Penerapan *Deep Neural Network* dengan *Dropout* dan *Cost-Sensitive Learning* untuk Prediksi Penyakit Stroke

Cynthia Caroline[#1], Ventje J. Lewi Engel, M.T., CEH[*2]

[#]*Program Studi Informatika, Institut Teknologi Harapan Bangsa*
Jalan Dipatiukur No. 80-84, Bandung, Indonesia 40132
[1]Cynthiacc512@gmail.com
[2]ventje@ithb.ac.id

Abstract*— A stroke is a condition that occurs when the blood supply to the brain is reduced. There are 2 types of stroke, namely ischemic stroke caused by blockage and hemorrhagic stroke caused by blood vessel rupture. Without blood, the brain cannot receive oxygen, so the affected cells will die immediately. In this study, a stroke prediction model will be built using a Deep Neural Network with dropout and Cost-Sensitive Learning and Probability Tuning to handle imbalanced classes from the dataset. The dataset used in this study is the Cerebral Stroke Prediction- Imbalanced Dataset sourced from Kaggle. This data will be preprocessed before being further processed into a model. The highest accuracy result achieved by the Deep Neural Network model is 84,50% using a learning rate of 0.01, hidden layer 3, epoch 100, and activation function tanh. The highest ROC result achieved by the Deep Neural Network model is 0,5303 using a learning rate of 0,01, hidden layer 4, epoch 10, and activation function tanh

***Keywords*— Stroke, Deep Neural Network, Dropout, Cost-Sensitive Learning, Probability Tuning.**

***Abstrak*— Stroke adalah kondisi yang terjadi ketika pasokan darah ke otak berkurang. Stroke terbagi 2 jenis, yaitu stroke iskemik yang disebabkan akibat penyumbatan dan storoke hemoragik yang disebabkan akibat pecahnya pembuluh darah. Tanpa darah, otak tidak dapat mendapat asupan oksigen sehingga sel yang terdampak akan segera mati. Pada penelitian ini, akan dibangun model prediksi penyakit stroke dengan menggunakan *Deep Neural Network* dengan *dropout* dan *Cost-Sensitive Learning* serta *Probability Tuning* untuk melakukan *handling* terhadap kelas yang tidak seimbang dari dataset. Dataset yang digunakan dalam penelitian ini adalah *Cerebral Stroke Prediction-Imbalanced Dataset* yang bersumber dari Kaggle. Data ini akan dilakukan *preprocessing* terlebih dahulu sebelum diolah lebih lanjut menjadi suatu model. Hasil akurasi tertinggi diraih oleh model *Deep Neural Network* adalah 84,50% dengan menggunakan *learning rate* 0,01, *hidden layer* 3, epoch 100, dan *activation function* tanh. Hasil ROC tertinggi diraih oleh model *Deep Neural Network* adalah 0,5303 dengan menggunakan *learning rate* 0,01, *hidden layer* 4, *epoch* 10, dan *activation function* tanh**

***Kata Kunci*— Stroke, *Deep Neural Network, Dropout, Cost-Sensitive Learning, Probability Tuning*.**

I. Pendahuluan

Stroke adalah kondisi yang terjadi ketika pasokan darah ke otak berkurang akibat penyumbatan (stroke iskemik) atau pecahnya pembuluh darah (stroke hemoragik) [1]. Tanpa darah, otak tidak akan mendapatkan asupan oksigen dan nutrisi, sehingga sel-sel pada area otak yang terdampak akan segera mati. Stroke dapat disebabkan oleh sejumlah faktor, yaitu faktor yang tidak dapat dimodifikasi, seperti berat badan saat lahir, gender, umur, dan etnis seseorang, faktor penyakit seperti hipertensi, belumnya pernah menderita stroke, mengalami *atrial fibrillation*, kelainan jantung, perubahan lipid, gangguan koagulasi, homosistein, diabetes mellitus, migrain, infeksi, *sleep apnea*, penyakit ginjal, dan penyakit arteri, dan faktor gaya hidup seperti, merokok, minum minuman beralkohol, obesitas, aktivitas fisik, diet, terapi hormon, stress, dan faktor sosial ekonomi [2].

