

## Artikel Penelitian

Marwan Gebran\*, Kathleen Connick, Hikmat Farhat, Frédéric Paletou, dan Ian Bentley

# Aplikasi pembelajaran mendalam untuk penentuan parameter bintang: I-membatasi hiperparameter

<https://doi.org/10.1515/astro-2022-0007>

diterima 22 November 2021; diterima 12 Januari 2022

**Abstrak:** Pembelajaran mesin adalah metode yang efisien untuk menganalisis dan menginterpretasikan data astro- nomis yang semakin banyak tersedia. Dalam penelitian ini, kami menunjukkan pendekatan pedagogis yang seharusnya bermanfaat bagi siapa pun yang ingin bereksperimen dengan teknik pembelajaran mendalam dalam teks penentuan parameter bintang. Dengan menggunakan arsitektur jaringan syaraf tiruan konvergen, kami memberikan gambaran langkah demi langkah tentang cara memilih parameter optimal untuk mendapatkan nilai paling akurat untuk para- meter bintang:  $T_{\text{eff}}$ ,  $\log g$ ,  $[M/H]$ , dan  $v_e$  di sini. Spektrum sintetis dengan derau acak digunakan untuk membatasi metode ini dan untuk meniru hasil pengamatan. Kami menemukan bahwa setiap parameter bintang membutuhkan kombinasi yang berbeda Kombinasi hiperparameter jaringan dan akurasi maksimum yang dicapai tergantung pada kombinasi ini serta rasio sinyal terhadap derau pengamatan, dan arsitektur jaringan. Kami juga menunjukkan bahwa teknik ini dapat diterapkan pada jenis spektral lain dalam rentang panjang gelombang yang berbeda setelah teknik ini dioptimalkan.

**Kata-kata kunci:** metode: analisis data, metode: statistik, metode: pembelajaran mendalam, teknik: spektroskopi, bintang: parameter fundamental



\* Penulis korespondensi: Marwan Gebran, Departemen Kimia dan Fisika, Saint Mary's College, Notre Dame, IN 46556, Serikat, e-mail: [mgebran@saintmarys.edu](mailto:mgebran@saintmarys.edu)  
 Kathleen Connick, Ian Bentley: Departemen Kimia dan Fisika, Saint Mary's College, Notre Dame, IN 46556, Amerika Serikat  
 Hikmat Farhat: Departemen Ilmu Komputer, Universitas Notre Dame-Louaize, PO Box 72, Zouk Mikael, Lebanon  
 Frédéric Paletou: Université de Toulouse, Observatoire Midi-Pyrénées, Irap, Cnrs, Cnes, 14 av. E. Belin, F-31400 Toulouse, Prancis

## 1 Pendahuluan

Aplikasi pembelajaran mesin (machine learning/ML) telah digunakan secara ekstensif dalam astronomi selama dekade terakhir (Baron 2019). Hal ini terutama disebabkan oleh sejumlah besar data yang diperoleh dari ruang angkasa dan observatorium berbasis darat. Oleh karena itu, ada kebutuhan untuk menganalisis data ini secara otomatis. Pendekatan statistik, reduksi dimensi, dekomposisi wavelet, ML, dan deep learning (DL) merupakan beberapa contoh upaya yang telah dilakukan untuk mendapatkan parameter bintang yang lebih akurat seperti temperatur efektif ( $T_{\text{eff}}$ ), gravitasi permukaan ( $\log g$ ), kecepatan rotasi ekuator yang diproyeksikan ( $v_e \sin i$ ), dan metalitas ( $[M/H]$ ) dengan menggunakan spektrum bintang pada rentang panjang gelombang yang berbeda (Guiglion *et al.* 2020, Passegger *dkk.* 2020, Portillo *dkk.* 2020, Wang *dkk.* , Wang *dkk.* 2020, Zhang *dkk.* , Bai *dkk.* 2019, Kassounian *dkk.* 2019, Fabbro *dkk.* ). 2018, Gill *dkk.* 2018, Li *dkk.* 2017, Gebran *dkk.* 2016, Paletou *dkk.* 2015a, b). DL adalah metode ML yang didasarkan pada jaringan syaraf tiruan (JST) yang biasanya tidak membutuhkan algoritma statistik khusus untuk memprediksi solusi, melainkan dipelajari pengalaman dan dengan demikian membutuhkan dataset yang sangat besar (Zhu *dkk.* 2016) untuk pelatihan agar dapat berkinerja dengan baik.

Gambaran umum tentang teknik otomatis yang digunakan dalam penentuan parameter bintang dapat ditemukan dalam studi Kassounian *dkk.* (2019). Kami akan menyebutkan beberapa penelitian terbaru yang melibatkan ML/DL. Meningkatnya daya komputasi dan ketersediaan paket ML yang dioptimalisasi (seperti Python, C++ , dan R) memungkinkan para astronom untuk beralih dari teknik klasik ke ML ketika menggunakan data yang besar. Salah satu percobaan pertama untuk mendapatkan parameter bintang dengan menggunakan jaringan syaraf tiruan dilakukan oleh Bailer-Jones (1997). Hasil penelitian ini menunjukkan bahwa jaringan saraf dapat memberikan klasifikasi tipe spektral yang akurat pada rentang tipe spektral B2 - M7.

Dafonte *dkk.* (2016) mempresentasikan arsitektur ANN yang mempelajari fungsi yang dapat menghubungkan para meter bintang dengan spektrum masukan. Mereka memperoleh residu dalam

penurunan metalitas di bawah 0,1 dex untuk bintang-bintang dengan magnitudo Gaia  $G_{\text{rvs}} < 12$  mag, yang menyumbang jumlah sekitar empat juta bintang yang akan diamati oleh spektrograf kecepatan radial satelit.<sup>1</sup> GaiaRamírez Vélez *dkk.* (2018) menggunakan algoritma ML untuk mengukur medan magnet longitudinal rata-rata pada bintang-bintang dari spektrum terpolarisasi dengan resolusi tinggi. Mereka menemukan peningkatan hasil yang cukup besar, sehingga memungkinkan kita untuk mengestimasi kesalahan yang terkait dengan pengukuran medan magnet bintang pada tingkat kebisingan yang berbeda.

Parks *dkk.* (2018) mengembangkan dan menerapkan arsitektur convolutional neural network (CNN) dengan menggunakan pembelajaran multi-tugas untuk mencari dan mengkarakterisasi serapan HI Ly yang kuat pada spektrum kuasar. Fabbro *dkk.* (2018) menerapkan arsitektur jaringan syaraf tiruan untuk menganalisis SDSS-III APOGEE DR13 dan spektrum bintang sintetis. Hasil penelitian ini menunjukkan bahwa parameter bintang ditentukan dengan presisi dan akurasi yang sama dengan pipeline APOGEE.

Sharma *dkk.* (2020) memperkenalkan pendekatan otomatis untuk klasifikasi spektrum bintang di wilayah optik menggunakan CNN. Mereka juga menunjukkan bahwa metode DL dengan jumlah lapisan yang lebih besar memungkinkan penggunaan detail yang lebih halus dalam spektrum, yang menghasilkan akurasi yang lebih baik dan generasi yang lebih baik dibandingkan dengan teknik ML tradisional.

Wang *dkk.* (2020) memperkenalkan metode DL, SPCANet, yang menghasilkan  $T_{\text{eff}}$  dan  $\log g$  serta 13 kelimpahan kimiawi untuk data Survei Resolusi Menengah LAMOST. Para penulis ini menemukan presisi kelimpahan hingga 0,19 dex untuk spektrum dengan rasio signal-to-noise (SNR) hingga  $\sim 10$ . Hasil dari SPCANet konsisten dengan hasil dari survei lainnya, seperti APOGEE, GALAH, dan RAVE, dan juga divalidasi dengan nilai literatur sebelumnya termasuk gugus dan bintang medan. Guiglion *dkk.* (2020) memperoleh parameter atmosfer dan kelimpahan spesies yang berbeda untuk 420165 spektrum RAVE. Mereka menunjukkan bahwa metode berbasis CNN memberikan cara yang ampuh untuk menggabungkan data spektroskopi, fotometri, dan astrometri tanpa perlu menerapkan pendahuluan apa pun dalam bentuk model evolusi bintang.

Baru-baru ini, Landa dan Reuveni (2021) memperkenalkan CNN multi-layer untuk meramalkan probabilitas kejadian flare matahari kelas M dan X. Chen *dkk.* (2021) memperkenalkan metode pengenalan AGN berdasarkan jaringan syaraf tiruan. Almeida *dkk.* (2021) menggunakan metode ML untuk menghasilkan model distribusi entri khusus (SED) dan mencocokkan pengamatan yang jarang dari inti galaksi aktif dengan luminositas rendah. Rhea *dkk.* (2020), Rhea dan Rousseau-Nepton (2021) menggunakan CNN

dan JST yang berbeda untuk mengestimasi parameter garis emisi dan rasio garis yang ada pada berbagai filter spektrofotometer SIELLE. Curran *dkk.* (2021) menggunakan DL yang dikombinasikan dengan algoritme k-Nearest Neighbour dan Decision Tree Regression untuk membandingkan keakuratan pergeseran fotometrik yang diprediksi dari sumber yang baru terdeteksi. Ofman *dkk.* (2022) menerapkan algoritma Kecerdasan Buatan *ThetaRay* pada 10.803 kurva cahaya dari peristiwa pelintasan ambang batas dan menemukan 39 target kandidat eksoplanet baru. Bickley *dkk.* (2021) mencapai akurasi klasifikasi 88% saat menyelidiki penggunaan CNN untuk klasifikasi penggabungan otomatis. Gafeira *dkk.* (2021) menggunakan teknik inversi berbantuan berdasarkan CNN untuk inversi profil Stokes matahari. Dalam konteks klasifikasi morfologi galaksi, Gan *dkk.* (2021) menggunakan jaringan adversarial generatif ML untuk mengubah gambar kabur Teleskop Subaru berbasis darat menjadi gambar Teleskop Luar Angkasa Hubble semu. Garraffo *dkk.* (2021) mempresentasikan StelNet, sebuah jaringan syaraf tiruan yang dilatih berdasarkan jejak evolusi bintang yang dengan cepat dan akurat memprediksi massa dan usia dari luminositas absolut dan temperatur efektif untuk bintang-bintang yang memiliki metalitas matahari.

Dalam naskah ini, kami menyajikan metode baru untuk memperoleh parameter atmosfer bintang, dan kami juga menunjukkan pengaruh masing-masing parameter CNN (seperti pilihan pengoptimal, fungsi rugi-rugi, dan fungsi aktivasi) terhadap keakuratan hasil. Kami akan memberikan prosedur yang dapat diikuti untuk menemukan konfigurasi yang paling tepat, terlepas dari arsitektur CNN. Hal ini dimaksudkan sebagai makalah pertama dari serangkaian makalah yang akan membantu komunitas astronomi untuk memahami efek pada akurasi prediksi dari sebagian besar parameter dan arsitektur jaringan. Parameter CNN sangat banyak dan untuk menemukan parameter yang optimal adalah tugas yang sangat sulit. Untuk melakukannya, kami melatih CNN dengan konfigurasi parameter yang berbeda menggunakan spektrum sintetis murni untuk tiga langkah pelatihan, validasi silang (selanjutnya disebut validasi), dan pengujian. Dengan menggunakan spektrum sintetis, kami memiliki akses ke parameter yang sebenarnya selama pengujian. Spektrum bising diuji untuk meniru pengamatan.

Kami telah membatasi pekerjaan kami pada jenis objek tertentu, Bintang A, karena seperti yang telah disebutkan sebelumnya, tujuannya bukan untuk menunjukkan seberapa baik kita dapat menurunkan parameter bintang yang dilabeli, tapi apa pengaruh parameter tertentu pada analisis spektrum bintang. Dengan menerapkan model kami pada bintang A, kami menggunakan hasil sebelumnya (Gebran *dkk.* 2016, Kassou- nian *dkk.* 2019) sebagai referensi untuk keakuratan yang diharapkan dari parameter bintang yang diturunkan. Dengan cara yang sama, rentang panjang gelombang dan daya penyelesaian dipilih untuk mewakili nilai-nilai yang digunakan oleh sebagian besar instrumen yang tersedia. Setelah kalibrasi hiperparameter dilakukan

□ □

<sup>1</sup> Besaran terbatas spektrometer kecepatan radial (RVS) adalah sekitar 15,5 mag (Cropper *et al.* 2014).

dilakukan, kami telah menguji konfigurasi jaringan optimal kami pada satu set bintang FGK di Bagian 6, menggunakan rentang panjang gelombang Paletou *dkk.* (2015a).

Data pelatihan, validasi, dan pengujian dijelaskan di Bagian 2. Bagian 3 membahas persiapan data sebelum pelatihan. Konstruksi jaringan syaraf dan pemilihan parameter dijelaskan di Bagian 4. Hasil dirangkum dalam Bagian 5. Penerapan jaringan optimal pada bintang FGK dilakukan pada Bagian

6. Diskusi dan kesimpulan dikumpulkan di Bagian 7.

## 2 Spektrum pelatihan

Basis data pembelajaran atau pelatihan kami (TDB) dibangun dari spektrum sintetis untuk bintang-bintang yang memiliki temperatur efektif.

antara 7.000 dan 10.000 K, dan rentang panjang gelombang 4.450-5.000. Rentang ini dipilih karena berada dalam domain tampak dan mengandung garis-garis logam dan Balmer yang peka terhadap semua parameter bintang ( $T_{\text{eff}}$ ,  $\log g$ ,  $[M/H]$ ,  $v_e$  di *sini*), terutama untuk jenis spektrum yang dipilih dalam penelitian ini. Wilayah ini juga tidak sensitif terhadap kecepatan mikroturbulen yang diadopsi menjadi  $\xi_t = 2$  km/s berdasarkan penelitian Gebran *dkk.* (2016, 2014). Gravitasi permukaan,  $\log g$ , dipilih dalam kisaran 2,0-5,0 dex. Kecepatan rotasi yang diproyeksikan,  $v_e$  di *sini*, dihitung antara 0 dan 300 km s<sup>-1</sup>. Metalitas,  $[M/H]$ , berada dalam kisaran

-1.5 dan+ 1.5 dex. Tabel 1 menampilkan kisaran semua parameter bintang. Spektrum ini digunakan untuk pelatihan dan fase validasi. Sekitar 55.000 spektrum sintetis bebas derau dihitung dengan menggunakan pilihan acak dari parameter bintang dalam kisaran Tabel 1. Spektrum ini digunakan sebagai pengganti pengamatan (data uji tanpa derau). SNR Gaussian, berkisar antara 5 dan 300, ditambahkan ke spektrum uji ini untuk memeriksa keakuratan teknik ini pada data berderau (data uji dengan derau).

Detail untuk perhitungan sintetis dapat ditemukan dalam studi Gebran *dkk.* (2016) atau Kasounian *dkk.* (2019). Singkatnya, model paralel bidang 1D

Atmosfer dihitung dengan menggunakan ATLAS9 (Kurucz 1992). Model-model ini berada dalam kesetimbangan termodinamika lokal (LTE) dan dalam kesetimbangan hidrostatik dan radiatif. Kami telah menggunakan fungsi distribusi opasitas yang baru dalam perhitungan (Castelli dan Kurucz 2003) serta pencampuran parameter panjang 0,5 untuk 7.000 K  $\leq T_{\text{eff}} \leq 8.500$  K, dan 1,25 untuk  $T_{\text{eff}} \leq 7.000$  K (Smalley 2004).

