dikenal sebelumnya. Mirip dengan pelatihan jaringan saraf konvolusional, proses berulang dalam waktu diuraikan menjadi perceptron multi-lapis. Setiap interval waktu dalam perceptron tersebut berfungsi sebagai lapisan tersembunyi. Namun, hanya satu matriks bobot yang digunakan untuk semua lapisan dalam perceptron tersebut. Oleh karena itu, untuk mengatur bobot, kita mengambil jumlah gradien untuk semua lapisan dan menghitung delta bobot sekali untuk gradien total dari semua lapisan.

Untuk membangun jaringan saraf berulang kami, kami akan menggunakan blok LSTM. Kami juga akan menambahkan tiga array untuk menyimpan data "memori", menggabungkan data input dan informasi status tersembunyi, serta gradien kesalahan data input. Nama dan fungsi dari metode kelas ini sesuai dengan yang telah dibahas sebelumnya. Namun, kode mereka memiliki beberapa perbedaan yang diperlukan untuk operasi algoritma. Mari kita bahas metode utama ini dengan lebih detail.



Gambar 3. Model RNN menggunakan blok LSTM

Metode inialisasi kelas menerima informasi dasar tentang blok yang sedang dibuat sebagai parameter. Dalam metode ini, pertama-tama kami memeriksa bahwa setidaknya satu neuron telah dibuat di setiap lapisan saraf dalam blok. Kemudian, kami memanggil metode yang sesuai dari kelas dasar. Setelah berhasil menyelesaikan metode tersebut, kami menginisialisasi lapisan tersembunyi blok, dan operasi yang diulang untuk setiap lapisan akan disediakan dalam metode terpisah yang disebut `InitLayer`. Setelah inialisasi lapisan saraf selesai, kami menginisialisasi array memori dengan nilai nol. Metode inialisasi lapisan saraf `InitLayer` menerima sebagai parameter referensi ke objek lapisan saraf yang diinisialisasi, jumlah neuron dalam lapisan, dan jumlah koneksi keluar.

2.1 Pemrosesan maju (*feed-forward*)

Pengiriman maju diimplementasikan dalam metode `feedForward`. Metode ini menerima referensi ke lapisan saraf sebelumnya sebagai parameter [15]. Di awal metode, kami memeriksa kevalidan pointer yang diterima dan memastikan adanya neuron dalam lapisan sebelumnya. Kami kemudian menghitung nilai-nilai dari gerbang-gerbang. Akhirnya, setelah semua perhitungan intermediet, kami menghitung array "memori" dan menentukan data output. Setelah mendapatkan gradien kesalahan, diperlukan untuk mengoreksi bobot semua lapisan saraf LSTM. Tugas ini diimplementasikan dalam metode `updateInputWeights`, yang menerima referensi ke lapisan saraf sebelumnya sebagai parameter. Penting untuk dicatat bahwa penyisipan referensi ke lapisan sebelumnya hanya dilakukan untuk menjaga struktur warisan. Kami mengatur loop bersarang untuk mengulang semua neuron dalam lapisan dan bobot mereka, sehingga memperbaiki matriks bobot.

Dalam lapisan konvolusi jaringan saraf, kita menciptakan 4 filter yang akan mencari pola dalam data total pembentukan candlestick dan pembacaan osilator pada candlestick yang sedang dianalisis. Jendela dan langkah filter akan sesuai dengan jumlah data per deskripsi candlestick. Dengan kata lain, ini akan membandingkan semua informasi tentang setiap candlestick dengan pola tertentu dan akan mengembalikan nilai konvergensi. Pendekatan ini memungkinkan untuk melengkapi data awal dengan informasi baru tentang candlestick (seperti menambahkan indikator tambahan untuk analisis, dan sebagainya) tanpa kehilangan kinerja yang signifikan.

Ukuran array fitur dikurangi dalam lapisan subsampling, dan hasilnya dilunakkan dengan rata-rata. EA itu sendiri memerlukan sedikit perubahan.