II. Metodologi.

A. Deep Neural Network (DNN)

Neural network adalah sebuah arsitektur yang cara kerjanya terinspirasi dari cara kerja otak. Otak terdiri dari kumpulan neuron yang saling terhubung. Setiap *neuron* menerima input dari output neuron lain dan kemudian melakukan perhitungan. *Neural network* terdiri dari kumpulan *perceptron* [3]. Perceptron adalah bagian terkecil dari arsitektur neural network [3]. Perbedaan *Neural Network* dengan *Deep Neural Network* adalah pada jumlah *hidden layer-nya*. Dalam Buku [4] disebutkan bahwa *hidden layer* 2 sudah termaksud deep, namun saat ini, sudah umum arsitektur DNN yang memiliki puluhan bahkan ratusan *hidden layer*, sehingga definisi deep sendiri sudah cukup kabur. Menurut Penelitian [5] disebutkan bahwa jika jumlah *hidden layer* lebih dari 2, sudah termaksud *Deep Neural Network*, dan dijelaskan bahwa *Neural Network* dengan lebih dari 1 *hidden layer* akan menjadi kompleks.

B. *Dropout*

Dropout merupakan salah satu teknik *regularization* yang bekerja dengan cara menonaktifkan neuron dalam *neural network* secara acak, dengan tujuan mengurangi *overfitting*. Algoritme dropout adalah, dalam setiap fase training, semua neuron (kecuali neuron output), mempunyai probabilitas p yang sementara diputus, dengan maksud akan diabaikan terlebih dahulu selama fase *training* kali ini, tetapi mungkin akan aktif saat fase berikutnya [6].

C. *Cost-Sensitive Learning*

Cost-sensitive learning adalah sebuah metode yang memperhitungkan kesalahan prediksi saat *training* model sebagai *cost*. *Cost-sensitive learning* cocok digunakan untuk masalah yang lebih mementingkan *false negative*. *Cost-sensitive learning* juga berusaha mengurangi kesalahan saat training data, yang disebut *error minimization*. Tujuan *cost-sensitive learning* adalah meminimalkan *cost* saat *training* dataset [7]. Cost-sensitive learning untuk *imbalance class* difokuskan pertama-tama adalah menetapkan *cost* yang berbeda untuk setiap jenis kesalahan klasifikasi, kemudian menggunakan metode khusus untuk menghitung *cost* tersebut. Untuk menghitung kesalahan klasifikasi dapat menggunakan cost matrix yang didasari oleh confusion matrix [8]. Algoritme *cost-sensitive learning* dinilai cocok untuk digunakan, karena algoritme ini dapat mengubah-ubah weight agar cost pada kelas mayoritas lebih kecil daripada *cost* pada kelas minoritas.

D. *Probability Tuning*

Probability tuning dapat mengurangi *overfitting* karena pada saat memakai metode pengukuran ROC *curve*, kita bisa menghitung nilai G-Mean terbesar yang akan digunakan untuk menentukan *threshold* terbesar. *Probability tuning* akan digunakan bersamaan dengan ROC *curve* dengan menghitung *true positive rate* dan *false positive rate*.

E. *Receiver Operating Characteristic* (ROC) *Curve*

Receiver Operating Characteristic (ROC) curve adalah metode untuk melakukan pengukuran, yang tujuannya melakukan *plotting true positive rate* (TPR) atau *recall* atau *sensitivity* dengan *false positive rate* (FPR). FPR sendiri adalah rasio negatif yang salah diklasifikasikan sebagai kelas positif. FPR diperoleh dari perhitungan 1 − specificity [9]. G-Mean atau *Geometric Mean* adalah sebuah pengukuran yang berfungsi untuk mengukur kelas *imbalanced* yang menyeimbangkan antara *sensitivity* dan *specificity*. Metode ROC AUC digunakan dalam penelitian ini dengan alasan AUC merupakan *classification-threshold-invariant*, artinya AUC mengukur kualitas prediksi model, terlepas *threshold* klasifikasi apapun yang dipilih [10]. Metode pengukuran *precision* dan *recall* kurang cocok untuk kasus *imbalanced* class, karena metode *precision* dan *recall* lebih berfokus kepada kelas minoritas saja, sedangkan ROC *curve* mencakup kedua kelas [11]. Dalam kasus ini, tidak hanya kelas positif (menderita stroke) saja yang penting, namun kedua kelas penting agar model bisa mengenali pola penderita stroke dan bukan penderita stroke dengan baik.

