Tugas Kecerdasan Artificial Lanjutan

DETEKSI PENIPUAN KARTU KREDIT MENGGUNAKAN ARTIFICIAL NEURAL NETWORK DENGAN PYTHON

Abdul Aziz

NIM. 3337220051

Universitas Sultan Ageng Tirtayasa

2024

A. PENDAHULUAN

Pada era saat ini, penggunaan AI tidak bisa dipisahkan dari kehidupan manusia sehari – hari, seperti Siri, Google Assistant, Cortana, ChatGPT, dan Bing. Kelajuan perkembangan teknologi AI saat ini sangat luar biasa, dengan adanya teknologi yang dapat bekerja sendiri seperti Mobil yang bekerja tanpa pengemudi atau diciptakannya ChatGPT yang memberikan solusi atas pertanyaan yang diajukan. Kecerdasan buatan memiliki potensi besar di masa depan karena juga dapat digunakan dalam VR dan AR. Oleh karena itu, sektor ini sedang difokuskan untuk pengembangan teknologi yang lebih maju lagi.

Artificial Intelligence adalah suatu cabang ilmu komputer yang mensimulasikan otak manusia, dimana dapat memproses suatu informasi dan dapat menentukan keputusan. Dimana pada AI terdapat cabang ilmu khusus yang mempelajari sebuah sistem dimana mesin akan mempelajari sesuatu seperti cara kerja otak manusia, itu yang dinamakan dengan Jaringan Syaraf Tiruan (JST). Dengan cara kerja yang seperti otak manusia JST atau ANN akan diberikan suatu input lalu setelah itu akan diproses pada layer berikutnya yang dinamakan dengan *hidden layer*. Pada JST bisa memiliki lebih dari satu *hidden layer*, yang paling umum adalah bentuk dari perceptron.

Perceptron pertamakali ditemukan oleh Frank Rosenblatt pada tahun 1958, perceptron adalah jaringan saraf paling sederhana dimana ada n inputan yang neuron dan outputnya berjumlah satu. Pada perceptron terdiri 3 langkah, pertama setiap inputan akan dikalikan dengan bobot dan akan dijumlahkan semua nilai yang sudah dikalikan dengan bobot. Bobot mewakili dominasi suatu masukan yang berpengaruh terhadap keluaran, contoh jika bobot w1

lebih besar dari bobot w2 maka artinya masukan dari x1 lebih berpengaruh dibanding masukan dari x2.

$$x.w = (x_1 \times w_1) + (x_2 \times w_2) + \cdots + (x_n \times w_n)$$

Langkah kedua akan dilakukan penambahan bias. Bias adalah suatu nilai offset yang terkadang diperlukan untuk memindahkan seluruh fungsi aktivasi kekiri ataupun kekanan untuk menghasilkan nilai output yang diinginkan.

$$z = x.w + b$$

Langkah terakhir adalah langkah dimana kita meneruskan nilai z yang didapat sebelumnya ke fungsi aktivitas non-linear. Pada perceptron terdapat fungsi langkah biner sebagai fungsi aktivitasnya. Fungsi aktivitas adalah suatu fungsi yang membuat jaringan syaraf dapat memiliki signifikansi yang baik terhadap kecepatan pembelajarannya. Selain dari fungsi biner terdapat juga fungi sigmoid yang dapat digunakan sebagai fungsi aktivasi.

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Pada rumus tersebut σ menunjukkan fungsi aktivasi sigmoid dan output yang kita peroleh dikenal sebagai nilai prediksi ŷ.