Kami telah menggunakan Hubeny dan Lanz (1992) SYNSPEC48 syn-

kode spektrum teik untuk menghitung semua spektrum yang dinormalisasi. Daftar garis yang diadopsi dirinci dalam studi Gebran *dkk.* (2016). Daftar ini terutama disusun dengan menggunakan data dari Kurucz *ghyperall.dat*<sup>2</sup>, VALD<sup>3</sup>, dan basis data NIST<sup>4</sup>.

Terakhir, resolving power disimulasikan ke  $R = 60.000$ .

Nilai ini berada dalam kisaran antara spektograf resolusi rendah dan tinggi. Teknik yang akan ditunjukkan pada bagian berikutnya dapat digunakan untuk resolusi apa pun. Konstruksi dan ukuran TDB akan dibahas di Bagian 5. Penggunaan spektrum sintetis dalam ML untuk membatasi parameter bintang telah terbukti mengalami apa yang disebut celah sintetis (Fabbro *et al.* 2018, Passetto *et al.* 2020). Kesenjangan ini merujuk pada perbedaan distribusi fitur antara data sintetis dan data yang diamati. Kami telah memutuskan untuk membatasi pekerjaan kami pada data sintetis karena dua alasan: pertama, kami ingin menghilangkan kerumitan langkah-langkah persiapan data (reduksi data, kalibrasi fluks, normalisasi fluks, koreksi kecepatan radial, dan seterusnya), dan kedua, karena tujuan kami adalah untuk menemukan strategi dan teknik yang harus diadopsi dalam ML untuk menurunkan parameter bintang. Kami sedang mengerjakan makalah mendatang yang berhubungan dengan arsitektur jaringan serta pilihan ukuran kernel dan jumlah neuron. Menggabungkan strategi terbaik untuk membatasi hiperparameter (naskah ini) serta arsitektur yang paling optimal (penelitian mendatang) akan memungkinkan kami untuk menggunakan kombinasi data sintetis dan data observasi dalam basis data pelatihan kami. Dengan parameter bintang yang sudah diketahui, data pengamatan ini akan memungkinkan kita untuk menghilangkan/meminimalkan kesenjangan sintetis.

dan membatasi parameter bintang dengan lebih baik.

Tabel 1: Tabel 1: Rentang parameter digunakan untuk perhitungan TDB spektrum sintetis

Parameter	Jangkauan
$T_{\text{eff}}$ (K)	[ 7,000, 11,000 ]
$\log g$ (dex)	[2.0, 5.0]
$[M / H]$ (dex)	[-1.5, 1.5]
$v_e$ di <i>sini</i> ( km s <sup>-1</sup> )	[0, 300]
$\lambda / \Delta \lambda$	60,000

## 3 Persiapan data

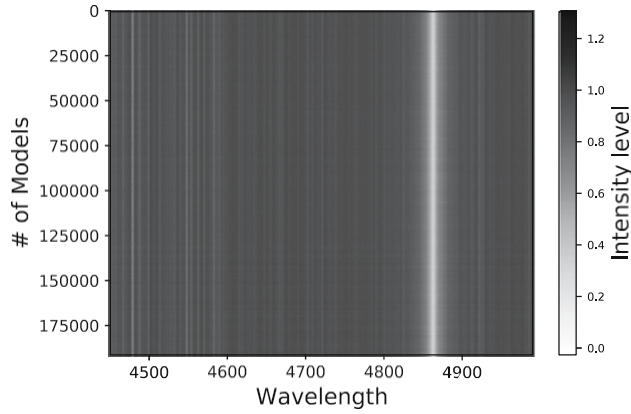
TDB berisi  $N_{\text{(spektrum)}}$  spektrum yang menjangkau gelombang Rentang panjang 4.450-5.000. Memiliki langkah panjang gelombang 0,05, hal ini menghasilkan  $N_{\text{fit}} = .800$  titik fluks per spektrum. TDB kemudian dapat diwakili oleh

□ □

2 <http://kurucz.harvard.edu>.

3 <http://www.astro.uu.se/vald/php/vald.php>.

4 <http://physics.nist.gov>.



Gambar 1: Peta warna yang merepresentasikan fluks untuk sampel basis data pelatihan menggunakan augmentasi data. Panjang gelombang dalam.

matriks  $\mathbf{M}$  dengan ukuran  $N_{\text{spektrum}} \times N_{fi}$ . Peta warna dari sub-sampel  $\mathbf{M}$  ditampilkan pada Gambar 1. Meskipun proses sin spektrum optik dinormalisasi, beberapa titik panjang gelombang dapat memiliki fluks yang lebih besar dari satu. Hal ini disebabkan oleh noise yang dimasukkan selama prosedur *augmentasi data*, yang akan dijelaskan di Bagian 4.1.1.

Melatih CNN menggunakan matriks  $\mathbf{M}$  membutuhkan waktu yang lama.

penjumlahan, terutama jika harus menggunakan rentang panjang gelombang yang lebih besar atau resolusi yang lebih tinggi. Oleh karena itu, kami telah menerapkan teknik pengurangan dimensi, yaitu analisis komponen utama (PCA), untuk mengurangi ukuran TDB pelatihan serta ukuran data sintetis validasi, pengujian, dan derau. Meskipun langkah ini bersifat opsional, kami merekomendasikan penggunaannya setiap kali data dapat diwakili oleh sejumlah kecil koefisien. PCA dapat mengurangi ukuran setiap spektrum dari  $N_{fi}$  menjadi  $n_k$ . Pilihan  $n_k$  tergantung pada banyak parameter, ukuran database, rentang panjang gelombang, dan bentuk garis spektrum. Sebagai langkah pertama, kita perlu menemukan komponen utama, dan untuk melakukannya, kita lanjutkan sebagai berikut.

Matriks  $\mathbf{M}$  dirata-ratakan di sepanjang sumbu  $N_{\text{spektrum}}$  dan hasilnya disimpan dalam sebuah vektor  $\bar{\mathbf{M}}$ . Kemudian, kita menghitung vektor eigen  $\mathbf{e}_k(fi)$  dari matriks varians-kovarians  $\mathbf{C}$  yang didefinisikan sebagai

$$\mathbf{C} = (\mathbf{M} - \bar{\mathbf{M}})^T (\mathbf{M} - \bar{\mathbf{M}}), \quad (1)$$

di mana superskrip "T" adalah singkatan dari transposisi operator.  $\mathbf{C}$  memiliki dimensi  $N_{fi} \times N_{fi}$ . Mengurutkan Vektor eigen dari matriks varians-kovarians yang semakin kecil akan menghasilkan "komponen utama". Setiap spektrum  $\mathbf{M}$  kemudian diproyeksikan pada komponen-komponen utama ini untuk menemukan koefisien yang sesuai,  $p_{(jk)}$ , yang didefinisikan sebagai

$$p_{jk} = (\mathbf{M}_j - \bar{\mathbf{M}}) \cdot \mathbf{e}_k. \quad (2)$$

Pemilihan jumlah koefisien diatur oleh kesalahan yang direkonstruksi sebagaimana dirinci dalam studi Paletou *dkk.* (2015a):

$$E(k)_{\text{maks}} = \left\langle \left| \frac{\mathbf{M}_j - \sum_{k=1}^{(k)_{\text{maks}}} p_{jk} \mathbf{e}_k - \bar{\mathbf{M}}}{\bar{\mathbf{M}}} \right| \right\rangle. \quad (3)$$

Kami memilih nilai  $n_k$  yang mengurangi rata-rata kesalahan rekonstruksi hingga mencapai nilai  $< 0,5\%$ . Sebagai contoh, dengan menggunakan basis data 25.000 spektrum dengan para

meter yang berkisar secara acak antara nilai pada Tabel 1 membutuhkan kurang dari tujuh koefisien untuk mencapai akurasi

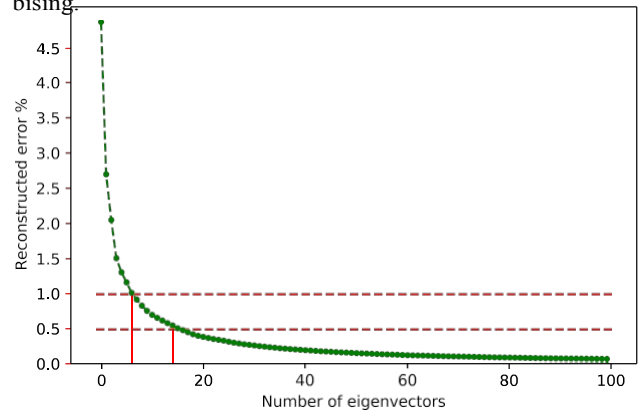
$< 1\%$ , dan nilai  $n_k = 17$  untuk mencapai kesalahan  $0,5\%$  sebagai ditunjukkan pada Gambar 2. Teknik ini telah menunjukkan keefektifannya.

ketika diterapkan pada data pengamatan sintetis dan/atau data pengamatan nyata dengan  $T_{\text{eff}} > 4.000$  K (lihat Gebran *dkk.* 2016, Paletou *dkk.* 2015a, b, untuk lebih jelasnya).

Menerapkan prosedur yang sama untuk semua TDB dan Dengan mengambil nilai maksimum yang akan digunakan untuk semua, kami telah mengadopsi nilai konstan untuk  $n_k = 50$ . Nilai ini memperhitungkan semua basis data yang akan ditangani dalam

pekerjaan ini, terutama bahwa beberapa data akan ditambah sebagai akan dijelaskan di Bagian 4.1.1. Ini berarti bahwa alih-alih melatih sebuah matriks yang memiliki dimensi  $N_{\text{spektrum}} \times N_{fi}$ , kita menggunakan matriks dengan dimensi  $N_{\text{spektrum}} \times n_k$ , dengan  $n_k \ll N_{fi}$ . Dalam hal ini, data baru kita terdiri dari sebuah matriks yang berisi koefisien yang dihitung dengan proyeksi spektrum pada  $n_k$  vektor eigen.

Prosedur proyeksi terhadap komponent utama ini kemudian diterapkan pada set data validasi, uji, dan spektrum bising.



Gambar 2: Rata-rata kesalahan yang direkonstruksi sebagai fungsi dari jumlah

komponen utama yang digunakan untuk proyeksi. Garis putus-putus masing-masing mewakili kesalahan 1 dan 0,5%. Untuk  $n_k > 17$ , spektrum dapat direkonstruksi dengan akurasi lebih dari 99,5%.



## 4 DL: ANN

Bagian ini dimulai dengan penjelasan singkat tentang pembelajaran yang diawasi (supervised<sup>5</sup> learning). Diberikan satu set data  $(X, Y)$ , tujuannya adalah untuk menemukan fungsi  $f$  sedemikian rupa sehingga  $f(X)$  sedekat mungkin dengan  $Y$ . Sebagai contoh, dapat berupa suhu efektif atau gravitasi permukaan dan  $X$  adalah spektrum yang sesuai. Ini adalah "pendekatan".

seness" biasanya diukur dengan mendefinisikan *fungsi kerugian*  $L(f(X), Y)$  yang mengukur perbedaan antara nilai yang telah ditentukan sebelumnya dan nilai aktual. Oleh karena itu, tujuan dari

proses pembelajaran adalah menemukan  $f$  yang meminimalkan  $L$  untuk kumpulan data yang diberikan  $(X, Y)$ . Pada akhirnya, keberhasilan metode pembelajaran dinilai dari seberapa baik metode tersebut dapat menggeneralisasi. Dengan kata lain, setelah  $f$  yang optimal ditemukan untuk set pelatihan

$(X, Y)$ , dan diberikan kumpulan data lain  $(U, V)$ , seberapa dekat

$f(U)$  ke  $V$ .

Salah satu metode yang paling sukses dalam menangani masalah seperti ini adalah ANN, sebuah subset dari ML. Seperti namanya, JST adalah sekumpulan blok bangunan yang terhubung yang disebut neuron yang dimaksudkan untuk meniru operasi neuron biologis (Anthony dan Bartlett 1999, Wang 2003). Berbagai jenis JST dapat dibangun dengan memvariasikan jumlah koneksi antara dan operasi masing-masing neuron. Operasi yang dilakukan oleh neuron-neuron ini bergantung pada sejumlah parameter yang disebut *bobot* dan beberapa fungsi non-linear yang disebut *aktivasi*. Pada tingkat tinggi, JST hanyalah fungsi  $f$  yang telah dijelaskan sebelumnya. Karena arsitektur jaringan dipilih di awal, menemukan  $f$  yang optimal bermuara pada menemukan bobot optimal yang meminimalkan fungsi biaya  $L$ .

Terlepas dari jenis yang digunakan, proses menemukan bobot optimal kurang lebih sama, dan bekerja sebagai berikut. Setelah arsitektur jaringan dipilih, bobot diinisialisasi, kemudian varian gradient descent diterapkan pada data pelatihan. Gradient descent mengubah parameter secara iteratif, dengan kecepatan tertentu yang sebanding dengan gradiennya, hingga nilai kerugiannya cukup kecil (Ruder 2016). Konstanta proporsionalitas disebut dengan *learning rate*. Meskipun proses ini sudah diketahui dengan baik, sampai saat ini belum ada resep yang jelas untuk pemilihan komponen yang berbeda. Kesulitan utama muncul dari fakta bahwa fungsi kerugian mengandung beberapa minima dengan sifat generalisasi yang berbeda. Dengan kata lain, tidak semua minima dari fungsi kerugian sama dalam hal generalisasi. Nilai minimum yang dicapai pada akhir fase pelatihan tergantung pada nilai awal yang dipilih untuk bobot, algoritma optimasi yang digunakan,

termasuk tingkat pembelajaran dan dataset pelatihan (Zhang *et al.* 2016). Dengan tidak adanya *preskripsi* teoretis yang jelas untuk komponen-komponen tersebut, kita harus mengandalkan pengalaman dan praktik terbaik (Bengio 2012).

Salah satu jenis JST yang populer adalah jaringan feedforward, di mana neuron disusun dalam beberapa lapisan, dengan output dari setiap lapisan terhubung *sepenuhnya* dengan input dari lapisan berikutnya. Dengan meningkatkan jumlah lapisan (asal kata "deep" dalam "DL"), banyak jenis data dapat dimodelkan dengan tingkat akurasi yang tinggi. Namun, JST yang terhubung penuh memiliki beberapa kekurangan, seperti jumlah parameter yang banyak, konvergensi yang lambat, overfitting, dan yang paling penting adalah kegagalan untuk mendeteksi pola *lokal*. Hampir semua kekurangan yang disebutkan di atas diatasi dengan menggunakan lapisan konvolusi.