F. *Learning Curves*

Learning curves adalah kurva yang mengukur model berdasarkan performanya. *Learning curves* digunakan untuk mengukur hasil dari *training* model pada *machine learning* secara bertahap. Dalam *learning curves*, terdapat 2 grafik, yaitu:

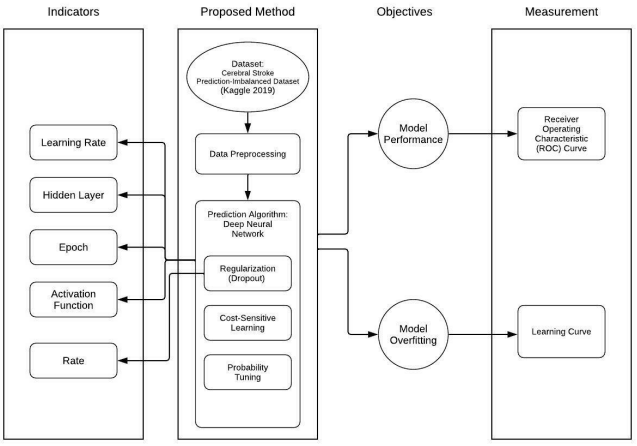
1. *Train learning curve*, yaitu *learning curve*s yang dihitung dari *dataset training* yang berfungsi untuk menggambarkan seberapa baik model belajar.
2. *Validation learning curve*, yaitu *learning curves* yang dihitung dari dataset *validation* yang berfungsi untuk menggambarkan seberapa baik model digeneralisasi.

III. Perancangan Sistem

A. Kerangka Pemikiran

Gambar 1 menunjukkan kerangka pemikiran dari metode yang diusulkan untuk melakukan prediksi penyakit stroke. Berikut merupakan beberapa indikator yang akan diuji:

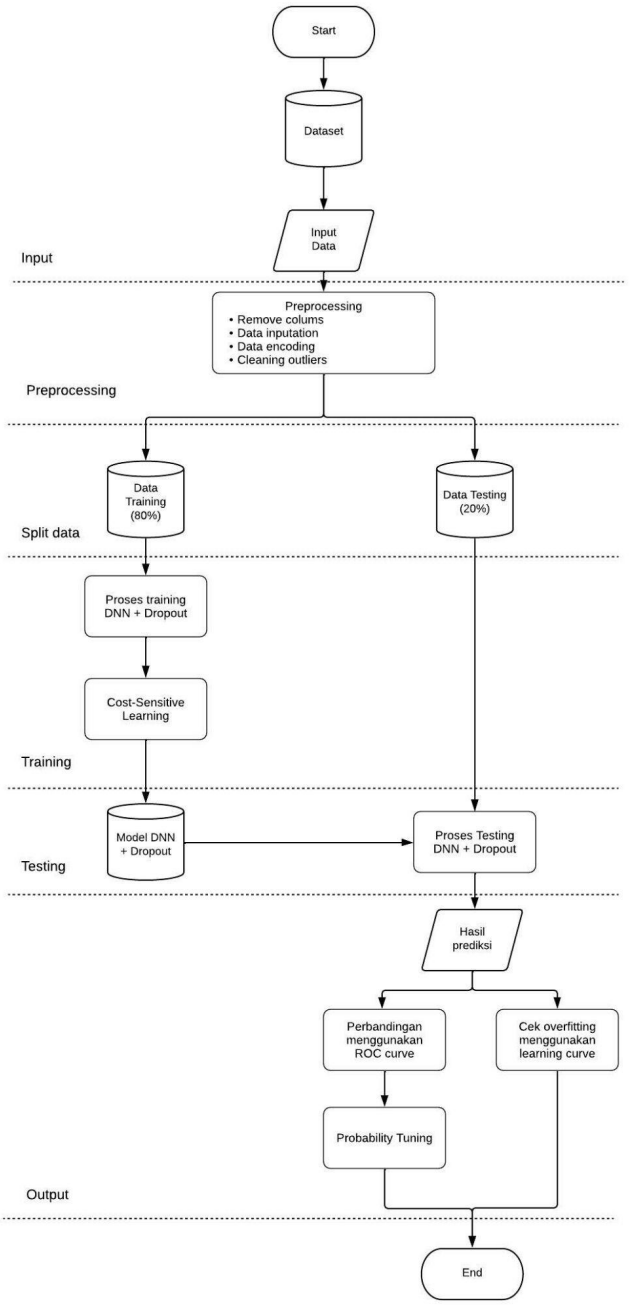
1) *Learning rate*, berfungsi untuk mengatur seberapa besar model melakukan update *weight*. Semakin kecil model dapat konvergen, tetapi diperlukan epoch yang besar, otomatis waktunya akan semakin lama. Jika terlalu besar, model tidak dapat konvergen.
2) *Hidden layer*, berfungsi untuk pola-pola yang tidak terlihat di *neural network*. Semakin banyak, maka akan semakin lambat, tetapi bisa menemukan pola yang kompleks.
3) *Epoch*, merupakan iterasi 1 siklus program selesai dijalankan. Semakin besar semakin bisa meningkatkan akurasi, namun akan semakin lama, dan dapat menyebabkan *overfitting*
4) *Activation function*, berfungsi untuk menentukan *output* dari *neural network*, dengan range 0 sampai 1, -1 sampai 1, 0 sampai x. Nilai *range* tergantung dari masing-masing activation function.
5) *Rate*, merupakan probabilitas mempertahankan unit. Probabilitas = 1 artinya tidak ada *dropout*, dan semakin kecil nilai probabilitasnya, semakin banyak *neuron* yang didropout. Semakin kecil nilai probabilitas, artinya semakin membutuhkan banyak neuron yang otomatis akan memperlambat *training* dan hasilnya akan cenderung *underfitting*.



Gambar 1 Kerangka Pemikiran

B. *Flowchart Global*

Model prediksi stroke ini dibangun menggunakan algoritma DNN dengan menggunakan *regularization dropout*. Setelah dilakukan *training*, model diharapkan dapat memprediksi kemungkinan orang yang terkena stroke dengan akurat dan cepat. Seperti pada Gambar 3.2, dimulai dari *preprocessing* dataset, lalu melakukan *training* dan *testing* untuk membuat model. Setelah model selesai dibuat, maka akan dilakukan proses *testing* dengan data yang bersumber dari data *testing*. Jika proses *testing* selesai, maka sistem akan menghasilkan prediksi, yang akan dicek performanya menggunakan ROC *curve*. Gambar 2 menunjukkan *flowchart* global pada model yang akan dibuat.



Gambar 2 *Flowchart* Global

C. *Dataset*

Data yang didapatkan merupakan dataset bernama *Cerebral Stroke Prediction-Imbalanced Dataset* yang didapatkan dari Mendeley Data dan diterbitkan di situs Kaggle [12]. Dataset berjumlah 43.400 data dan memiliki 12 variabel yaitu *id, gender, age, hypertension, heart disease, ever married, work type, residence type, avg glucose level, bmi, smoking status*, dan *stroke*. Isi data dalam dataset ini merupakan data asli yang diambil dari situs *HealthData.gov. HealthData.gov* adalah sebuah situs web pemerintah Amerika Serikat yang dikelola oleh *U.S. Department of Health & Human Services*. Dataset

ini bersifat *imbalance* dengan data orang yang menderita stroke sebesar 783 dan data orang yang tidak menderita penyakit stroke sebesar 42.617 data.