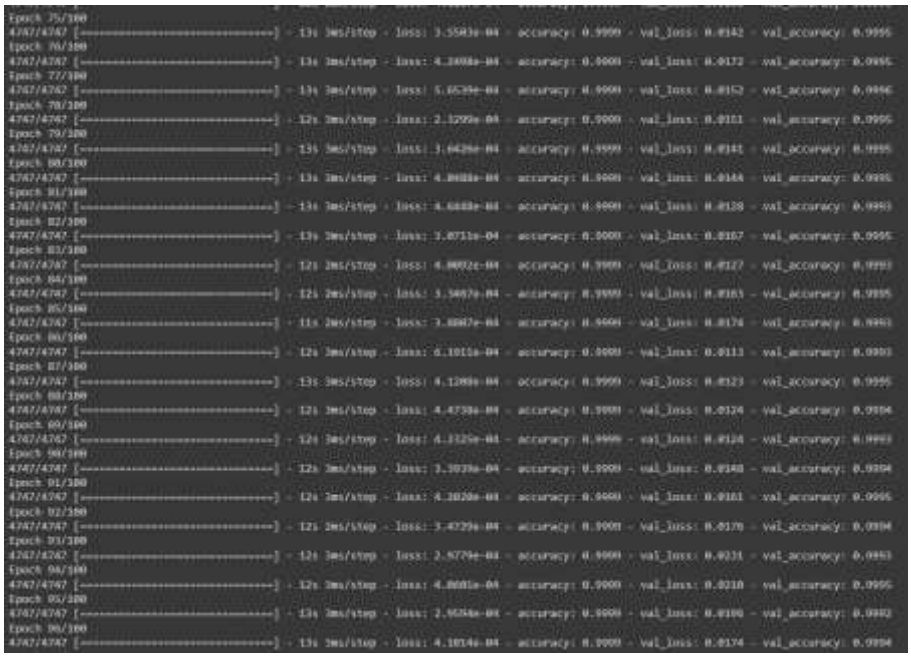
B. MODEL DAN HASIL SIMULASI

Studi kasus yang akan dibawa pada laporan kali ini bertujuan untuk mendeteksi penipuan kartu kredit, sebelumnya kita memerlukan data terlebih dahulu agar bisa melakukan *preprocessing data*, kita akan memakai dataset publik yang berisi informasi numerik yang berasal dari transformasi PCA, terdapat beberapa variabel didalam dataset tersebut yaitu V1 sampai V28 adalah data rahasia yang sudah diubah dengan transformasi PCA serta terdapat variabel waktu dan jumlah. Pada dataset tersebut juga terdapat variabel kelas dimana jika bernilai satu maka terjadi penipuan sedangkan jika bernilai nol maka terdapat kasus lain.

Sebelum melakukan permodelan, data akan dilakukan *preprocessing* dengan 80% digunakan sebagai *data training* dan 20% sebagai *data testing*. Setelah melakukan *preprocessing* simulasi akan dilakukan dengan mendefiniskan terlebih dahulu model ANN menggunakan Keras pada Python. Pada simulasi pertama model ANN yang dimana terdapat 3 *hidden layer* dengan fungsi aktivasinya ReLU dan 1 output dengan fungsi aktivasinya sigmoid.

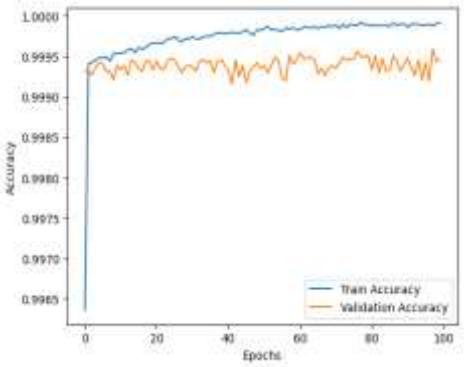
Pada *hidden layer* pertama digunakan 64 *nodes*, *hidden layer* kedua digunakan sebanyak 32 *nodes*, dan terakhir untuk *hidden layer* ketiga digunakan 16 *nodes*. Penggunaan jumlah nodes tersebut bertujuan karena 64-32-16 memberikan keseimbangan yang baik antara kapasitas model dengan efisiensi komputasi.

Selanjutnya, model dikompilasi dengan *optimizer* Adam yang bertujuan agar model dapat secara efisien untuk menyesuaikan learning rate dan model dapat mencapai konvergensi lebih cepat. Selain dikompilasi dengan *optimizer* Adam, model juga dikompilasi dengan *loss function binary* dikarenakan data memang diklasifikasikan melalui biner.

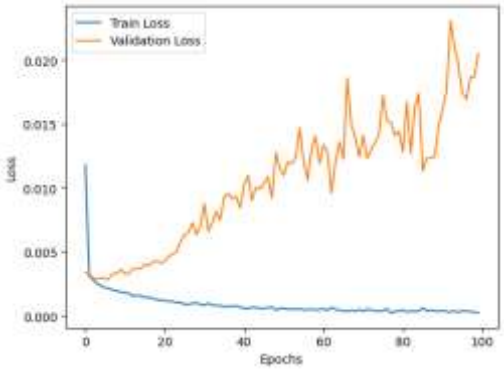


Gambar 1 Proses pelatihan dengan *epoch* 100 dan *batch size* 48

Dengan *epoch* sebanyak 100 dan *batch size* sebanyak 48, kita bisa mengetahui bahwa *train loss* jauh lebih dikit dibanding *validation loss*, bukan hanya itu, kita juga mengetahui bahwa tingkat akurasi pada *train accuracy* dan *validation accuracy* cukup mendapatkan gap yang besar. Ini berarti data yang digunakan untuk dataset memiliki ketidakseimbangan data, hal ini membuat gap antara data *train* dan *validation* besar.