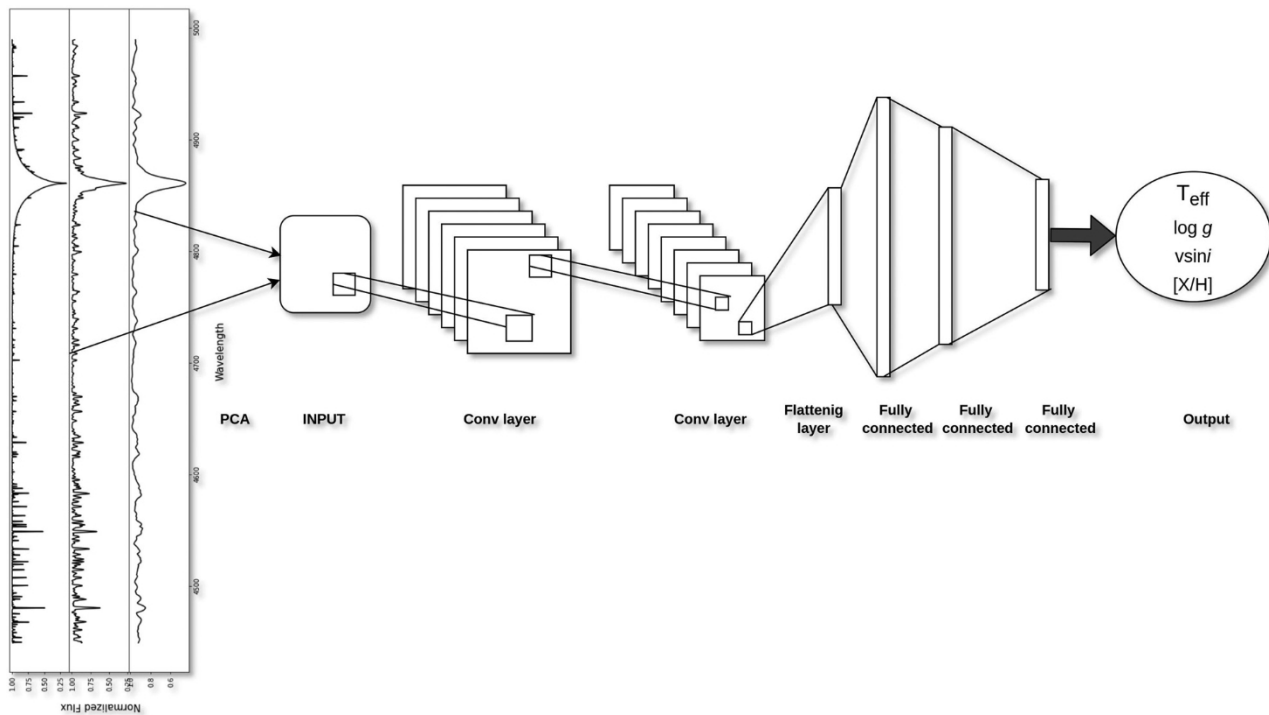
### 4.1 CNN

CNN adalah jaringan *multi-layer* di mana setidaknya satu dari *lapisannya* adalah lapisan konvolusi LeCun (1989). Seperti namanya, output dari lapisan konvolusi adalah hasil dari operasi konvolusi, bukan perkalian matriks, seperti pada lapisan feedforward, pada inputnya. Biasanya, operasi konvolusi ini dilakukan melalui serangkaian filter. CNN telah sangat berhasil dalam tugas pengenalan gambar (Yim *et al.* 2015). Pada umumnya, CNN digunakan bersama dengan pooling layer. Dalam penelitian ini, karena input ke CNN telah diproses dengan PCA untuk mengurangi dimensi basis data pelatihan, kami memutuskan untuk tidak menggunakan pooling layer dalam penelitian kami. Meskipun CNN sebagian besar digunakan untuk memproses data gambar, yang dapat dilihat sebagai data grid 2D, CNN juga dapat digunakan untuk data 1D.

Arsitektur CNN berbeda di antara berbagai *studi*. Tidak ada model yang sempurna, semuanya tergantung pada jenis dan ukuran data input, dan pada jenis parameter yang ditentukan sebelumnya. Dalam pekerjaan ini, kami tidak akan membatasi arsitektur model, tetapi kami akan memberikan strategi terbaik untuk membatasi parameter model untuk arsitektur yang spesifik dan terdefinisi. Gambar 3 menunjukkan diagram alir dari sebuah CNN. Tabel 2 menunjukkan lapisan yang berbeda, bentuk keluaran untuk setiap lapisan, dan jumlah parameter yang digunakan dalam model kami. Pada tabel yang sama, "Conv" adalah singkatan dari convolutional layer, "Flat" untuk lapisan perataan yang mengubah matriks data menjadi satu dimensi, dan "Full" adalah singkatan dari lapisan yang terhubung penuh. Jumlah total parameter yang harus dilatih setiap iterasi adalah 764.357. Pemilihan arsitektur didasarkan pada prosedur trial and error yang

□ □

<sup>5</sup> Pembelajaran terawasi mengacu pada algoritme yang menghitung model prediktif menggunakan titik data dengan label/hasil yang diketahui.



Gambar 3: Arsitektur CNN yang digunakan dalam ini. Pengurangan dimensi PCA mengubah spektrum menjadi matriks koefisien input. Masukan ini melewati beberapa lapisan konvolusi dan lapisan yang terhubung sepenuhnya untuk melatih data dan memprediksi parameter bintang.

kami lakukan untuk menemukan model terbaik yang dapat menangani semua jenis basis data pelatihan yang digunakan dalam penelitian ini. Strategi pemilihan jumlah lapisan tersembunyi dan ukuran lapisan konvolusi akan dijelaskan pada makalah yang akan datang. Kami memutuskan untuk melakukan semua pengujian menggunakan platform ML TensorFlow<sup>6</sup> dengan antarmuka Keras<sup>7</sup>. Alasannya adalah karena kedua opsi ini bersifat **open-source** dan ditulis dalam bahasa Python.

Meskipun waktu kalkulasi merupakan salah satu **faktor** penting yang membatasi pilihan jaringan, kami telah memutuskan untuk tidak mempertimbangkannya ketika memilih jaringan yang optimal. Alasannya adalah karena waktu kalkulasi sangat bergantung pada arsitektur jaringan yang tidak dibahas dalam artikel ini. Dua parameter juga membatasi waktu kalkulasi, jumlah epoch, dan ukuran batch (terkait dengan ukuran TDB). Waktu kalkulasi meningkat dengan bertambahnya jumlah epoch dan menurun dengan bertambahnya ukuran batch. Tujuan utama dari pekerjaan ini adalah untuk menemukan konfigurasi optimal untuk parameter secara independen dari waktu kalkulasi dan arsitektur Jaringan. Sebagai patokan, dengan menggunakan Database 70.000 spektrum dan 50 vektor eigen, dibutuhkan

sekitar 17 jam untuk menjalankan CNN lebih dari 2.000 epoch dengan menggunakan 64 batch dan Dropout 30%. Perhitungan ini adalah

dilakukan pada CPU Intel Core i7-8750H @ 2.20 GHz × 6 CPU.

#### 4.1.1 Penambahan data

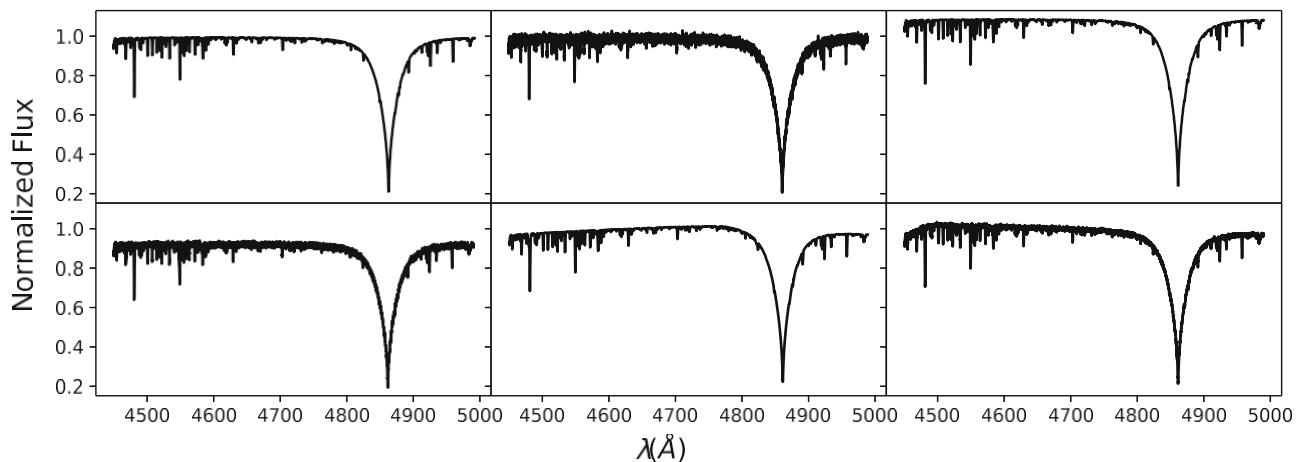
Augmentasi data adalah teknik regularisasi yang meningkatkan keragaman data pelatihan dengan menerapkan transformasi yang berbeda pada data yang sudah ada. Teknik ini biasanya digunakan untuk klasifikasi gambar (Shorten dan Khoshgoftaar 2019) dan pengenalan suara (Jaitly dan Hinton 2013). Kami menguji pendekatan ini dalam prosedur kami untuk mempertimbangkan

Tabel 2: Berbagai lapisan yang digunakan dalam CNN yang digunakan dalam pekerjaan kami

Lapisan	Bentuk keluaran	# Parameter
Conv	50 × 8	40
Conv	50 × 4	132
Conv	50 × 4	68
Datar	200	0
Penuh	1,024	205,824
Penuh	512	524,800
Penuh	64	32,832
Penuh	10	650
Penuh	1	11

□ □  
6 <https://www.tensorflow.org/>.

7 <https://keras.io/>.



Gambar 4: Efek penambahan data pada bentuk spektrum. Kiri atas: spektrum mewakili spektrum sintesis asli. Tengah atas: Derau Gaussian yang ditambahkan ke spektrum sintesis. Kanan atas: spektrum sintesis dengan intensitas yang dikalikan dengan faktor skala konstan. Kiri bawah: Derau Gaussian ditambahkan ke spektrum sintesis dan dikalikan dengan faktor skala konstan. Tengah bawah: spektrum sintesis dengan intensitas yang dikalikan dengan polinomial tingkat dua. Kanan bawah: Derau Gaussian ditambahkan ke spektrum sintesis dan dikalikan dengan polinomial tingkat dua. Semua spektrum ini memiliki parameter bintang yang sama ( $T_{\text{eff}} = 8.800 \text{ K}$ ,  $\log g = 4.3 \text{ dex}$ ,  $v_e \sin i = 45 \text{ km s}^{-1}$ ,  $[M/\text{H}] = 0.0 \text{ dex}$ ).

memperhitungkan beberapa modifikasi yang dapat terjadi pada bentuk spektrum yang diamati karena normalisasi yang buruk atau reduksi data yang tidak tepat. Kami juga memperhitungkan fakta bahwa spektrum yang diamati dipengaruhi oleh noise dan bahwa proses pembelajaran harus menyertakan efek parameter ini.

Untuk setiap spektrum di TDB, lima replika dibuat per- yang terbentuk. Masing-masing dari kelima replika ini memiliki nilai fluks yang berbeda, namun semuanya memiliki label bintang yang sama, yaitu  $T_{\text{eff}}$ ,  $\log g$ ,  $[M/H]$ , dan  $v_e \sin i$ . Modifikasi yang dilakukan adalah sebagai berikut:

- Derau Gaussian ditambahkan ke spektrum dengan SNR yang berkisar secara acak antara 5 dan 300.
- Fluks dikalikan dengan cara yang seragam dengan faktor penskalaan antara 0,95 dan 1,05.
- Fluks dikalikan dengan faktor penskalaan baru dan noise ditambahkan.
- Fluks dikalikan dengan polinomial tingkat dua dengan nilai berkisar antara 0,95 dan 1,05 dan memiliki maksimumnya dipilih secara acak antara 4.450 dan 5.000 Å.
- Fluks dikalikan dengan polinomial tingkat dua dan noise Gaussian yang ditambahkan ke dalamnya.

Tujuan dari pilihan ini adalah untuk meningkatkan dimensi TDB dari  $N_{\text{spektrum}} \times N_{(f)}$  menjadi  $6 \times N_{(\text{spektrum})} \times N_{(f)}$  dan untuk memperkenalkan beberapa modifikasi dalam spektrum pelatihan yang dapat muncul dalam pengamatan yang perlu kita analisis. Modifikasi tersebut noise dan penyimpangan yang biasa diamati dari normalisasi kontinum yang sempurna.

Distorsi dalam spektrum yang diamati dapat muncul karena pemilihan yang buruk pada titik kontinum. Kami telah menguji dua opsi, dengan dan tanpa penambahan data, dan opsi Hasil perhitungan ditunjukkan pada Bagian 5. Gambar 4 menampilkan satu spektrum sintetik yang memiliki  $T_{\text{eff}} = 8.800 \text{ K}$ ,  $\log g = 4.0 \text{ dex}$ ,  $v_e \sin i = 14 \text{ km s}^{-1}$ , dan  $[M/H] = 0.0 \text{ dex}$  serta lima modifikasi tambahan yang dilakukan pada spektrum ini.

Kami telah memutuskan untuk menggunakan SNR kontinu antara 5 dan 300, tetapi modifikasi yang berbeda dapat diuji. Sebagai contoh, González-Marcos *dkk.* (2017) mengadaptasi SNR spektrum yang digunakan dalam set data pelatihan ke SNR spektrum yang membutuhkan parameter atmosfer (set evaluasi). Mereka menyimpulkan bahwa dalam kasus  $T_{\text{eff}}$ , hanya dibutuhkan dua model regresi (SNR = 50 dan 10) untuk mencakup seluruh rentang SNR.

#### 4.1.2 Inisialisasi: Kernel dan bias

Inisialisasi menentukan cara menetapkan bobot awal. Ada berbagai cara untuk melakukan inisialisasi, dan kami akan menguji yang berikut ini:

- *No1*: bobot diinisialisasi dengan 0. Dalam hal ini, aktivasi di semua neuron adalah sama dan turunan dari fungsi kerugian serupa untuk setiap bobot di setiap neuron. Hal ini menghasilkan perilaku linier untuk model.
- *Ones*: perilaku yang serupa dengan tetapi menggunakan nilai 1, bukan 0.
- *RandomNormal*: inisialisasi dengan distribusi normal.

- *RandomUniform*: inisialisasi dengan distribusi yang seragam.
- *TruncatedNormal*: inisialisasi dengan distribusi normal terpotong.
- *VarianceScaling*: inisialisasi yang menyesuaikan skalanya dengan bentuk bobot.
- *Ortogonal*: inisialisasi yang menghasilkan matriks ortogonal acak.
- *Identitas*: inisialisasi yang menghasilkan matriks identitas.
- *Lecun\_uniform*: Inisialisasi seragam LeCun (Lecun *et al.* 1998).
- *Glorot\_normal*: Inisialisasi normal Xavier (Glorot dan Bengio 2010).
- *Glorot\_uniform*: Inisialisasi seragam Xavier (Glorot dan Bengio 2010).
- *he\_normal*: Inisialisasi normal (He *et al.* 2015).
- *Lecun\_normal*: Inisialisasi normal LeCun (Lecun *et al.* 1998).
- *he\_uniform*: Inisialisasi penskalaan varians seragam (He *et al.* 2015).

Untuk semua inisialisasi ini, bias diinisialisasi dengan nilai nol. Nanti akan ditunjukkan bahwa sebagian besar inisialisasi ini memberikan akurasi yang sama, kecuali nol dan satu.

#### 4.1.3 Pengoptimal

Setelah arsitektur jaringan (yang diparameterkan) dipilih, langkah selanjutnya adalah menemukan nilai optimal untuk parameter. Jika kita menyatakan dengan  $\theta$  kumpulan parameter kolektif, maka, menurut definisi, nilai optimal,  $\theta_*$ , adalah nilai yang meminimalkan fungsi kerugian tertentu  $L(\theta)$ ; ukuran perbedaan antara nilai prediksi dan nilai aktual. Masalah optimasi ini, biasanya, diselesaikan dengan cara berulang, dengan menghitung gradien fungsi kerugian sehubungan dengan parameter.

Biarkan  $\theta_t$  menyatakan parameter pada iterasi  $t$ .