IV. Perancangan Sistem

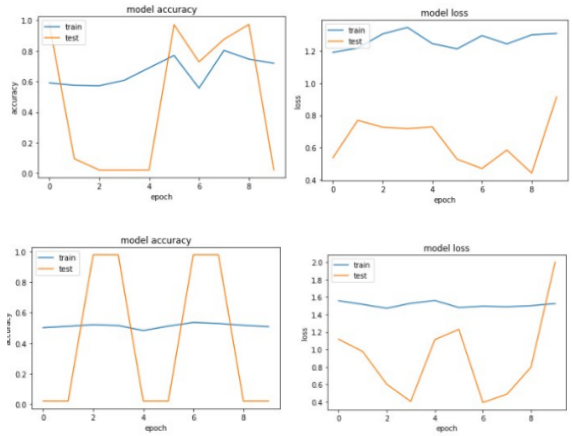
Tabel I
Kombinasi Hyperparameter yang Akan Diuji

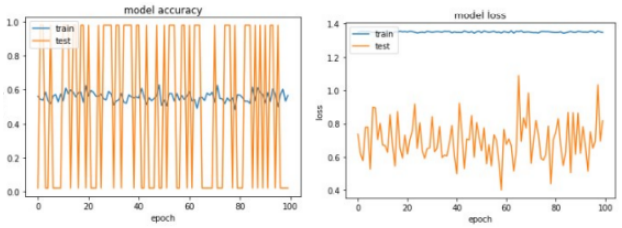
	Learning Rate	***Hidden Layer***	***Epoch***	***Activation Function***
	0,1	5	10	ReLu
	0,01	10	50	*Tanh*
		20	100	
Total	**2**	**3**	**3**	**2**
Total Pengujian	= 2 * 3* 3* 2 = **36**			

Pengujian pertama akan dilakukan hanya menggunakan algoritme *Deep Neural Network* dengan menggunakan 4 jenis *hyperparameter* dengan jumlah kombinasi sebesar 36 kombinasi. Pengujian selanjutnya yaitu menguji algoritme *Deep Neural Network* dengan *dropout* dengan menggunakan 5 jenis *hyperparameter*. Kombinasi yang akan digunakan dalam pengujian ini adalah 4 hasil kombinasi terbaik dari percobaan sebelumnya, lalu ditambahkan dengan *rate*, sehinggal dalam pengujian ini terdapat sebesar 8 kombinasi.

A. *Pengujian Deep Neural Network*

Pada pengujian menggunakan metode *Deep Neural Network*, yang ditambah *cost-sensitive learning*, hasil menunjukkan bahwa akurasi paling tinggi adalah 97,89% dan paling rendah adalah 2,11%. Pengujian menunjukkan tidak ada perubahan walaupun *learning rate, hidden layer, epoch,* dan *activation function* diganti. Teknik *probability tuning* tidak dapat digunakan dalam pengujian kali ini, dikarenakan setiap pengujian terdapat kasus TPR atau FPR yang berisi nan, sehingga tidak bisa dilakukan perhitungan G-Mean dan ROC. Di bawah ini, terdapat gambar *learning curve* dari 3 model.





Gambar 3 *Learning Curve model*

B. *Pengujian Deep Neural Network dengan Dropout*

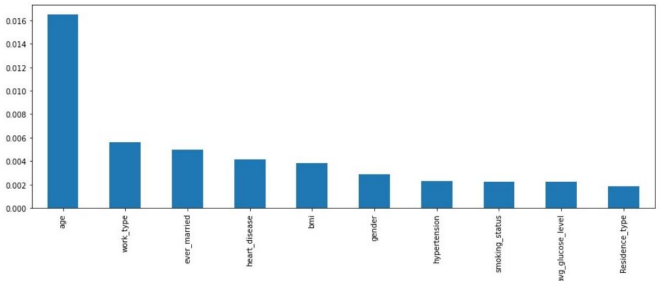
Pada pengujian menggunakan metode Deep Neural Network, yang ditambah *cost-sensitive learning*, hasil menunjukkan bahwa akurasi paling tinggi adalah 97,89% dan paling rendah adalah 20,84%. Namun, akurasi 97,89% tersebut memiliki ROC dan G-Mean nan, sehingga teknik *probability tuning* tidak bisa dilakukan dan dalam c*onfusion matrix*, model tersebut memiliki kecenderungan *overfitting*.