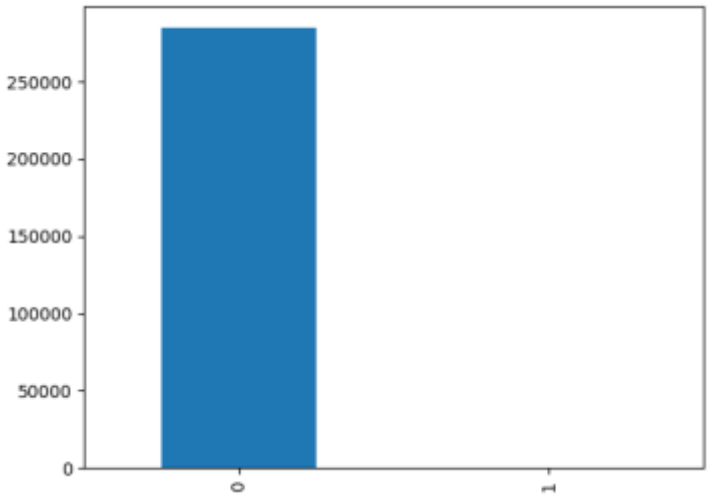


Gambar 2 menunjukkan nilai akurasi terhadap *data train* dan *data validation* memiliki gap yang jauh.



Gambar 3 menunjukkan nilai *loss* terhadap *data train* dan *data validation* memiliki gap yang jauh.

Dengan ketidakseimbangan kelas yang dimiliki, kita perlu untuk mengatasi hal tersebut, dengan beberapa langkah yang bisa dilakukan, diantaranya adalah penggunaan validasi silang atau *cross-validation*, bisa juga melakukan teknik *sampling*, dan pergantian penilaian akurasi yang sebelumnya akurasi dinilai saat *training* bisa diganti dengan dilakukannya penilaian akurasi menggunakan *Area Under the Precision-Recall Curve.* Kita juga perlu untuk menguji data dengan *epoch* yang berbeda dan *batch size* yang berbeda untuk mencari titik optimal dimana data bisa dilatih dengan akurat.

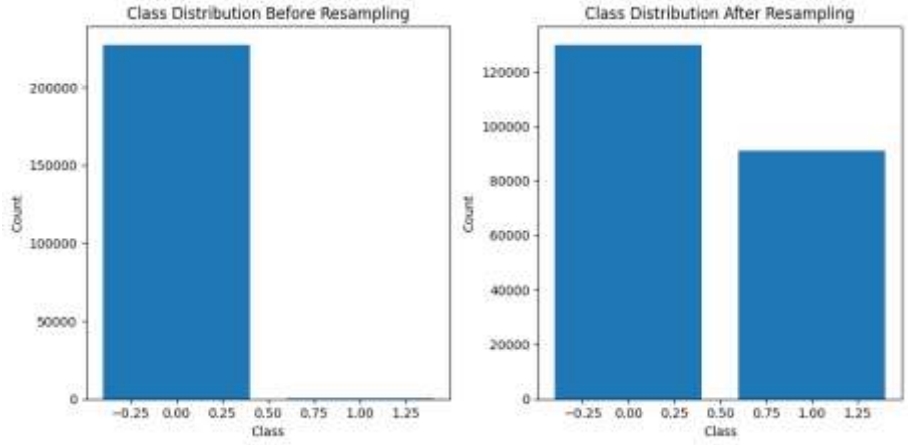


Gambar 4 menunjukkan *overfitting* data dimana klasifikasi *class* 0 terdapat sebanyak 227.454 contoh sedangkan *class* 1 hanya sebanyak 391 contoh.