Proses optimasi iteratif menghasilkan urutan nilai,  $\theta_1, \dots, \theta_t$ , yang konvergen ke nilai optimal  $\theta_*$ . Pada langkah tertentu  $t$ , kita mendefinisikan sejarah dari proses tersebut

sebagai himpunan  $\mathcal{Y}_t = \{\theta_t, L(\theta_t), \nabla L(\theta_t)\}^T$ . Nilai-nilai  $\theta_{t+1}$  adalah diperoleh dari  $\theta_t$  menurut beberapa aturan pembaruan

$$\theta_{t+1} = \gamma_t(H_t, \gamma_t), \quad (4)$$

di mana  $\gamma_t$  adalah satu set hiperparameter seperti laju pembelajaran.

Teknik pengoptimalan yang berbeda menggunakan aturan pembaruan yang berbeda. Sebagai contoh, dalam apa yang disebut sebagai penurunan gradien "vanilla", aturan pembaruan bergantung pada gradien yang terbaru saja:

$$\theta_{t+1} = \theta_t - \gamma \nabla L(\theta_t). \quad (5)$$

Metode lain termasuk seluruh sejarah dengan ketergantungan fungsional yang berbeda pada gradien dan tingkat yang berbeda untuk setiap langkah (lihat Choi *et al.* 2020 untuk survei). Teknik optimasi yang berbeda tersedia di keras dan kami akan menguji yang berikut ini:

- *Adam*: estimasi momen adaptif yang banyak digunakan untuk masalah dengan noise dan gradien yang jarang. Secara praktis, pengoptimal ini hanya memerlukan sedikit penyetelan untuk masalah yang berbeda.
- *RMSprop*: perambatan kuadrat rata-rata akar yang secara iteratif memperbarui tingkat pembelajaran untuk setiap parameter yang dapat dilatih dengan menggunakan rata-rata kuadrat dari gradien sebelumnya.
- *Adadelta*: ini adalah delta adaptif, di mana delta mengacu pada perbedaan antara bobot saat ini dan bobot yang baru saja diperbarui. Metode ini juga berfungsi sebagai metode penurunan gradien stokastik.
- *Adamax*: metode keturunan gradien stokastik adaptif dan varian dari Adam didasarkan pada norma tak terhingga. Metode ini juga kurang sensitif terhadap tingkat pembelajaran dibandingkan pengoptimal lainnya.
- *Nadam*: Adam yang dipercepat Nesterov yang digunakan untuk gradien dengan noise atau dengan kelengkungan yang tinggi. Ini menggunakan proses pembelajaran yang dipercepat dengan peluruhan eksponensial dari rata-rata bergerak untuk gradien sebelumnya dan saat ini. Ini juga merupakan algoritma laju pembelajaran adaptif dan membutuhkan lebih sedikit penyetelan hiperparameter.

#### 4.1.4 Tingkat pembelajaran

Seperti yang telah disebutkan di awal bagian ini, laju pembelajaran dapat mempengaruhi nilai minimum yang dicapai oleh fungsi kerugian dan oleh karena itu memiliki efek yang besar terhadap sifat generalisasi solusi. Pada artikel ini, kami mengikuti rekomendasi Bengio (2012) dan memilih nilai laju pembelajaran setengah dari laju terbesar yang menyebabkan divergensi.

#### 4.1.5 Putus sekolah

Dropout adalah teknik regularisasi untuk jaringan syaraf dan model DL yang mencegah jaringan dari overfitting (Srivastava *et al.* 2014). Ketika dropout diterapkan, neuron-neuron yang dipilih secara acak akan dihapus pada setiap iterasi pelatihan dan tidak berkontribusi pada perambatan maju dan tidak ada pembaruan bobot yang diterapkan pada neuron-neuron ini.



selama propagasi mundur. Secara statistik, hal ini memiliki efek melakukan rata-rata ensemble pada sub net yang berbeda-karya yang diperoleh dari jaringan dasar asli. Kami mencoba untuk menemukan jumlah optimal untuk fraksi neuron yang keluar. Lapisan-lapisan yang keluar diletakkan setelah setiap konvolusi

satu. Pengujian dilakukan dengan fraksi putus sekolah berkisar antara 0 dan 1.

#### 4.1.6 Pooling

Pooling layer adalah cara untuk menurunkan sampel fitur (yaitu mengurangi dimensi data) dalam database dengan mengambil beberapa bagian selama pelatihan. Metode pooling yang paling umum adalah rata-rata dan max pooling Zhou dan Chellappa (1988). Rata-rata meringkas intensitas rata-rata dari fitur-fitur dalam sebuah patch dan max pooling hanya mempertimbangkan nilai yang paling kuat (yaitu nilai tertinggi) dalam sebuah patch. Ukuran patch dan jumlah filter yang digunakan ditentukan oleh pengguna. Cara standar untuk melakukannya adalah dengan menambahkan lapisan penyatuan setelah lapisan konvolusi dan ini dapat diulang satu kali atau lebih dalam CNN yang diberikan. Namun, pooling membuat input tidak berubah-ubah terhadap terjemahan kecil. Dalam pendeteksian gambar, kita perlu mengetahui apakah fitur-fitur tersebut ada dan bukan posisi persisnya. Itulah mengapa teknik ini terbukti bermanfaat saat menganalisis gambar (Goodfellow *et al.* 2016). Hal ini tidak terjadi pada spektrum karena posisi garis harus diketahui dengan pasti (Bagian 5). Namun, seperti yang telah dibahas sebelumnya, pooling layer tidak diperlukan dalam kasus kami karena dimensi TDB telah dikurangi secara drastis dengan menerapkan PCA.

#### 4.1.7 Fungsi aktivasi

Fungsi aktivasi adalah transformasi non linier yang diterapkan pada output dari lapisan dan output ini kemudian dikirim ke lapisan neuron berikutnya sebagai input. Fungsi aktivasi memainkan peran penting dalam mendapatkan output dari sebuah model, menentukan akurasi dan efisiensi komputasi. Dalam beberapa kasus, fungsi aktivasi dapat mencegah jaringan untuk konvergen.

Fungsi aktivasi untuk lapisan *dalam* dari jaringan haruslah nonlinier, jika tidak, tidak peduli seberapa dalam jaringan tersebut, fungsi ini akan setara dengan lapisan tunggal (yaitu regresi/regresi logistik). Karena itu, kami telah menguji lima fungsi aktivasi yang adalah sebagai berikut:

$$\text{— sigmoid: } f(x) = \frac{1}{1 + e^{-x}}$$

$$\begin{aligned} \text{— tanh: } f(x) &= \frac{e^x - e^{-x}}{e^x + e^{-x}} \\ \text{— relu: } f(x) &= \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases} \\ \text{— elu: } f(x) &= \begin{cases} x, & x \geq 0 \\ \alpha(e^x - 1), & x < 0 \end{cases} \end{aligned}$$

$$\text{— selu: } f(x) = \begin{cases} \text{fix}, & x \geq 0 \\ \text{fix} \alpha(e^x - 1), & x < 0 \end{cases}$$

Penting untuk dicatat bahwa pada bagian ini kita membahas pilihan fungsi aktivasi untuk lapisan dalam saja. Pilihan aktivasi untuk lapisan terakhir biasanya lebih ditentukan oleh jenis masalah dan bagaimana seseorang memodelkannya. Sebagai contoh, jika seseorang melakukan klasifikasi biner, maka aktivasi seperti sigmoid biasanya digunakan (atau softmax untuk klasifikasi multikelas) dan diinterpretasikan sebagai probabilitas. Namun, untuk masalah seperti regresi, aktivasi linier biasanya digunakan untuk lapisan terakhir. Dalam kasus kami, yang merupakan masalah regresi murni, lapisan terakhir akan memiliki fungsi aktivasi linier.

Sigmoid dan tanh membatasi besarnya output dari lapisan menjadi  $\leq 1$ . Akan tetapi, keduanya memiliki masalah *gradien* yang *menghilang* (Glorot *et al.* 2011). Untuk magnitudo yang relatif besar, kedua fungsi tersebut akan *jenuh* dan gradiennya menjadi sangat kecil. Karena jaringan dalam mengandalkan backpropagation untuk melatih gradien, beberapa lapisan pertama, sebagai produk dari lapisan berikutnya, menjadi semakin kecil. Kelas penyearah **aktivasi**, relu, elu, dan seterusnya tampaknya meminimalkan masalah gradien yang hilang. Selain itu, mereka juga menghasilkan **representasi** yang jarang, yang tampaknya memberikan hasil yang lebih baik (He *et al.* 2015, Maas 2013).

#### 4.1.8 Fungsi kerugian

Fungsi kerugian mengontrol kesalahan prediksi NN seperti yang dijelaskan di Bagian 4. Ini adalah kriteria penting dalam mengontrol pembaruan bobot dalam NN, terutama selama propagasi mundur. Pemilihan jenis fungsi kerugian diputuskan tergantung pada jenis label keluaran. Jika keluarannya adalah **variabilitas** kategorikal, seseorang dapat menggunakan crossentropy kategorikal atau crossentropy kategorikal jarang. Jika kita berurusan dengan klasifikasi biner, crossentropy biner akan menjadi pilihan normal untuk fungsi kerugian. Terakhir, jika terjadi masalah regresi seperti yang digunakan dalam penentuan parameter spektrum bintang, maka digunakan varian dari fungsi rugi-rugi galat kuadrat rata-rata. Dalam pekerjaan kami, kami telah menguji fungsi-fungsi berikut:

$$\text{— Rata-rata galat kuadrat: } \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

$$(N) i 1 =$$

– Kesalahan logaritmik kuadrat rata-rata:  $\frac{1}{2} \sum_{i=1}^N (\log \frac{1+y_i}{1+y_i})$

$$(N) i 1 = 1+y_i$$

– Kesalahan absolut rata-rata:  $\frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$

$y$  adalah label yang sebenarnya,  $\hat{y}$  adalah label yang diprediksi, dan  $N$  adalah jumlah spektrum dalam dataset pelatihan.

Pemilihan fungsi rugi-rugi dapat berbeda dari satu penelitian dengan penelitian lainnya (Rosasco *et al.* 2004). Oleh itu, kami telah menguji ketiga fungsi di atas dalam menurunkan parameter bintang.

#### 4.1.9 Zaman

Jumlah epoch adalah berapa kali seluruh dataset digunakan untuk proposisi maju dan mundur. Jumlah Epoch mengontrol berapa kali bobot neuron diperbarui. Sambil meningkatkan jumlah Epoch, kita dapat berpindah dari underfitting ke overfitting melewati solusi optimal untuk jaringan kita.

#### 4.1.10 Batch

Alih-alih memasukkan seluruh dataset pelatihan ke dalam NN, kita dapat membaginya menjadi  $N_{\text{Batch}}$  batch dan melakukan iterasi pada semua batch per epoch. Dalam hal ini, jumlah iterasi adalah jumlah batch yang dibutuhkan untuk menyelesaikan satu epoch. Batch digunakan untuk menghindari kejenuhan memori komputer dan penurunan kecepatan iterasi. Namun, pemilihan jumlah batch yang optimal tidaklah mudah. Nilai yang diadopsi biasanya 32, 64, atau 128 (Keskar *et al.* 2016).

Salah satu ukuran keberhasilan yang paling penting untuk jaringan syaraf tiruan dalam adalah seberapa baik jaringan tersebut menggeneralisasi beberapa data uji, yang tidak termasuk dalam fase pelatihan. Pada jaringan saraf tiruan saat ini, fungsi kerugian memiliki banyak minimum. Banyak penelitian eksperimental telah menunjukkan bahwa, selama fase pelatihan, *jalur untuk mencapai* nilai minimum sama pentingnya dengan nilai akhir (Neyshabur *et al.* 2017, Zou *et al.* 2019, Zhang *et al.* 2016). Aturan praktis yang baik adalah bahwa ukuran kelompok yang "kecil", kurang dari 1% ukuran data, dapat menggeneralisasi dengan lebih baik daripada kelompok yang "besar", sekitar 10% dari data pelatihan (Keskar *et al.* 2016).

## 5 Hasil dan analisis

Pengaruh setiap parameter CNN pada akurasi parameter bintang telah diuji. Untuk , kami memiliki

menggunakan CNN yang sama dengan parameter yang sama untuk semua

pengujian kami sambil mengubah hanya yang terkait pada setiap waktu. Misalnya, untuk menemukan angka zaman terbaik, kami memperbaiki fungsi aktivasi, pengoptimal, nomor batch, persentase dropout, fungsi kerugian, dan inisialisasi kernel saat melakukan iterasi pada jumlah epoch. Parameter yang sama digunakan lagi untuk menemukan persentase dropout yang optimal dan seterusnya. Nilai-nilai tetap yang digunakan dalam perhitungan ini adalah  $he\_normal$  untuk inisialisasi kernel, rata-rata kesalahan kuadrat untuk loss function, pengoptimalisasi "ADAM", fungsi aktivasi relu, 50% dropout, 64 batch. Pengujian ini dilakukan dengan epoch 100, 500, 1.000, 2.000, 3.000, 4.000, dan 5.000. Pada semua pengujian, distribusi Pelatihan dan Validasi masing-masing adalah 80% dan 20%.

Hasilnya akan berupa kombinasi dari rentang kesalahan pengujian-

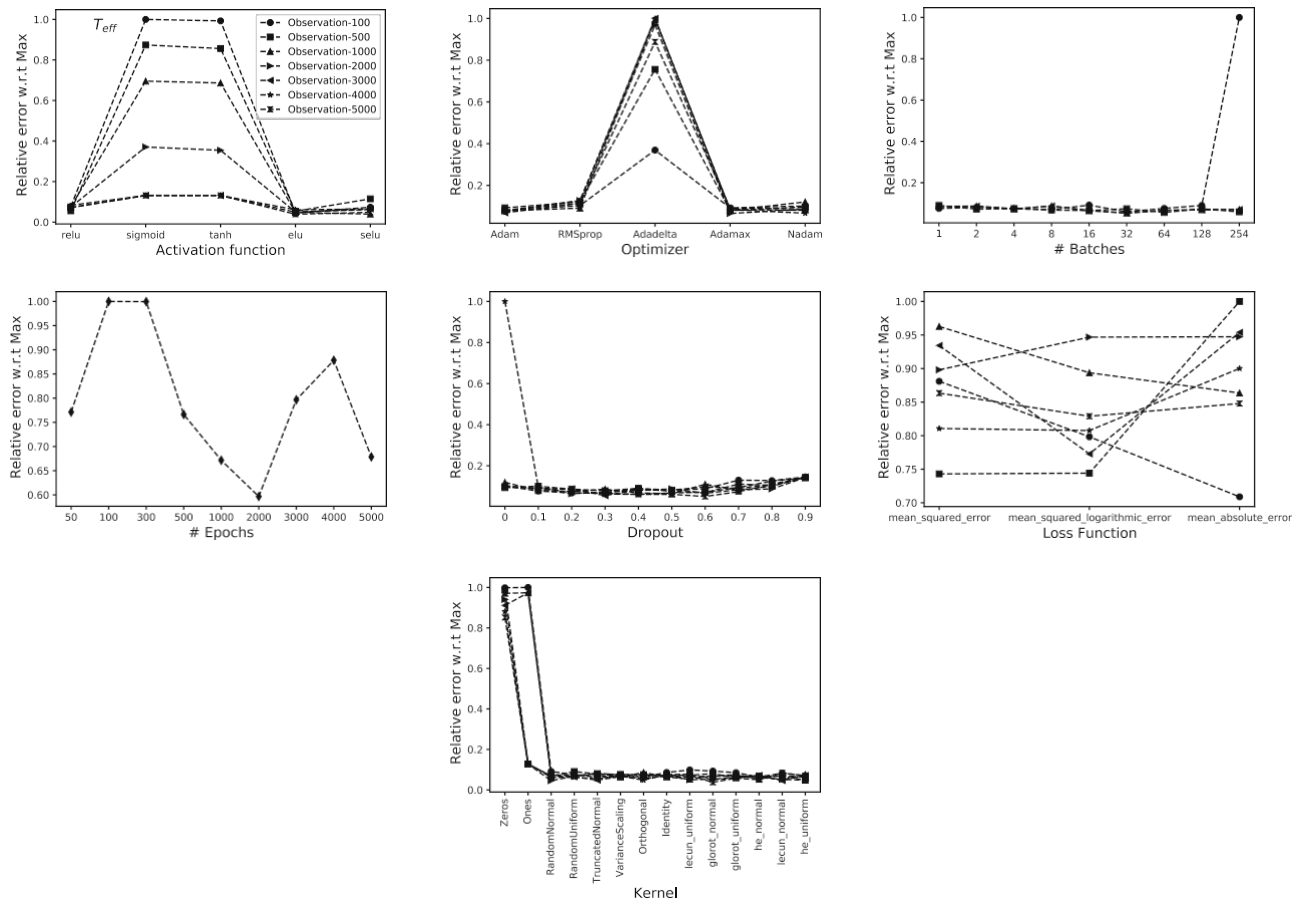
ning pada jumlah epoch yang berbeda untuk setiap parameter bintang dan konfigurasi CNN. Variasi dengan jumlah epoch memastikan bahwa tren adalah nyata dan bukan karena minimum lokal sebagai akibat dari rendahnya jumlah iterasi. Pengujian ini merupakan kumpulan 110.000 spektrum sintetik, setengahnya tanpa derau dan setengahnya lagi dengan derau acak seperti yang telah dijelaskan pada Bagian 2.