Tabel II
Hasil pengujian Deep Neural Network dengan Dropout

Rate	**ROC**	**Akurasi**
0,1	nan	97,89%
0,1	*nan*	*97,89%*
0,01	*0,4981*	*20,84%*

C. *Pengujian Tambahan dengan Feature Selection Information Gain*

Hasil akurasi tertinggi adalah 97,89%, namun hasil dari confusion matrix model tersebut menunjukkan bahwa model sangat *overfitting*. Model lain tidak *overfitting*, namun memberikan hasil yang sangat buruk. Akurasi tertinggi model hanya 50,37% dan ROC tertinggi hanya 50,28%. Dalam kasus prediksi stroke, yang hanya memprediksi 2 kemungkinan (sakit stroke atau tidak sakit stroke), peluang 50% tentu sama saja dengan menebak. Hal ini mungkin saja terjadi dikarenakan fitur yang dipakai terlalu banyak dan tidak relevan untuk prediksi stroke. Oleh karena itu, akan dilakukan *feature selection* dengan metode *information gain*. Gambar 3 di bawah ini menunjukkan hasil variabel yang paling berpengaruh.

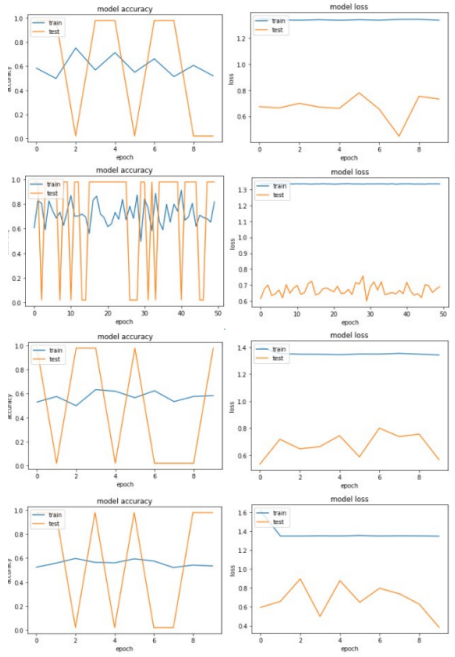


Gambar 4 Hasil variabel yang paling berpengaruh dari *information gain*

Gambar 4 di atas menunjukkan bahwa fitur age adalah fitur yang sangat berpengaruh terhadap prediksi stroke. Fitur selanjutnya masih memiliki pengaruh, namun sudah tidak signifikan terhadap prediksi stroke. Berdasarkan diagram di atas maka hanya akan diambil 5 fitur terbaik saja, dikarenakan

fitur selanjutnya sudah semakin tidak signifikan. Pengujian ini mengambil 4 hasil terunik dari model sebelumnya yang didasari *learning curve.*

1) Pengujian DNN dengan *information gain*, mendapatkan hasil akurasi hanya di kombinasi pertama yang memiliki perubahan drastis, sedangkan ROC-nya tidak mengalami perubahan, namun kombinasi lainnya memiliki nilai yang sama. Di bawah ini, terdapat gambar *learning curve* untuk percobaan yang sudah dilakukan.



Gambar 5 Perbandingan Learning curve untuk setiap percobaan

2) Pengujian DNN dan *Dropout* dengan *information gain*, mendapatkan hasil akurasi dan ROC yang cenderung meningkat, namun tidak signifikan. Hal ini mungkin saja terjadi dikarenakan penggunaan *hidden layer* yang terlalu banyak yang mengakibatkan model mengalami *overfitting*. Oleh sebab itu, akan dilakukan kembali pengujian dengan *hidden layer* yang lebih kecil untuk menghindari overfitting