1. Menggunakan Teknik *Resampling*

Teknik resampling diperlukan untuk mengatasi kelemahan pada dataset yang mengalami *overfitting*. Salah satu teknik *resampling* yang umum digunakan adalah metode

SMOTE (*Synthetic Minority Over-sampling Technique*). SMOTE bekerja dengan menciptakan sampel sintetis baru untuk kelas minoritas dalam dataset, membantu menyeimbangkan distribusi kelas dan mengurangi risiko *overfitting* pada model pembelajaran mesin.

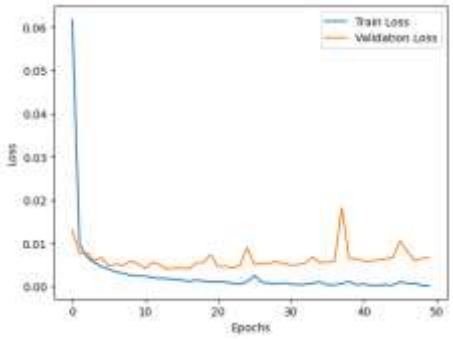


Gambar 5 Perubahan distribusi *class* setelah menggunakan teknik *resampling*

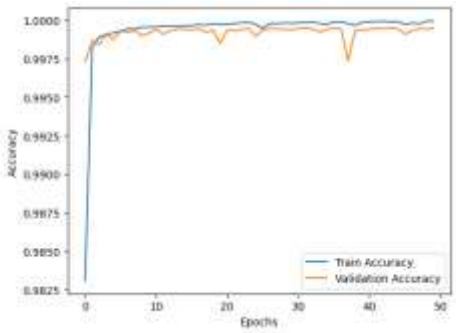
2. Penggunaan *cross-validation*

Sebelumnya kita memakai teknis validasi train-test-validation split dan akan diubah menjadi *cross-validation* dengan alasan untuk meningkatkan keandalan evaluasi model serta meminimalkan risiko *overfitting* yang terdapat pada data yang dimiliki. Dengan *cross-validation*, data dibagi menjadi beberapa subset yang digunakan secara bergantian sebagai data pelatihan dan pengujian, memberikan setiap titik data kesempatan untuk menjadi bagian dari set validasi. Hal ini memungkinkan evaluasi model yang lebih komprehensif terhadap berbagai data.

Karena dataset pengujian memiliki jumlah data yang besar, peningkatan *batch size* menjadi 128 dan penurunan *epoch* menjadi 50 akan dilakukan. Hal ini bertujuan untuk meningkatkan efisiensi komputasi dan mempercepat waktu pelatihan atau pengujian model sambil tetap mempertahankan akurasi.



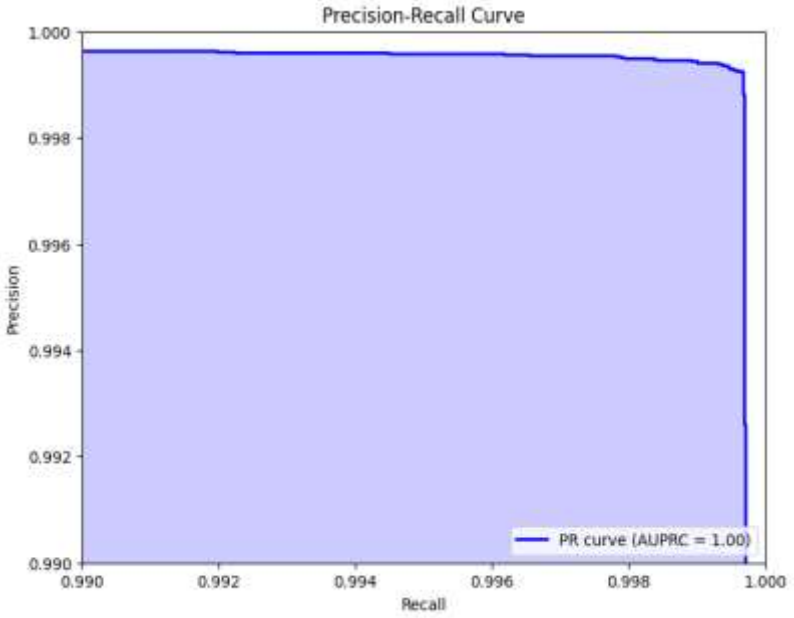
Gambar 6 menunjukkan nilai *loss* terhadap *data train* dan *data validation* yang dimana gapnya berkuran dengan teknik *resampling*.



Gambar 7 menunjukkan nilai akurasi terhadap *data train* dan *data validation* yang dimana gapnya berkurang dengan teknik *resampling*.