Untuk memvisualisasikan hasil dengan lebih baik dan untuk mendapatkan kesimpulan yang lebih baik tentang konfigurasi optimal, kami menampilkan pada Gambar 5-8 kesalahan relatif pengamatan. Kesalahan ini dihitung dengan membagi nilai dengan deviasi standar pengamatan maksimum di semua konfigurasi (yaitu termasuk semua simulasi epoch). Hal ini akan memungkinkan kita untuk menargetkan nilai minimum dan menentukan parameter terbaik.

Berikut ini, kami menunjukkan hasil yang diperoleh dengan menggunakan dataset pelatihan yang terdiri dari 40.000 spektrum sintesis yang dihasilkan secara acak dalam rentang Tabel 1. Pada Bagian 5.5, kami membahas efek dari penggunaan basis data pelatihan yang kecil atau besar dan efek dari penggunaan atau tidak adanya penambahan data.

### 5.1 Suhu efektif

Menurut Gambar 5, penggunaan fungsi aktivasi relu atau elu menghasilkan kesimpulan yang sama dengan selisih beberapa persen. Dan ini dapat diterapkan secara independen dari jumlah zaman. Sedangkan untuk Pengoptimal, pengoptimal Adam dan Adamax tampaknya secara konsisten akurat di semua jumlah epoch. Jumlah batch yang optimal ditemukan antara 32 dan 64. Jumlah epoch terkait erat dengan nomor batch, namun, dalam kasus 64 batch, jumlah epoch optimal ditemukan



Gambar 5: Pengaruh variasi parameter CNN terhadap akurasi  $T_{\text{eff}}$  untuk jumlah epoch yang berbeda. Hasilnya ditampilkan dengan membagi standar deviasi observasi dengan nilai maksimumnya pada setiap pengujian.

2,000. Faktor dropout, seperti yang diperkenalkan pada Bagian 4.1.5, merupakan teknik regularisasi yang menghindari overfitting. Ini berarti bahwa nilai optimal tergantung pada ukuran database pelatihan. Dalam kasus 40.000 database sampel kami, dropout optimal ditemukan antara 10 dan 60%. Jaringan syaraf meminimalkan fungsi kerugian dan karenanya mendapatkan koefisien yang akan digunakan nanti untuk memprediksi parameter pengamatan. Di antara tiga fungsi kerugian yang kami uji, perbedaan kecil ditemukan di antara mereka. Kami akan menggunakan kesalahan logaritmik kuadrat rata-rata untuk  $T_{\text{eff}}$ . Terakhir, inisialisasi koefisien jaringan dapat dilakukan dengan menggunakan inisialisasi apa pun dengan pengecualian Nol dan Satu. Jaringan syaraf cenderung terjebak pada nilai minimum lokal ketika menggunakan dua opsi ini. Rasio deviasi standar terhadap nilai maksimum menunjukkan variasi naik dan turun yang menyerupai pola jigsaw terhadap jumlah epoch. Hal terutama disebabkan oleh fakta bahwa titik-titik yang berhubungan dengan run yang berbeda. Selain itu, kita juga dapat melihat bahwa variasi kesalahan relatif sesuai dengan variasi terkecil di antara hiperparameter yang berbeda, dan ini terjadi pada semua bintang.

parameter. Tentu saja, pencarian akan bergantung pada ukuran basis data pelatihan, wilayah spektral, tipe spektral, resolusi, dan sebagainya.

Konfigurasi optimal yang kami temukan untuk  $T_{\text{eff}}$  cor-merespons parameter berikut ini:

Fungsi aktivasi: relu. Pengoptimal:

Adam.

Batch: 64.

Zaman: 2.000.

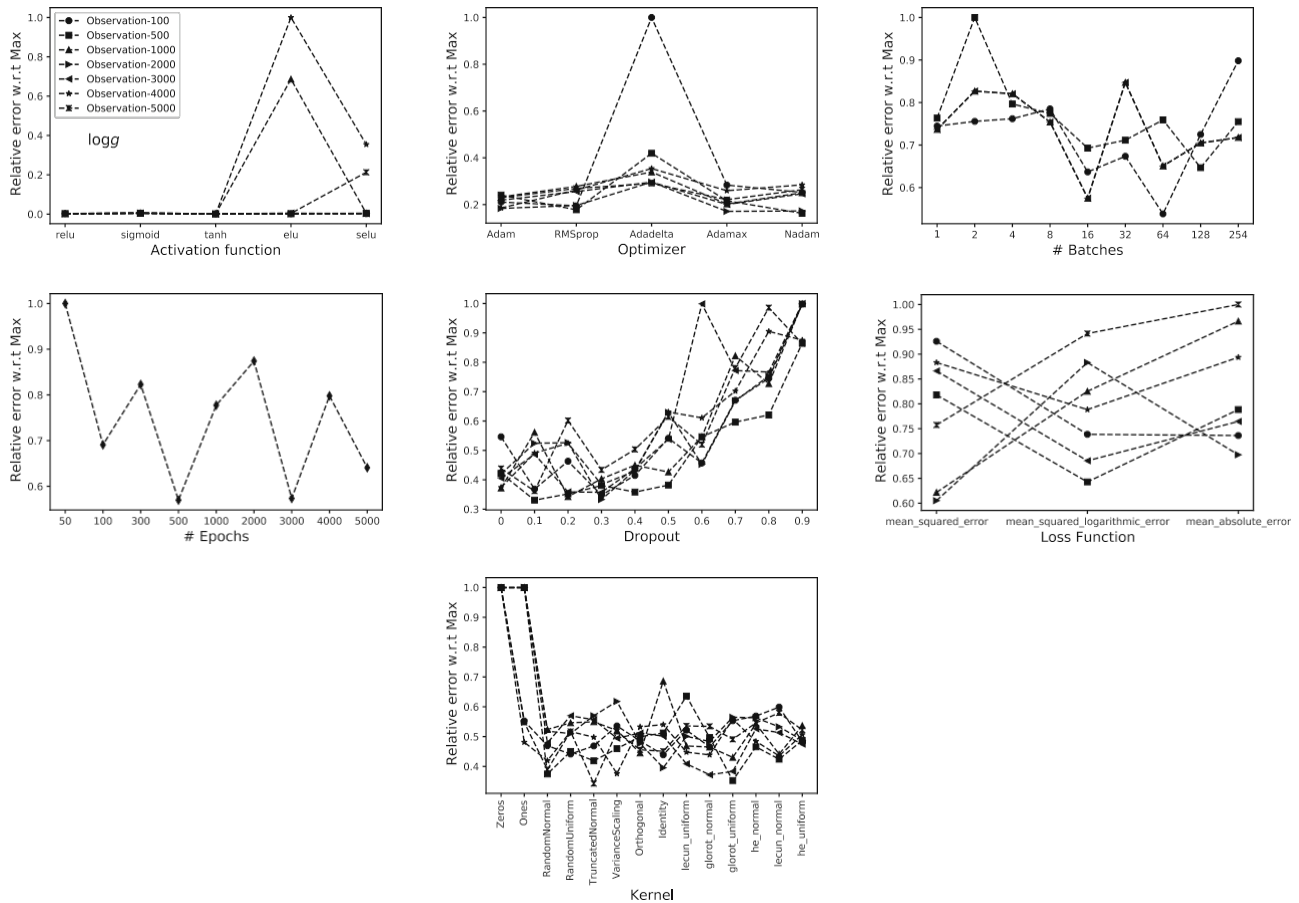
Putus sekolah: 30%.

Fungsi kerugian: rata-rata kesalahan logaritmik kuadrat.

Inisialisasi kernel: he\_normal.

## 5.2 Gravitasi permukaan

Akurasi untuk gravitasi berperilaku berbeda dengan  $T_{\text{eff}}$  sehubungan dengan berbagai parameter. Menurut Gambar 6, nilai optimal ditemukan sebagai relu atau tanh untuk fungsi aktivasi; Adam, Adamax, atau



Gambar 6: Sama seperti Gambar 5, tetapi untuk kayu gelondongan.

RMSprop untuk pengoptimal, jumlah batch antara 32 dan 128, jumlah epoch 3.000, fraksi Dropout antara 0,3 dan 0,4, fungsi kerugian kesalahan logaritmik kuadrat rata-rata, dan semua jenis inisialisasi kecuali nol dan satu.

Dalam hal *logging*, konfigurasi optimal ditemukan dengan menggunakan parameter berikut ini:

Fungsi aktivasi: tanh. Pengoptimal:

Adamax.

Batch: 128.

Zaman: 3.000.

Putus sekolah: 30%.

Fungsi kerugian: rata-rata kesalahan logaritmik kuadrat.

Inisialisasi kernel: he\_normal.

### 5.3 Metalikitas

Parameter metalitas, [M/H], juga berperilaku berbeda dengan  $T_{\text{eff}}$  dan *logg*. Seperti yang terlihat pada Gambar 7, [M / H]

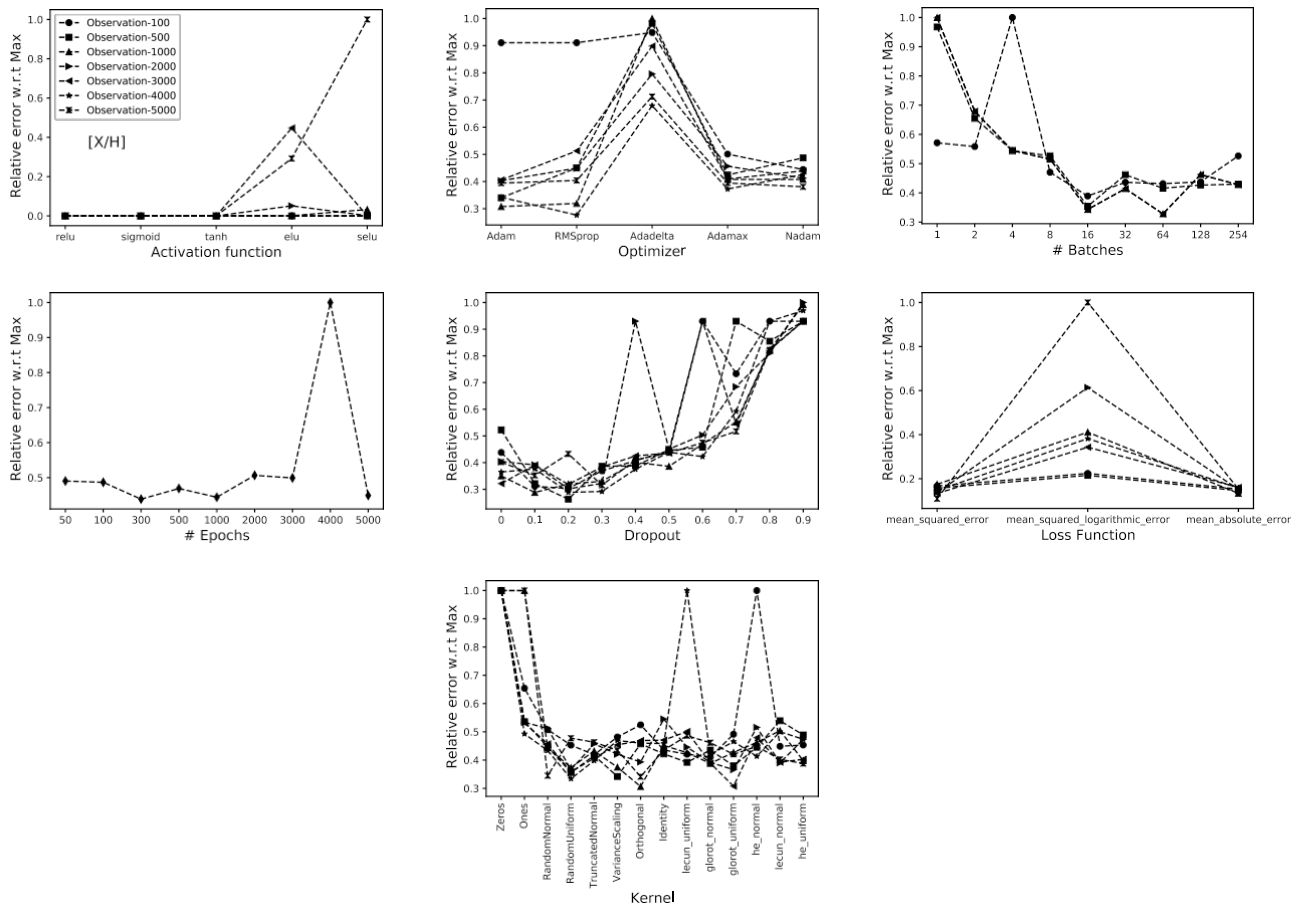
membutuhkan kombinasi parameter yang berbeda dalam CNN kami untuk mencapai hasil yang optimal. fungsi aktivasi tanh atau relu memberikan kesalahan paling kecil di sebagian besar situasi jumlah epoch. Pengoptimal Adam dan RMSprop memberikan hasil yang serupa dengan perbedaan beberapa persen. Kombinasi 16 batch dan 1.000 epoch sesuai untuk mendapatkan [M/H] dengan kesalahan yang rendah. Sebuah dropout antara 10 dan 30%, kesalahan absolut rata-rata untuk fungsi rugi-rugi, dan inisialisasi kernel RandomU- niform harus digunakan untuk mencapai akurasi setinggi mungkin untuk [M / H]. Teknik kami diterapkan pada bintang A dan diekstrapolasi pada bintang FGK (Bagian 6). Namun, pertimbangan khusus harus diperhitungkan ketika menurunkan metalitas bintang-bintang dingin karena adanya garis-garis molekul yang hadir dalam spektrum (Passegger *et al.* 2021).

Untuk [M/H], konfigurasi optimal ditemukan dengan menggunakan parameter berikut ini:

Fungsi aktivasi: tanh. Pengoptimal:

Adam.

Batch: 16.



Gambar 7: Sama seperti Gambar 5, tetapi untuk [M/H].

Zaman: 1.000.

Putus sekolah: 20%.

Fungsi kerugian: rata-rata kesalahan absolut.