D. *Pengujian Tambahan dengan Hidden Layer yang Lebih Kecil*

TABEL III
KOMBINASI HYPERPARAMETER YANG AKAN DIUJI

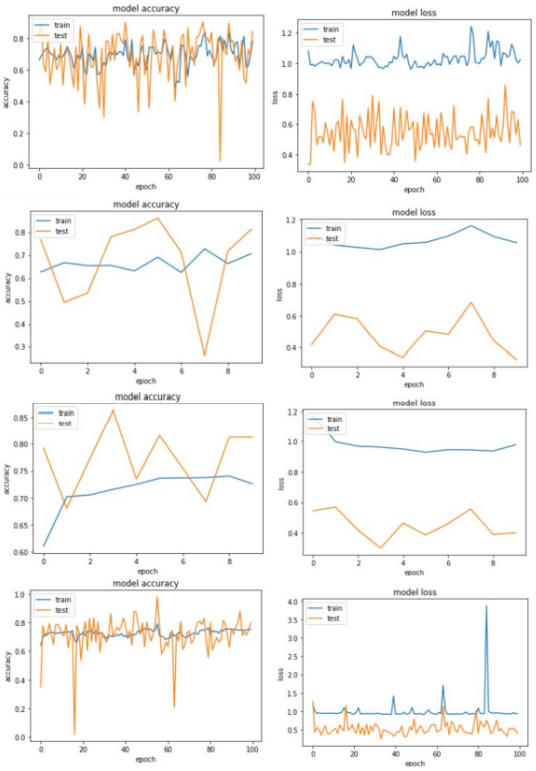
	Learning Rate	***Hidden Layer***	***Epoch***	***Activation Function***
	0,1	2	10	ReLu
	0,01	3	50	*Tanh*
		4	100	
Total	**2**	**3**	**3**	**2**
Total Pengujian	= 2 * 3* 3* 2 = **36**			

1) Pengujian *Deep Neural Network*, hasil menunjukkan bahwa pada saat *learning rate* 0,01 akurasi paling tinggi adalah 97,89%, namun akurasi 97,89% merupakan hasil model yang *overfitting*. Akurasi model yang tinggi namun tidak mengalami *overfitting* adalah kombinasi ke-11, yaitu dengan akurasi 84,50%, sedangkan akurasi paling rendah 2,11% diraih oleh 3 kombinasi. Hal ini dikarenakan *epoch* yang terlalu besar pada *hidden layer* yang tergolong cukup besar, sehingga model tidak bisa melakukan prediksi dengan benar. Pada *learning rate* 0,1, model tidak dapat melakukan prediksi dengan baik. Hal ini dikarenakan *learning rate* yang terlalu besar yang menyebabkan model tidak dapat konvergen. Beberapa hasil ROC dan akurasi tertinggi dapat dilihat pada tabel IV di bawah ini.

TABEL IV
HASIL PENGUJIAN DEEP NEURAL NETWORK

No.	**ROC**	**Akurasi**
11	0,5301	0,8450
13	0,5303	0,8130
14	0,5266	0,8124
18	0,5204	0,7981

Di bawah ini, terdapat gambar perbandingan *learning curve* untuk setiap model.



Gambar 6 Perbandingan Learning curve untuk setiap percobaan

2) Pengujian *Deep Neural Network* dan *Dropout*, hasil menunjukkan bahwa akurasi paling tinggi adalah 78,74% dan paling rendah adalah 20,77%. Untuk akurasi 3 model tertinggi diraih oleh *activation function* tanh, dan semua model yang menggunakan *activation function* ReLu memiliki hasil akurasi yang buruk, terutama saat menggunakan *rate* 0,8.

Beberapa hasil ROC dan akurasi tertinggi dapat dilihat pada tabel V di bawah ini.