Perubahan ini terbatas pada kelas jaringan saraf, yaitu deklarasi variabel dan pembuatan instansi. Pengujian dilakukan menggunakan pasangan EURUSD dengan kerangka waktu H1. Dua Expert Advisor, satu dengan jaringan saraf konvolusi dan yang lainnya dengan jaringan yang sepenuhnya terhubung, diluncurkan secara bersamaan pada grafik yang berbeda dari simbol yang sama, dalam terminal yang sama. Parameter lapisan yang sepenuhnya terhubung dari jaringan saraf konvolusi sesuai dengan parameter jaringan yang sepenuhnya terhubung dari Expert Advisor kedua, yaitu kita hanya menambahkan lapisan konvolusi dan subsampling ke jaringan yang telah ada sebelumnya.

2.2 Dataset

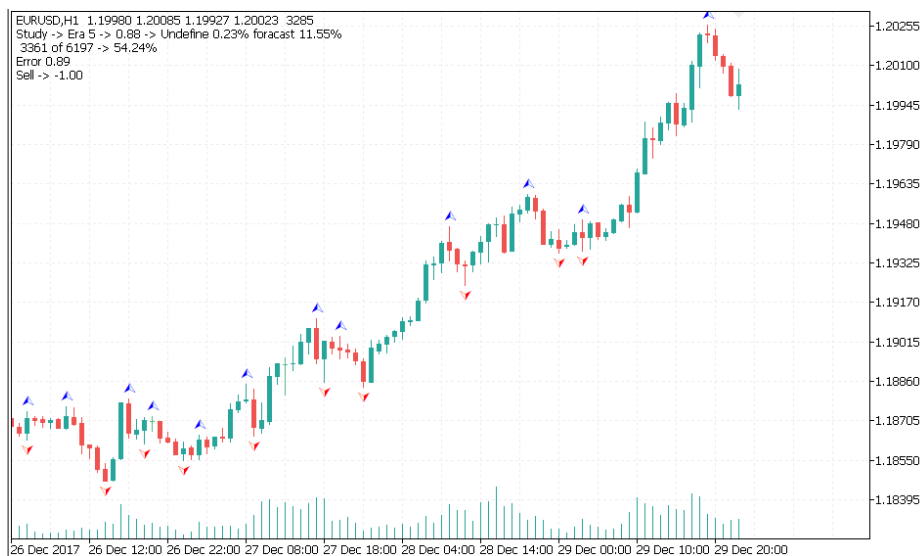
Dalam pembahasan berikut, kami mengarahkan fokus kami pada harga mata uang EUR/USD, dengan memanfaatkan data harian selama dua tahun terakhir, yaitu 2018 hingga 2019, sebagaimana ditunjukkan pada gambar 5. Tujuan kami adalah untuk memprediksi arah pergerakan mata uang yang menjadi fokus kami, menggunakan data historis sebagai dasarnya. Secara khusus, kami mempertimbangkan jendela waktu yang tipikal dengan data harga pembukaan, harga tertinggi, harga terendah, harga penutupan, dan volume (OHLCV) per jam.



Gambar 4. Grafik harga mata uang Euro – Dollar (EUR/USD)

3. HASIL DAN PEMBAHASAN

Pada bagian ini, kami menyajikan hasil komputasi terkait dengan proses pelatihan. Semua jaringan saraf tiruan (NN) dilatih menggunakan MQL, yaitu Bahasa Permintaan Meta untuk MetaTrader. Setiap jaringan dilatih sebanyak 134 epoch. Nilai tinggi ini dipilih karena hasil eksperimen menunjukkan bahwa pelatihan dengan jumlah epoch yang lebih sedikit menyebabkan jaringan yang mendalam menjadi overfit terhadap set pengujian dan hanya mempelajari distribusi. Selain itu, pelatihan dalam waktu yang lebih lama diperlukan untuk memahami tren konvergensi dengan lebih baik. Jika setelah beberapa waktu kesalahan entropi silang mulai meningkat, kita dapat memilih model yang memiliki kinerja terbaik. Sebagai algoritma optimisasi, kami menggunakan pendekatan Adam, lihat [9], dengan perhitungan gradien terkait yang dihitung dengan algoritma BPTT, sementara kami menggunakan perangkat keras GPU, yaitu Intel® UHD, untuk mengurangi biaya komputasi.