Inisialisasi kernel: RandomUniform.

## 5.4 Proyeksi kecepatan rotasi khatulistiwa

Akhirnya, dalam kasus kecepatan rotasi yang diproyeksikan ekuatorial,  $v_e$  *sini*, tanh tampaknya merupakan fungsi aktivasi yang optimal secara independen dari epoch dan jumlah batch (Gambar 8). Pengoptimal Adam atau Adamax dapat digunakan untuk  $v_e$  *sini* dengan perbedaan kecil dalam akurasi yang dihasilkan. Kombinasi 32 batch dengan 3.000 epoch adalah kombinasi yang memberikan kesalahan minimum untuk nilai  $v_e$  *sini* yang diturunkan. Fraksi putus antara 0,1 dan 0,4 menghasilkan kesalahan yang sangat dekat. Kesalahan kuadrat rata-rata dapat digunakan untuk fungsi kerugian dan semua inisialisasi kernel juga dapat diterapkan kecuali nol dan satu untuk alasan yang sama seperti yang dijelaskan pada Bagian 5.1.

Dalam kasus  $v_e$  *di sini*, konfigurasi optimal ditemukan dengan menggunakan parameter berikut ini:

Fungsi aktivasi: tanh. Pengoptimal:

Adamax.

Batch: 32.

Zaman: 3.000.

Putus sekolah: 30%.

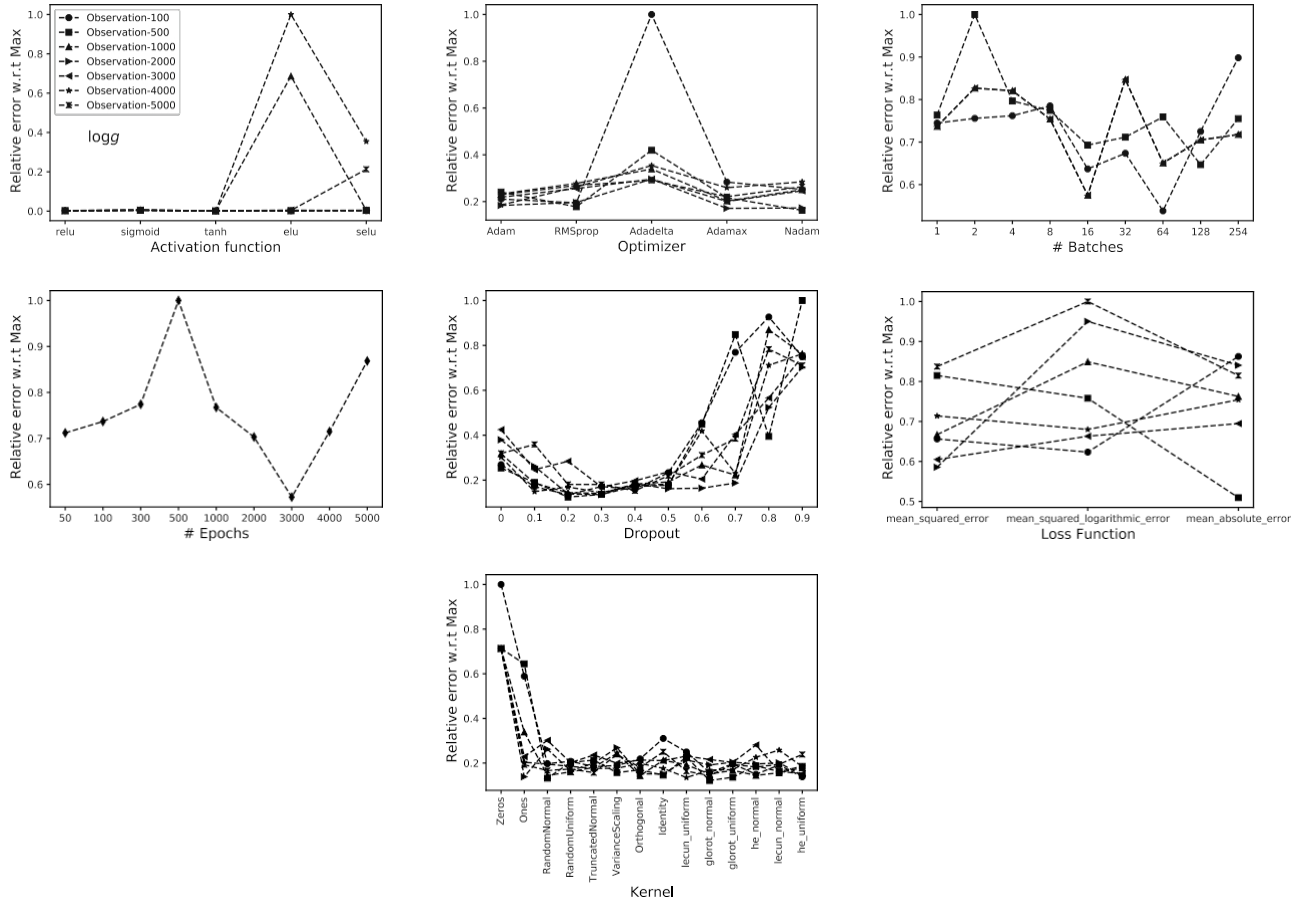
Fungsi kerugian: rata-rata kesalahan

kuadrat. Inisialisasi kernel: he\_Uniform.

## 5.5 Ukuran basis data dan peran augmentasi

Untuk memeriksa ketergantungan kinerja CNN pada ukuran set pelatihan, tiga basis data digunakan. Basis data pertama (TDB1) berisi 25.000 spektrum sintesis acak seperti yang dijelaskan pada Bagian 5, basis data kedua (TDB2) berisi 40.000 spektrum acak, dan basis data ketiga (TDB3) berisi 70.000 spektrum, yang dihasilkan dari





Gambar 8: Sama seperti Gambar 5 tetapi untuk  $v_e$  di sini.

rentang parameter TDB1 yang sama. Kami juga telah memeriksa pentingnya menggunakan Augmentasi Data sebagai teknik regularisasi untuk memperoleh parameter yang akurat (lihat Bagian 4.1.1 untuk detailnya).

Untuk setiap parameter bintang, kami menggunakan CNN optimal dengan konfigurasi yang diturunkan di Bagian 5.1-54. Setiap konfigurasi diuji dengan TDB1, TDB2, dan TDB3 dengan dan tanpa Augmentasi Data. Gambar 9 menampilkan rata-rata standar deviasi relatif untuk setiap parameter bintang sehubungan dengan nilai maksimum, untuk set pelatihan, validasi, pengujian, dan observasi. mengukur proksi-proksi ini untuk ketidakpastian teknologi, Tabel 3 mengumpulkan deviasi standar untuk empat parameter bintang sebagai fungsi dari basis data pelatihan.

Menurut Tabel 3, setiap parameter berperilaku berbeda secara menyeluruh sehubungan dengan perubahan basis data. Hal ini terutama disebabkan oleh jumlah nilai unik parameter dalam database. Oleh itu, [M/H] diwakili dengan baik oleh TDB1 tanpa penambahan data, sedangkan  $T_{\text{eff}} \log g$  dan

$v_e$  di sini membutuhkan basis data yang lebih besar agar dapat direpresentasikan dengan baik.  $\log g$  dapat direpresentasikan dengan baik dengan TDB3 dengan augmentasi data, sedangkan  $T_{\text{eff}}$  dapat diprediksi dengan TDB2 dengan augmentasi data. Terakhir,  $v_e$  di sini dapat diprediksi menggunakan TDB3 dengan penambahan data.

## 5.6 Akurasi untuk konfigurasi yang optimal

Setelah memilih konfigurasi optimal untuk setiap parameter bintang, parameter yang diprediksi ditampilkan pada Gambar 10 sebagai fungsi dari input untuk pelatihan, validasi, dan dua set set data uji. Semua data

titik-titik tersebut terletak di sekitar garis  $y=x$ . Dispersi pengamatan di sekitar garis tersebut disebabkan oleh spektrum dengan signal to noise yang sangat rendah. Akurasi yang kami temukan dengan menggunakan arsitektur CNN kami tampaknya sesuai untuk bintang A karena mereka sebanding dengan sebagian besar

menggunakan alat klasik (Aydi *et al.* 2014) atau alat statistik yang lebih rumit (Gebran *et al.* 2016, Kassou- nian *et al.* 2019). Hal yang sama berlaku untuk semua parameter.

Untuk memverifikasi efek derau pada parameter yang telah ditentukan sebelumnya, Gambar 11 menampilkan variasi akurasi nilai prediksi sehubungan dengan SNR input. Gambar tersebut juga menampilkan pengamatan yang bergantung pada nilai  $v_e$  di *sini*. Alasan untuk itu adalah bahwa

peningkatan  $v_e$  di *sini* menyebabkan pencampuran dalam spektrum dan sehingga lebih sedikit informasi yang dapat digunakan dalam prediksi. Ini

tercermin dalam kasus  $v_e$  rendah di *sini*

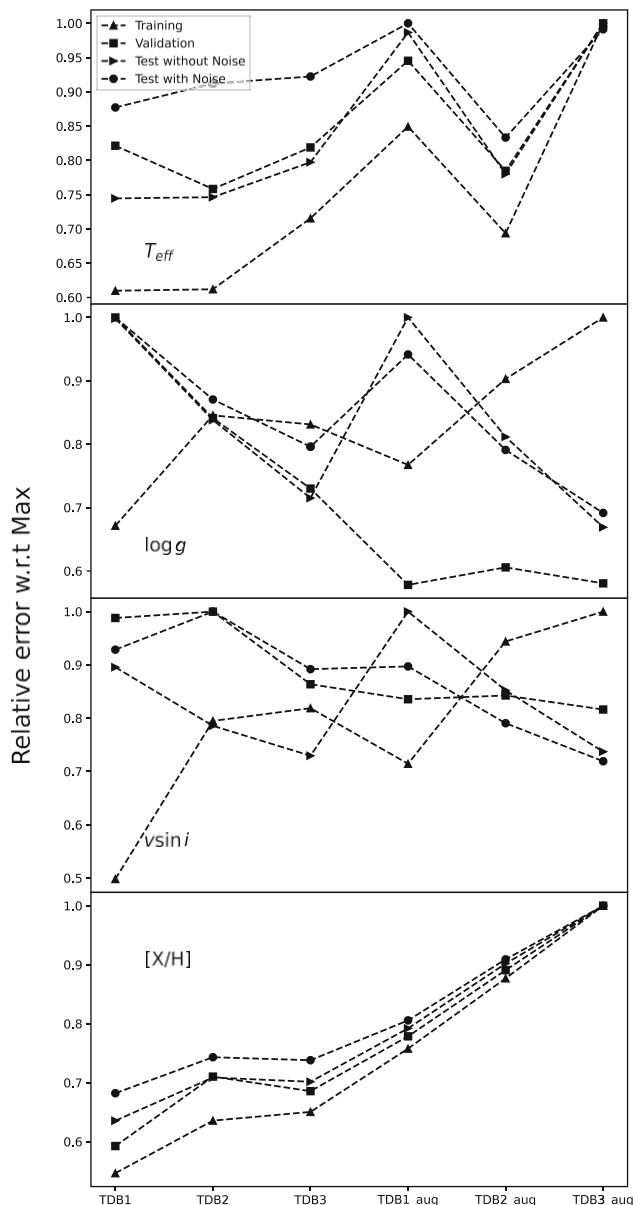
yang mana

Nilai prediksi ditemukan lebih akurat daripada kasus  $v_e$  yang besar di *sini*.

## 6 Mengekstrapolasi ke tipe spektral lainnya

Untuk memverifikasi seberapa universal

bahwa pengoptimalan kode tidak bergantung pada



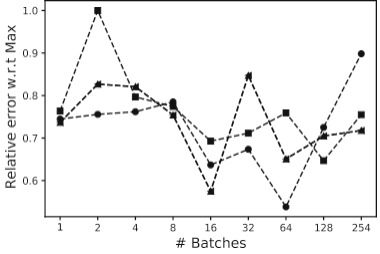
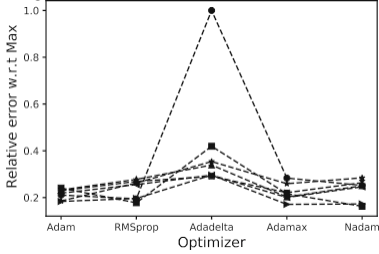
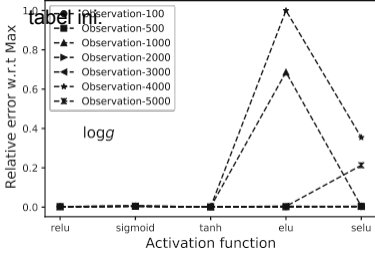
Gambar 9: Kesalahan relatif untuk setiap parameter bintang menggunakan TDB1, TDB2, dan TDB3 dengan dan tanpa

Tabel 3: Standar deviasi yang diturunkan untuk setiap parameter menggunakan TDB1, TDB2, dan TDB3 dengan dan tanpa penambahan data

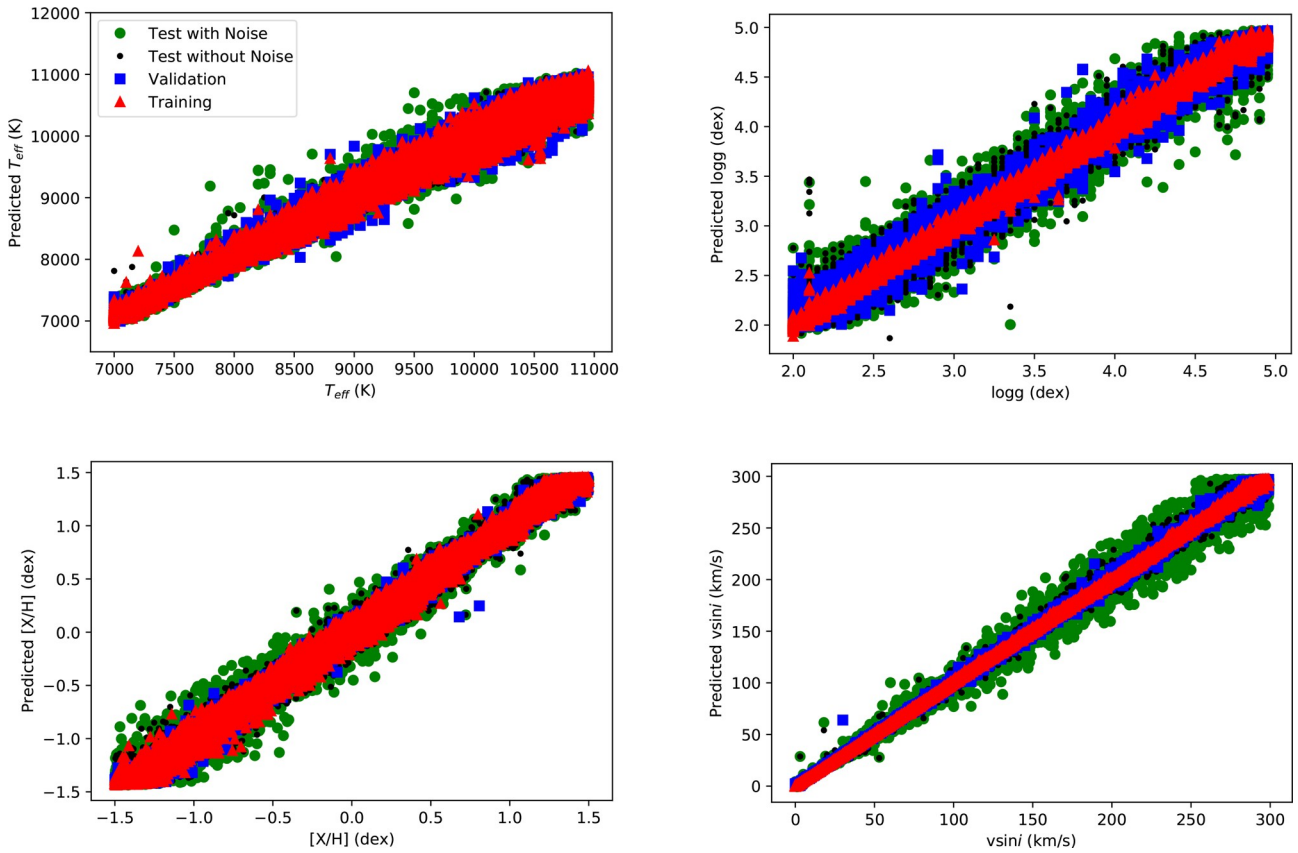
Basis data	$\sigma_{T_{\text{eff}}} \text{ (K)}$	$\sigma_{\log g} \text{ (dex)}$	$\sigma_{[M/H]} \text{ (dex)}$	$\sigma_{v_{\text{dri}}} \text{ km s}^{-1}$
TDB1				
Pelatihan	78	0.03	0.06	0.97
Validasi	112	0.11	0.07	4.00
Uji tanpa kebisingan	98	0.10	0.07	3.30
Uji dengan kebisingan	133	0.13	0.07	5.25
TDB1 dengan penambahan data				
Pelatihan	109	0.03	0.09	1.40
Validasi	129	0.07	0.09	3.36
Uji tanpa kebisingan	129	0.10	0.10	3.66
Uji dengan kebisingan	152	0.12	0.10	5.00
TDB2				
Pelatihan	79	0.04	0.08	1.55
Validasi	104	0.10	0.08	4.00
Uji tanpa kebisingan	99	0.09	0.09	2.90
Uji dengan kebisingan	139	0.11	0.09	5.50
TDB2 dengan penambahan data				
Pelatihan	89	0.04	0.10	1.85
Validasi	107	0.07	0.10	3.40
Uji tanpa kebisingan	103	0.09	0.10	3.12
Uji dengan kebisingan	127	0.10	0.11	4.36
TDB3				
Pelatihan	92	0.04	0.07	1.60
Validasi	112	0.08	0.08	3.50
Uji tanpa kebisingan	105	0.08	0.08	2.70
Uji dengan kebisingan	140	0.10	0.09	4.90
TDB3 dengan penambahan data				
Pelatihan	128	0.04	0.11	1.95
Validasi	136	0.06	0.11	3.20
Uji tanpa kebisingan	131	0.07	0.11	2.70
Uji dengan kebisingan				