3. Pengukuran Akurasi menggunakan *Area Under the Precision-Recall Curve* (AUPRC)

Karena ketidakseimbangan kelas yang terdapat dalam dataset, pengukuran akurasi biasa tidak akan efektif untuk mengevaluasi keseluruhan kinerja model. Sedangkan, AUPRC memberikan cara yang lebih baik untuk mengevaluasi kinerja model karena memperhitungkan *presisi* dan *recall*, sehingga lebih cocok untuk situasi ketidakseimbangan kelas. Oleh karena itu penggunaan AUPRC pada dataset yang tidak seimbang menjadi pilihan yang baik dalam pengukuran kinerja model.



Gambar 8 Nilai AUC-ROC (0.98) menunjukkan bahwa model mampu membedakan antara data positif dan negatif dengan tingkat akurasi yang tinggi.

.

4. Penggunaan *Confusion Matrix*

Dengan dataset yang mengklasifikasikan antara dua kasus pada kartu kredit, yaitu transaksi yang normal dan transaksi yang mencurigakan (*fraud*), *confusion matrix* menjadi pilihan yang tepat untuk mengukur performa pada model pembelajaran mesin. Dalam kasus ini, *confusion matrix* membantu dalam mengevaluasi seberapa baik model dapat membedakan antara transaksi yang sah dan transaksi yang mencurigakan.

Dalam konteks dataset kartu kredit, *confusion matrix* dapat membantu kita untuk mengetahui seberapa baik model dapat mendeteksi transaksi *fraud* (TP), seberapa sering transaksi normal salah diidentifikasi sebagai *fraud* (FP), dan seberapa sering transaksi *fraud* salah diidentifikasi sebagai normal (FN). Dengan demikian, *confusion matrix* memberikan wawasan yang berharga tentang seberapa efektif model dalam membedakan transaksi yang sah dan mencurigakan, yang merupakan informasi penting dalam pengelolaan risiko keamanan kartu kredit.

Gambar 9 menunjukkan hasil *confusion matriks* yang baik. Ini menjadi tanda bahwa model dapat mengklasifikasikan data dengan tingkat presisi yang tinggi.

Pada gambar 8, model dapat memiliki tingkat akurasi yang tinggi dengan perhitungan nilai akurasi seperti dibawah ini.

$$Akurasi = \frac{TP + TN}{TP + FP + FN + TN}$$

Dengan perhitungan menggunakan rumus tersebut didapat bahwa nilai akurasi dari model sebanyak 99.95%. Ini menunjukkan bahwa model memiliki kinerja yang baik. Bukan hanya pada nilai akurasi, pada nilai precision dan recall yang tinggi menunjukkan bahwa model tidak hanya baik untuk mengindentifikasi nilai yang positif, tetapi juga tidak salah dalam mengidentifikasi nilai negatif yang dianggap sebagai nilai positif.

C. KESIMPULAN

1. Model ANN didefinisikan dengan tiga *hidden layer* menggunakan fungsi aktivasi ReLU dan satu output dengan fungsi aktivasi sigmoid.
2. Model dikompilasi dengan *optimizer* Adam dan *loss function binary*. Dengan parameter khusus seperti jumlah *epoch* dan *batch size* untuk mencapai konvergensi dengan cepat.
3. Pengenalan data yang tidak seimbang diperlukan untuk menangani permasalahan *overfitting* pada data, solusi yang diusulkan berupa teknik *oversampling* dengan metode SMOTE.
4. Evaluasi yang dilakukan tidak hanya melibatkan pengukuran akurasi, tetapi dipertimbangkan menggunakan metrik lain seperti AUPRC dan *Confusion Matrix*.

REFERENCE

Brownlee, J. (2021, 3 17). *Imbalanced Classification*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

DR. MARIA SUSAN ANGGREANY, S. M. (2022, 11 01). *Confusion Matrix*. Retrieved from Binus University: https://socs.binus.ac.id/2020/11/01/confusion-matrix/

Keras. (n.d.). *Adam*. Retrieved from Keras: https://keras.io/api/optimizers/adam/

scikit-learn. (n.d.). *scikit-learn*. Retrieved from https://scikit-learn.org/

APPENDIKS

https://colab.research.google.com/drive/1QQtX0FDrMwyChP8YbH9lMZe6x9YG-5GV?usp=sharing