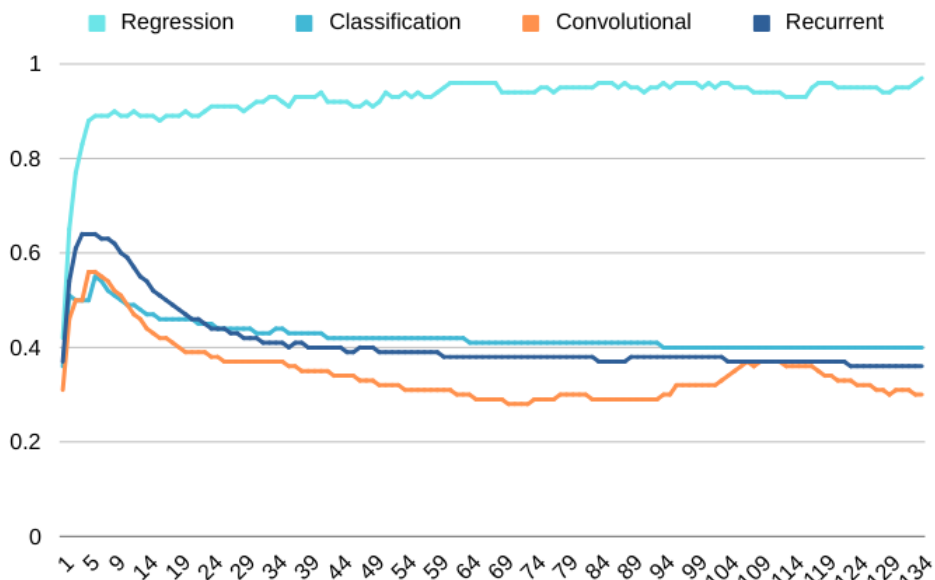


Gambar 5. Proses pelatihan agen untuk masing-masing model

Tujuan dari jaringan saraf konvolusional adalah untuk menguji operasinya. Tujuan utama dari jaringan saraf ini adalah untuk belajar memprediksi fraktal pada candlestick saat ini. Untuk tujuan ini, kita menyediakan informasi tentang pembentukan candlestick terakhir N dan data dari 4 osilator untuk periode yang sama ke dalam jaringan saraf.

Pengujian telah menunjukkan peningkatan kinerja yang kecil dalam jaringan saraf konvolusi. Meskipun ada penambahan dua lapisan, waktu pelatihan rata-rata untuk satu epoch (berdasarkan hasil 100 epoch) dari jaringan saraf konvolusi adalah 9 jam 35 menit, sedangkan untuk jaringan rekuren adalah 10 jam 45 menit.

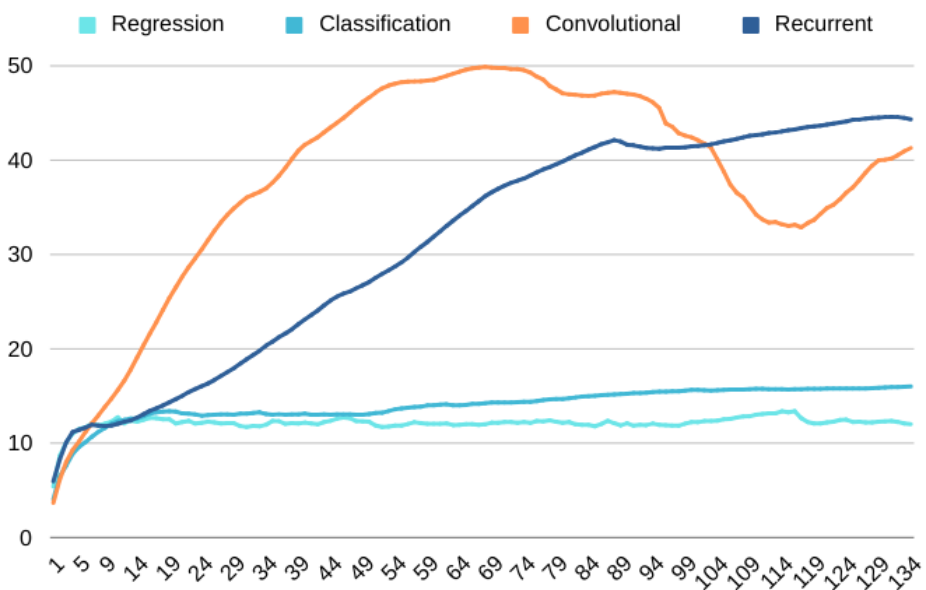
Tentu saja, penggunaan 4 lapisan saraf internal dalam setiap blok LSTM dan kompleksitas algoritma itu sendiri memengaruhi kinerja, dan akibatnya kecepatan jaringan saraf seperti itu sedikit lebih lambat daripada jaringan konvolusi yang sebelumnya dipertimbangkan. Namun, galat rata-rata kuadrat akar dari jaringan berulang jauh lebih rendah.