penambahan data sebagai dataset pelatihan.

Nilai untuk Pelatihan, Validasi, dan dua set Tes digambarkan dalam







Gambar 10: Prediksi parameter bintang menggunakan konfigurasi CNN optimal untuk  $T_{\text{eff}}$ ,  $\log g$ ,  $v_e \sin i$ , dan  $[M/H]$  sebagai fungsi dari input untuk basis data pelatihan, validasi, dan pengujian serta untuk pengamatan yang ditambahkan noise.

Selain itu, kami juga menguji prosedur ini pada bintang-bintang FGK. Untuk itu, kami menghitung TDB khusus untuk bintang FGK dengan menggunakan parameter yang ditampilkan pada Tabel 4. Rentang panjang gelombang yang dipilih bertepatan dengan rentang panjang gelombang yang digunakan oleh Paletou *dkk.* (2015a). Rentang ini sensitif terhadap semua parameter bintang yang bersangkutan.

Sebuah basis data berisi 50.000 spektrum sintesis acak dengan label bintang yang telah diketahui digunakan dalam pelatihan. Sekitar 20.000 data uji, dengan dan tanpa , dihitung dalam rentang yang sama pada Tabel 4 untuk digunakan sebagai verifikasi. NN optimal yang diperkenalkan pada Bagian 5 digunakan lagi, sebagai bukti konsep, untuk FGK TDB. Hasilnya ditampilkan pada Tabel 5 untuk pelatihan, validasi, dan pengujian.

Karena kecepatan rotasi bintang FGK yang rendah ( $v_e \sin i < 100 \text{ km s}^{-1}$ ), maka hasil yang diperoleh lebih akurat. Hal ini tidak mengherankan karena  $v_e \sin i$  secara drastis mempengaruhi bentuk garis-garis seperti pada bintang A. Kesalahan turunan pada parameter bintang ditemukan sebesar 82 K, 0,07 dex,  $0,90 \text{ km s}^{-1}$ , dan 0,06 dex untuk  $T_{\text{eff}}$ ,  $\log g$ ,  $v_e \sin i$ , dan  $[M/H]$ , masing-masing (Tabel 5). Hasil ini sangat mencapai

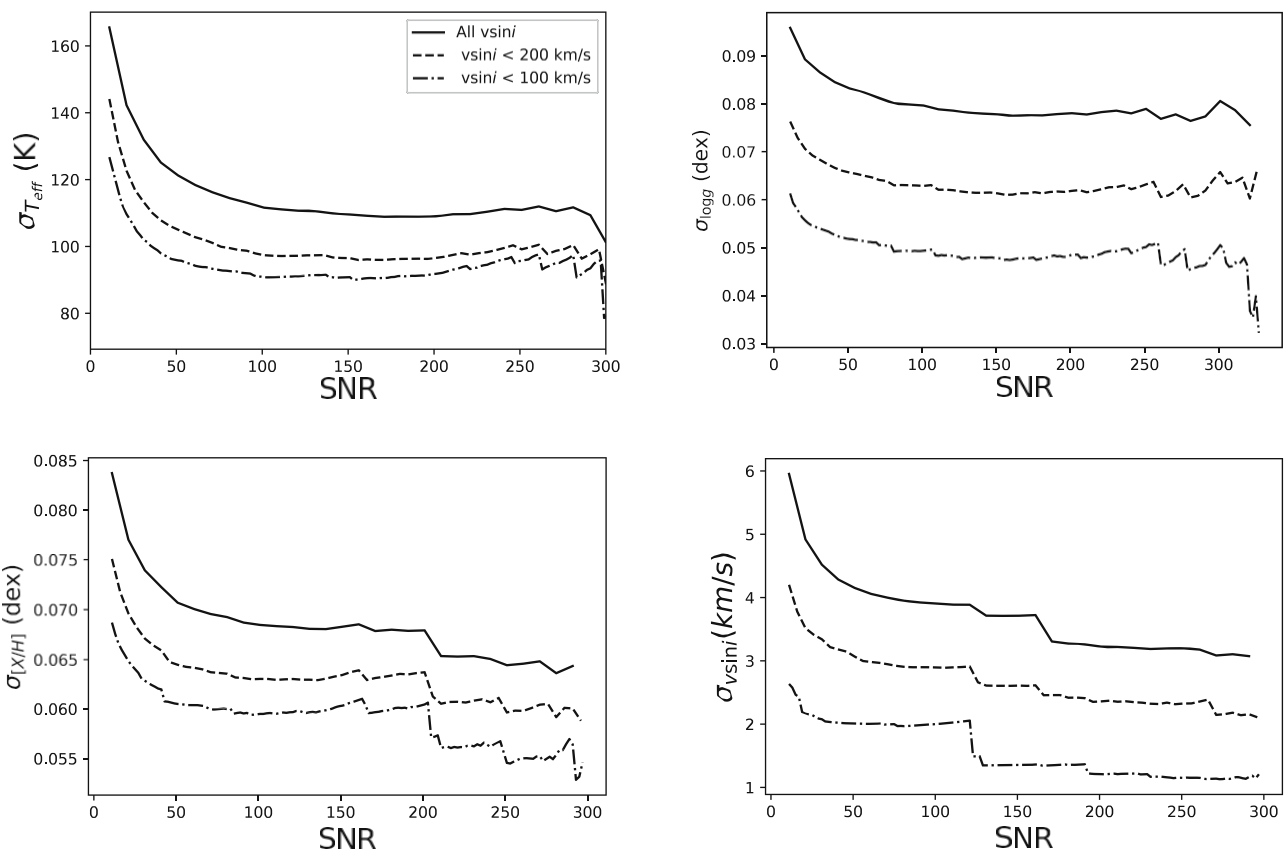
Meskipun demikian, kita harus mewaspadai komplikasi yang akan muncul ketika menggunakan pengamatan nyata, terutama dalam kasus bintang-bintang M yang dingin. Bintang-bintang ini telah dianalisis dalam konteks pencarian exoplanet (Shan *dkk.* 2021, Passegger *dkk.* 2020) dan menunjukkan adanya kerumitan pada spektrumnya, terutama yang berkaitan dengan normalisasi kontinum. Mengadaptasi persiapan data dan CNN tidak dapat dihindari untuk memperhitungkan efek-efek ini. Hasil-hasil juga menunjukkan bahwa ketika menurunkan parameter bintang untuk tipe spektral tertentu, wilayah panjang gelombang harus dipilih sesuai dengan garis/baris spektral yang paling sensitif terhadap variasi parameter yang dicari.

## 7 Diskusi dan kesimpulan

Tujuan dari pekerjaan ini tidak hanya untuk menemukan alat terbaik

untuk prediksi parameter yang akurat tetapi juga untuk menunjukkan langkah-langkah yang harus diambil untuk





Gambar 11: Bilah galat rata-rata untuk parameter bintang yang diprediksi dalam pengamatan sebagai fungsi dari SNR dan untuk rentang rotasi bintang yang berbeda.

Tabel 4: Rentang parameter digunakan untuk perhitungan spektrum sintetis TDB FGK

Parameter	Jangkauan
$T_{\text{eff}}$ (K)	[ 4,000, 7,000 ]
$\log g$ (dex)	[3.0, 5.0]
[M / H] (dex)	[-1.5, 1.5]
$v_e \text{ di } sini$ ( km s <sup>-1</sup> )	[0, 100]
$\lambda$ (Å)	[ 5,000, 5,400 ]
$\lambda / \Delta \lambda$	60,000

Tabel 5: Simpangan baku untuk setiap parameter menggunakan TDB untuk bintang FGK

	Pelatihan	Validasi	Tes (tanpa suara)	Tes (kebisingan)
$\sigma_{T_{\text{eff}}} \text{ (K)}$	59	62	62	82
$\sigma_{\log g} \text{ (dex)}$	0.04	0.05	0.05	0.07
$\sigma_{v \sin i} \text{ (km s}^{-1}\text{)}$	0.40	0.50	0.55	0.90
$\sigma_{[M/H]} \text{ (dex)}$	0.04	0.05	0.05	0.06

pemilihan optimal dari parameter CNN. Seringkali para ilmuwan menggunakan DL sebagai kotak hitam tanpa menjelaskan pilihan parameter dan/atau arsitekturnya. Dalam naskah ini, kami telah menjelaskan alasan pemilihan hiperparameter tertentu sambil menekankan pendekatan pedagogis. Untuk mendapatkan alat yang lebih efektif, seseorang harus mengubah arsitektur model. Arsitektur model tergantung pada jenis dan rentang input. Dalam pekerjaan ini, kami telah memperbaiki arsitektur dan melakukan iterasi pada hiperparameter saja.

Bagian 5.1-54 menunjukkan bahwa untuk setiap parameter bintang, pengaturan jaringan harus diubah. Ini berarti bahwa untuk jaringan tertentu dan parameter bintang tertentu, a Studi harus dilakukan untuk menemukan konfigurasi hiperparameter yang optimal. Hal ini disebabkan oleh kontribusi parameter bintang tertentu pada bentuk spektrum masukan. Dengan menggunakan dekomposisi PCA, kami telah mengurangi ukuran parameter input menjadi hanya 50 titik per spektrum sambil mempertahankan lebih dari 99,5% informasi. Hal ini direkomendasikan untuk database yang besar dan rentang panjang gelombang yang luas dan dapat menghindari penggunaan lapisan penyatuan ekstra dalam jaringan. Teknik proyeksi ini tidak hanya dapat digunakan untuk bintang AFGK tetapi juga dapat digunakan untuk bintang-bintang yang lebih dingin.

Lebih jauh lagi, (Houdebine *dkk.* 2016, Paletou *dkk.* 2015b), Sarro *dkk.* (2018) telah menerapkan model regresi pengejaran proyeksi berdasarkan koefisien kompresi analisis komponen independen untuk memperoleh  $T_{\text{eff}}$ ,  $\log g$ , dan  $[M/H]$  bintang tipe-M.

Meskipun arsitektur CNN tidak dioptimalkan, kami mampu, dengan menggunakan strategi menemukan parameter hiper terbaik, untuk mencapai tingkat akurasi yang sebanding dengan teknik yang diadopsi lainnya. Faktanya, kami menemukan untuk bintang A, akurasi rata-rata 0,08 dex untuk  $\log g$ , 0,07

dex untuk  $[M/Jam]$ ,  $3,90 \text{ km s}^{-1}$  untuk  $v_e \text{ di sini}$ , dan 127 K untuk  $T_{\text{eff}}$ . Dalam

untuk kasus bintang dengan  $v_e \text{ di sini}$  kurang dari  $100 \text{ km s}^{-1}$ , kami menemukan ketepatannya adalah 90 K, 0,06 dex, 0,06 dex, dan

$2,0 \text{ km s}^{-1}$ , untuk  $T_{\text{eff}}$ ,  $\log g$ ,  $[M/Jam]$  dan  $v_e \text{ di sini}$ , masing-masing.

Nilai akurasi ini bergantung pada sinyal terhadap derau dan berkurang selama sinyal terhadap derau meningkat. Ekstrapolasi teknik ini pada bintang FGK juga menunjukkan bahwa jaringan yang sama dapat diterapkan pada tipe spektral yang berbeda dan rentang panjang gelombang yang berbeda.

Teknik yang kami gunakan dalam artikel ini dapat ditransfer ke masalah klasifikasi apa pun yang melibatkan jaringan saraf. Di masa depan, kami berencana untuk mengembangkan strategi untuk menemukan arsitektur CNN terbaik yang bergantung pada data masukan dan jenis parameter yang diprediksi. Setelah arsitektur dan konfigurasi parameter ditentukan, kami akan menguji prosedur pada spektrum observasi seperti yang kami lakukan dalam penelitian Paletou *dkk.* (2015a),

Paletou *dkk.* (2015b), Gebran *dkk.* (2016), dan Kassounian *dkk.* (2019). Dengan hanya menggunakan data pengamatan atau kombinasi spektrum sintetis dan pengamatan nyata dengan parameter yang sudah diketahui, kita dapat membatasi label bintang yang diturunkan sekaligus meminimalkan kesenjangan sintetis yang kritis (Fabbro *et al.* 2018). Satu lagi kriteria yang harus diperhatikan adalah ketika menerapkan teknik ini pada pengamatan riil, persiapan data yang matang harus dilakukan untuk memperhitungkan karakteristik setiap tipe spektrum (misalnya normalisasi kontinum pada bintang M dan bintang raksasa, dan jumlah garis yang rendah pada bintang panas).

**Konflik kepentingan:** Penulis menyatakan tidak ada konflik kepentingan.

## Referensi

- Almeida I, Duarte R, Nemmen R. 2021. Model pembelajaran mendalam untuk emisi multi-panjang gelombang dari inti galaksi aktif dengan luminositas rendah. *nuclei*. arXiv e-prints. halaman arXiv: 2102.05809.
- Anthony M, Bartlett PL. 1999. Pembelajaran Jaringan Syaraf Tiruan: Dasar-dasar Teoritis. Cambridge: Cambridge University Press.
- Aydi E, Gebran M, Monier R, Royer F, Lobel A, Blomme R. 2014. Prosedur otomatis untuk menurunkan parameter fundamental B dan bintang A: Aplikasi pada gugus muda NGC 3293. Dalam: Ballet J, Martins F, Bornaud F, Monier R, Reylé C, editor, SF2A-2014: Prosiding pertemuan tahunan Perhimpunan Astronomi dan Astrofisika Prancis, hal. 451-455.
- Bai Y, Liu J, Bai Z, Wang S, Fan D. 2019. Regresi pembelajaran mesin temperatur efektif bintang pada data gaia kedua rilis. *AJ*, 158(2):93.
- Bailer-Jones CAL. 1997. Klasifikasi jaringan syaraf dari bintang spektrum. *PASP*. 109:932.
- Baron D. 2019. Pembelajaran Mesin dalam Astronomi: sebuah tampilan praktis di atas. arXiv e-prints, halaman arXiv: 1904.07248.
- Bengio Y. 2012. Rekomendasi praktis untuk pelatihan berbasis gradien dari arsitektur dalam. Dalam Jaringan saraf: trik perdagangan. Berlin, Heidelberg: Springer.
- Bickley RW, Bottrell C, Hani MH, Ellison SL, Teimoorinia H, Yi KM, *dkk.* 2021. Identifikasi jaringan saraf konvolusi dari galaksi pasca penggabungan di UNIONS menggunakan IllustrisTNG. *MNRAS*. 504:372-92.
- Castelli F, Kurucz RL. 2003. Kisi-kisi baru dari model ATLAS9 atmosfera. Dalam Piskunov N, Weiss WW, Gray DF, editor. Pemodelan dari Atmosfer Bintang. vol. 210, hal. A20.
- Chen BH, Goto T, Kim SJ, Wang TW, Santos DJD, Ho SCC, *dkk.* 2021. Model pengenalan inti galaksi aktif berdasarkan jaringan saraf dalam. *MNRAS*, 501(3):3951-3961.
- Choi D, Shallue CJ, Nado Z, Lee J, Maddison CJ, Dahl GE. 2020. Pada perbandingan empiris pengoptimal untuk pembelajaran mendalam. arXiv preprint arXiv: 1910.05446.
- Cropper M, Katz D, Sartoretti P, Panuzzo P, Seabroke G, Smith M, *dkk.* 2014. Kinerja spektrometer kecepatan radial Gaia. Dalam EAS Publications Series. vol. 67-68 p. 69-73.
- Curran SJ, Moss JP, Perrott YC. 2021. Pergeseran merah fotometrik QSO menggunakan pembelajaran mesin dan jaringan saraf. *MNRAS*. 503:2639-2650.
- Dafonte C, Fustes D, Manteiga M, Garabato D, Álvarez MA, Ulla A, *dkk.* 2016. Pada estimasi parameter bintang dengan prediksi yang tidak tepat dari jaringan saraf tiruan generatif: aplikasi untuk spektrum simulasi Gaia RVS. *A&A*. 594:A68.
- Fabbro S, Venn KA, O'Brian T, Bialek S, Kieley CL, Jahandar F, *dkk.* 2018. Aplikasi pembelajaran mendalam dalam analisis spektrum bintang. *MNRAS*. 475(3):2978-2993.
- Gafeira R, Orozco Suárez D, Milić I, Quintero Noda C, Ruiz Cobo B, Uitenbroek H. 2021. Inisialisasi pembelajaran mesin untuk mempercepat erasi inversi profil Stokes. *A&A*. 651:A31.
- Gan FK, Bekki K, Hashemizadeh H. 2021. SeeingGAN: Galactic Image Deblurring dengan Deep Learning untuk Klasifikasi Morfologi Galaksi yang Lebih Baik. arXiv e-prints, halaman arXiv:2103.09711.
- Garraffo C, Protopapas P, Drake JJ, Becker I, Cargile P. 2021. StelNet: Jaringan Syaraf Hirarkis untuk Inferensi Otomatis dalam Karakterisasi Bintang. arXiv e-prints, halaman arXiv:2106.07655.
- Gebran M, Farah W, Paletou F, Monier R, Watson V. 2016. Metode baru untuk inversi parameter atmosfer bintang A/Am. *A&A*. 589:A83.
- Gebran M, Monier R, Royer F, Lobel A, Blomme R. 2014. Mikroturbulensi pada bintang A / F Am / Fm. Dalam Mathys G, Griffin ER, Kochukhov O, Monier R, Wahlgren GM, editor. Menempatkan Bintang dalam Konteks: Evolusi, Lingkungan, dan Bintang Terkait, Prosiding Konferensi Internasional. 2013 Jun 3-7; Moskow, Rusia. p. 193-198.

- Gill S, Maxted PFL, Smalley B. 2018. Parameter atmosfer bintang FGK menggunakan analisis wavelet spektrum CORALIE. *A&A*. 612:A111.
- Glorot X, Bengio Y. 2010. Memahami kesulitan melatih jaringan saraf tiruan feedforward dalam. Dalam Teh YW, Titterington M, editor. *Prosiding Konferensi Internasional ke-13 tentang Kecerdasan Artifisial dan Statistik*. 2010 Mei 13-15; Sardinia, Italia. *JMLR*, 2010. p. 249-256.
- Glorot X, Bordes A, Bengio Y. 2011. Penyearah saraf yang jarang dalam jaringan. Dalam Gordon G, Dunson D, Dudek M, editor. *Prosiding Konferensi Internasional Keempat Belas tentang Kecerdasan Buatan dan Statistik*, volume 15 dari *Prosiding Penelitian Pembelajaran Mesin*. hal. 315-323. FL, AMERIKA SERIKAT: Fort Lauderdale, Lokakarya dan Konferensi *JMLR Prosiding*.
- González-Marcos A, Sarro LM, Ordieres-Meré J, Bello-García A. 2017. Evaluasi teknik kompresi data menyimpulkan parameter atmosfer bintang dari spektrum resolusi tinggi. *MNRAS*. 465(4):4556-4571.
- Goodfellow I, Bengio Y, Courville A. 2016. *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- Guiglion G, Matijević G, Queiroz ABA, Valentini M, Steinmetz M, Chiappini C, *dkk*. 2020. Percobaan Kecepatan RADIAL (RAVE): Parameterisasi spektrum RAVE berdasarkan jaringan saraf konvolusi. *A&A*. 644:A168.
- He K, Zhang X, Ren S, Sun J. 2015. Menggali lebih dalam tentang penyearah: Melampaui kinerja tingkat manusia pada klasifikasi imagenet. Dalam *Konferensi Internasional IEEE 2015 tentang Visi Komputer (ICCV)*. p. 1026-1034. Doi: 10.1109/ICCV.2015.123.
- Houdebine ER, Mullan DJ, Paletou F, Gebran M. 2016. Korelasi rotasi-aktivitas pada bintang katai K dan M. I. Parameter Bintang dan Kompilasi v sin I dan P/sin I untuk Sampel Besar Katai K dan M Akhir. *ApJ*. 822(2):97.
- Hubeny I, Lanz T. 1992. Metode linearisasi lengkap yang dipercepat untuk menghitung atmosfer bintang model NLTE. *A&A*. 262(2):501-514.
- Jaitly N, Hinton E. 2013. Gangguan panjang saluran vokal (VTLP) meningkatkan pengenalan suara. Dalam *Prosiding Lokakarya ICML tentang Pembelajaran Mendalam untuk Audio, Ucapan, dan Bahasa*. vol. 117: hal. 21.
- Kassounian S, Gebran M, Paletou F, Watson V. 2019. Regresi Invers Irisan: aplikasi untuk parameter bintang fundamental. *Buka Astron*. 28(1):68-84.
- Keskar NS, Mudigere D, Nocedal J, Smelyanskiy M, Tang PTP. 2016. Pada pelatihan batch besar untuk pembelajaran mendalam: Kesenjangan generalisasi dan minima yang tajam. *mengutip arxiv:1609.04836* Komentar: Diterima sebagai makalah konferensi di *ICLR 2017*.
- Kurucz RL. 1992. Data atom dan molekul untuk perhitungan opasitas. *RMXAA*. 23:45.
- Landa V, Reuveni Y. 2021. Jaringan saraf konvolusi dimensi rendah untuk klasifikasi deret waktu jilatan api matahari GOES. *arXiv e-prints*, halaman *arXiv*: 2101.12550.
- Lecun Y, Bottou L, Bengio Y, Haffner P. 1998. Berbasis gradien pembelajaran yang diterapkan pada pengenalan dokumen. *Proc IEEE*. 86(11):2278-2324.
- LeCun Y. 1989. Generalisasi dan strategi desain jaringan. *Perspektif Koneksiisme*. 19:143-155.
- Li X-R, Pan R-Y, Duan F-Q. 2017. Parameterisasi spektrum bintang menggunakan jaringan saraf tiruan. *Res Astronom Astrophys*. 17(4):36.
- Maas AL. 2013. Penyearah nonlinier meningkatkan jaringan saraf model akustik. Dalam *Proc ICML*. Vol. 30, No. 1, p. 3.
- Neyshabur B, Bhojanapalli S, Mcallester D, Srebro N. 2017. Menjelajahi generalisasi dalam pembelajaran mendalam. Dalam Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, *dkk*. editor. *Kemajuan dalam Pemrosesan Informasi Saraf Sistem*. vol 30, p. 5947-5956. Curran Associates, Inc.
- Ofman L, Averbuch A, Shlisselberg A, Benaun I, Segev D, Rissman A. 2022. Identifikasi otomatis kandidat exoplanet transit dalam data Transiting Exoplanet Survey Satellite (TESS) NASA dengan metode pembelajaran mesin. *New Astron*. 91:101693.
- Paletou F, Böhm T, Watson V, Trouillet JF. 2015a. Inversi parameter fundamental bintang dari spektrum resolusi tinggi ESPaDOnS dan Narval. *A&A*. 573:A67.
- Paletou F, Gebran M, Houdebine ER, Watson V. 2015b. Inversi temperatur efektif berbasis analisis komponen utama untuk bintang-bintang tipe akhir. *A&A*. 580:A78.
- Parks D, Prochaska JX, Dong S, Cai Z. 2018. Pembelajaran mendalam tentang spektrum quasar untuk menemukan dan mengkaraktirasi sistem Ly yang teredam. *MNRAS*. 476(1):1151-1168.
- Passegger VM, Bello-García A, Ordieres-Meré J, Antoniadis-Karnavas A, Marfil E, Duque-Arribas C, *dkk*. 2021. Metalikitas di katai M: Menyelidiki teknik penentuan yang berbeda. *arXiv e-prints*, halaman *arXiv*: 2111. 14950.
- Passegger VM, Bello-García A, Ordieres-Meré J, Caballero JA, Schweitzer A, González-Marcos A, *dkk*. 2020. CARMENES mencari exoplanet di sekitar katai M. Pendekatan pembelajaran mendalam untuk menentukan parameter fundamental bintang target. *A&A*. 642:A22.
- Portillo SKN, Parejko JK, Vergara JR, Connolly AJ. 2020. Pengurangan dimensi spektrum SDSS dengan variasi penyandi otomatis. *AJ*. 160(1):45.
- Ramírez Vélez JC, Yáñez Márquez C, Córdova Barbosa JP. 2018. Menggunakan algoritme pembelajaran mesin untuk mengukur magnetik bintang bidang. *A&A*. 619:A22.
- Rhea C, Rousseau-Nepton L. 2021. Penerapan pembelajaran mesin untuk spektrum optik - kendala kinematik. Dalam *American Astronomical Society Meeting Abstracts*. volume 53 dari *American Astronomical Society Meeting Abstracts*. 208.01.
- Rhea C, Rousseau-Nepton L, Prunet S, Hlavacek-Larrondo J, Fabbro S. 2020. Pendekatan pembelajaran mesin untuk pengamatan spektroskopi unit medan integral. *Kinematika wilayah I. H ii*. *ApJ*. 901(2):152.
- Rosasco L, Vito ED, Caponnetto A, Piana M, Verri A. 2004. Apakah fungsi kehilangan semua sama? *Komputasi Saraf*. 16(5):1063-1076.
- Ruder S. 2016. Tinjauan umum tentang algo optimasi gradient descent. *rithms*. *CoRR*. abs/1609.04747.
- Sarro LM, Ordieres-Meré J, Bello-García A, González-Marcos A, Solano E. 2018. Estimasi parameter atmosfer bintang tipe-M: perspektif pembelajaran mesin. *MNRAS*. 476(1):1120-1139.
- Shan Y, Reiners A, Fabbian D, Marfil E, Montes D, Tabernero HM, *dkk*. 2021. Pencarian exoplanet di sekitar bintang katai M oleh CARMENES. Garis-garis vanadium yang terbelah sangat halus pada spektrum bintang dingin. *A&A*. 654:A118.
- Sharma K, Kembhavi A, Kembhavi A, Sivarani T, Abraham S, Vaghmare K. 2020. Penerapan jaringan saraf konvolusi untuk klasifikasi spektral bintang. *MNRAS*. 491(2):2280-2300.

- Shorten C, Khoshgoftaar T. 2019. Sebuah survei tentang augmentasi data gambar- tasi untuk pembelajaran mendalam. *J Big Data*. 6:1-48.
- Smalley B. 2004. Pengamatan konveksi pada bintang-bintang tipe A. Dalam Zverko J, Ziznovsky J, Adelman SJ, Weiss WW, editor. *Prosiding International Astronomical Union 2004 (IAUS224), The A-Star Puzzle*. hal. 131-138. Cambridge, Inggris: Cambridge University Press.
- Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. 2014. Putus sekolah: cara sederhana untuk mencegah jaringan saraf dari overfitting. *J Mach Learn Res*. 15(1):1929-1958.
- Wang R, Luo AL, Chen JJ, Hou W, Zhang S, Zhao YH, LAMOST MRS kolaborasi, *et al*. 2020. SPCANet: parameter bintang dan jaringan kelimpahan kimia untuk survei resolusi menengah LAMOST-II- tion. *ApJ*. 891(1):23.
- Wang S-C. 2003. *Jaringan syaraf tiruan*. hal. 81-100. US, Boston, MA: Springer.
- Yim J, Ju J, Jung H, Kim J. 2015. Klasifikasi citra menggunakan jaringan syaraf tiruan konvolusi dengan fitur multi-tahap. Dalam Kim J-H, Yang W, Jo J, Sincak P, Myung H, editor. *Teknologi dan Aplikasi Kecerdasan Robot 3*. Cham: Springer International Publishing, hal. 587-594.
- Zhang B, Liu C, Deng L-C. 2020. Menurunkan Label Bintang dari Spektrum LAMOST dengan Stellar LAbel Machine (SLAM). *ApJS*. 246(1):9.
- Zhang C, Bengio S, Hardt M, Recht B, Vinyals O. 2016. Memahami pembelajaran yang mendalam membutuhkan pemikiran ulang tentang Dalam Konferensi Internasional ke-5 tentang Representasi Pembelajaran, ICLR 2017 - Jalur Konferensi Prosiding. *arXiv:1611.03530*.
- Zhou Y.T., Chellappa R. 1988. Komputasi aliran optik menggunakan jaringan syaraf tiruan. Dalam ICNN. p. 71-78.
- Zhu X, Vondrick C, Fowlkes CC, Ramanan D. 2016. Apakah kita membutuhkan lebih banyak data pelatihan ? *Int J Comput Vision*. 119(1):76-92.
- Zou D, Cao Y, Zhou D, Gu Q. 2019. Gradient descent mengoptimalkan jaringan relu dalam yang diparameterisasi secara berlebihan. *Mach Learn*. 109:467-492.