Gambar 6. Perbandingan error pelatihan untuk masing-masing model

Hasil pelatihan yang direpresentasikan dalam grafik menampilkan variasi tingkat kesalahan pada model-model jaringan syaraf yang diuji. Untuk model jaringan syaraf regresi, terlihat bahwa tingkat kesalahan awalnya rendah namun mengalami lonjakan signifikan setelah melewati epoch ke-9, dengan kesalahan yang tetap pada level tinggi mendekati 1, hal ini dapat mengindikasikan terjadinya overfitting atau adanya masalah ketidakstabilan selama proses pembelajaran. Sementara itu, model jaringan syaraf klasifikasi menunjukkan konsistensi kesalahan yang relatif stabil di sekitar nilai 0.4, tanpa fluktuasi yang berarti. Dalam kasus jaringan syaraf konvolusional, terdapat tren penurunan

kesalahan yang konsisten, walaupun sempat ada sedikit peningkatan kesalahan sebelum akhirnya mencapai titik stabil di sekitar nilai 0.3. Di sisi lain, model jaringan syaraf rekuren mengalami peningkatan kesalahan pada tahap awal, namun setelah itu kesalahan menurun dengan signifikan dan model mencapai kondisi stabil, di mana penambahan lebih lanjut pada jumlah pelatihan tidak secara signifikan meningkatkan kinerja model, dengan tingkat kesalahan akhir berada di rentang antara 0.4 dan 0.3.



Gambar 7. Perbandingan akurasi hasil pelatihan untuk masing-masing model

Dalam penelitian ini, analisis kinerja empat model pembelajaran mesin yang berbeda menunjukkan hasil yang beragam selama fase pelatihan. Model regresi memulai dengan skor yang rendah dan hanya menunjukkan peningkatan yang sangat kecil sepanjang pelatihan, menandakan bahwa model ini mengalami kesulitan untuk belajar atau memperbaiki diri dari data yang diberikan. Sebaliknya, model klasifikasi memulai dengan skor yang lebih tinggi dan mencatat peningkatan yang lebih tajam dan konsisten, mencapai skor yang stabil di akhir pelatihan, dengan jumlah akurasi tidak lebih dari 20%. Sedangkan untuk model konvolusional, yang juga memulai dengan skor yang rendah, menunjukkan peningkatan yang cepat dan mencapai puncak kinerjanya sebelum mengalami sedikit penurunan, namun kemudian kembali peningkatan akurasi. Sementara itu, model berulang (rekuren) menunjukkan pola yang mirip dengan model regresi tetapi dengan peningkatan yang sedikit lebih baik, menempatkannya di antara model regresi dan klasifikasi dalam hal kinerja yang

menandakan akurasi yang paling unggul di antara keempat model. Kesimpulannya, model rekuren menonjol sebagai model dengan kinerja terbaik, sedangkan model regresi menunjukkan peningkatan paling minim, dengan model klasifikasi dan berulang berada di posisi tengah.

4. KESIMPULAN

Penelitian ini menggali kemampuan machine learning, khususnya melalui penerapan neural network (NN) dan Recurrent Neural Network (RNN) termasuk model Long Short-Term Memory (LSTM), dalam konteks trading forex. Meskipun machine learning menjanjikan kecepatan dan akurasi dalam analisis data pasar, terdapat tantangan signifikan seperti risiko keuangan yang tinggi dan ketidakpastian model dalam memprediksi pasar yang fluktuatif. RNN, dengan kemampuan memproses data berurutan dan mempertahankan memori jangka pendek, bersama dengan LSTM yang dirancang untuk memelihara informasi penting sepanjang waktu, menawarkan solusi potensial terhadap keterbatasan NN dalam analisis data berurutan.

Tabel 1. Hasil akhir perbandingan dengan empat model neural network dengan 134 epoch

Test Result	Regression Neural Network	Classification Neural Network	Convolutional Neural Network	Recurrent Neural Network
<i>Root mean square error</i>	0.97	0.40	0.30	0.36
<i>Hit percentage</i>	12.03%	16.02%	41.29%	44.33%

Hasil eksperimental menunjukkan variasi dalam kinerja antara model-model yang berbeda selama pelatihan. Model regresi mengalami kesulitan dalam pembelajaran dengan tingkat kesalahan yang meningkat signifikan, sementara model klasifikasi menunjukkan konsistensi yang lebih stabil. Model jaringan syaraf konvolusional menunjukkan tren penurunan kesalahan yang konsisten, menandakan peningkatan kemampuan prediksi. Paling mencolok, RNN menunjukkan penurunan kesalahan yang signifikan setelah tahap awal, mencapai kondisi stabil dengan kinerja yang mengesankan. Hal ini menegaskan potensi RNN, khususnya LSTM, dalam meramalkan seri waktu pasar forex, menjadikannya alat yang berharga dalam trading forex yang memanfaatkan machine learning.

REFERENSI

- [1] N. Karnaukh, A. Ranaldo, and P. Söderlind, "Understanding FX Liquidity," *Rev. Financ. Stud.*, vol. 28, no. 11, pp. 3073–3108, 2015.
- [2] M. L. De Prado, *Advances in financial machine learning*. books.google.com, 2018.
- [3] T. J. Sejnowski, *The deep learning revolution*. books.google.com, 2018.
- [4] L. Di Persio and M. Frigo, "Maximum likelihood approach to Markov switching models," *WSEAS Trans. Bus. Econ.*, vol. 12, pp. 239–242, 2015.
- [5] L. Di Persio and M. Frigo, "Gibbs sampling approach to regime switching analysis of financial time series," *J. Comput. Appl. Math.*, vol. 300, pp. 43–55, 2016.
- [6] L. Di Persio, O. Honchar, L. Di Persio, and O. Honchar, "Artificial neural networks architectures for stock price prediction: Comparisons and applications," *Int. J. circuits, Syst. signal Process.*, vol. 10, 2016.
- [7] L. DI PERSIO and C. Benazzoli, "Default contagion in financial networks," *Int. J. Math. Comput. Simul.*, vol. 10, pp. 112–117, 2016.
- [8] E. Hurwitz and T. Marwala, "State of the Art Review for Applying Computational Intelligence and Machine Learning Techniques to Portfolio Optimisation," *arXiv Prepr. arXiv0910.2276*, 2009.
- [9] F. Butaru, Q. Chen, B. Clark, S. Das, A. W. Lo, and A. Siddique, "Risk and risk management in the credit card industry," *J. Bank. Financ.*, vol. 72, pp. 218–239, 2016.
- [10] M. P. Naeini, H. Taremiyan, and H. B. Hashemi, "Stock market value prediction using neural networks," in *2010 international conference on computer information systems and industrial management applications (CISIM)*, 2010, pp. 132–136.
- [11] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Twenty-fourth international joint conference on artificial intelligence*, 2015.
- [12] J. Wang and J. Wang, "Forecasting energy market indices with recurrent neural networks: Case study of crude oil price fluctuations," *Energy*, 2016.
- [13] S. Selvin, R. Vinayakumar, "Stock price prediction using LSTM, RNN and CNN-sliding window model," *Adv.*, 2017.
- [14] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [15] A. Oztekin, R. Kizilaslan, S. Freund, and A. Iseri, "A data analytic approach to forecasting daily stock returns in an emerging market," *Eur. J. Oper.*, 2016.