TABEL V
HASIL PENGUJIAN DEEP NEURAL NETWORK DENGAN DROPOUT

No.	ROC	Akurasi
1	0,5017	0,7510
2	0,4999	0,6879
3	0,5012	0,7874
5	0,5017	0,5009

E. *Pengujian Tanpa Cost-Sensitive Learning dan Probability Tuning*

Pengujian ini dilakukan agar dapat membandingkan hasil dari algoritme DNN dan *dropout* saja dengan hasil dari algoritme DNN dan dropout yang dipadukan dengan teknik *cost-sensitive learning* dan *probability tuning*. Hasil dari pengujian ini untuk metode DNN mengalami *overfitting*, dengan akurasi 0,9789% dan ROC yang tidak bisa dihitung. Namun disisi lain, metode DNN dan *dropout* tidak mengalami *overfitting.*

V. SIMPULAN

Penelitian ini mendapatkan kombinasi hyperparameter terbaik dalam penelitian ini adalah *learning rate* 0,01, *hidden layer* 3, *epoch* 100, dan *activation function* tanh dengan akurasi sebesar 84,50%.

Metode *dropout* memengaruhi hasil ROC dan akurasi model (tanpa menggunakan *cost-sensitive learning* dan probability tuning). Akurasi tertinggi sebesar 75,11% dan ROC tertinggi sebesar 0,5017, sedangkan jika tidak menggunakan *dropout*, model mendapat akurasi hingga 97,89% tetapi ROC tidak dapat dihitung dan model tersebut mengalami *overfitting*.

Metode *cost sensitive learning* dan *probability tuning* membantu menyelesaikan masalah *overfitting* hanya saat menggunakan algoritme DNN. Metode *cost-sensitive learning* mendapatkan akurasi tertinggi 84,50% dan tanpa metode *cost-sensitive learning* mendapat akurasi 97,89% namun mengalami *overfitting*. Jika menggunakan algoritme DNN dan *dropout*, maka hasil akurasi terbaiknya sama, yaitu sebesar 78,74%.

Nilai ROC terbaik pada model prediksi stroke adalah 0,5303 dengan kombinasi *hyperparameter learning rate* 0,01, *hidden layer* 4, *epoch* 10, dan *activation function* tanh.

Hidden layer yang terlalu besar dapat membuat hasil prediksi menjadi *overfitting*, seperti apa yang terjadi dalam beberapa percobaan model ini.

Penambahan metode *feature selection information gain* hampir tidak berpengaruh pada akurasi dan ROC.

Hasil akurasi dan ROC model yang menggunakan *cost-sensitive learning* dan *probability tuning* dinilai lebih baik tanpa menggunakan *dropout*.

Sangat sulit untuk mendapatkan akurasi lebih dari 90% yang tidak mengalami *overfitting*, dikarenakan penyakit stroke dapat disebabkan dari berbagai faktor yang mungkin saja tidak tercatat di dataset yang dipakai.

DAFTAR REFERENSI

[1] C. Chugh, ”Acute Ischemic Stroke: Management Approach” Indian Journal of Critical Care Medicine, vol. 23, pp. S140–S146, 2019..

[2] B. Norrving, Stroke and Cerebrovascular Disorders, Oxford, 2014.

[3] A. Geron, ´ Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow 2nd Edition, O’Reilly Media, Inc, 2019.

[4] S. Haykin, Neural Networks and Learning Machines Third Edition, Pearson, 2009.

[5] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, H. Arshad, ”State-of-the-art in artificial neural network applications: A survey,” Heliyon, vol. 4, no. 11, 2018.

[6] J. Grus, Data Science from Scratch, O’Reilly Media, Inc, 2015.

[7] A. Fernandez, S. Garc ´ ´ıa, M. Galar, R. C. Prati, B Krawczyk, and F. Herrera, ”Learning from Imbalanced Data Sets,” Springer, 2014.

[8] J. Brownlee. ”Cost-Sensitive Learning for Imbalanced Classification”. Machine Learning Mastery. 2020. [Online]. Available: https://machinelearningmastery.com/cost-sensitive-learning-for-imbalanced-classification/

[9] S. Haykin, Neural Networks and Learning Machines Third Edition, Pearson, 2009

[10] K. Munir, H. Elahi, A. Ayub, F. Frezza, A. Rizzi, ”Cancer Diagnosis Using Deep Learning: A Bibliographic Review,” Cancers, vol. 11, no. 9, pp. 1235, 2019.

[11] Y. Ma, H. He., Imbalanced Learning: Foundations, Algorithms, and Applications 1st Edition, Wiley, 2013.

[12] S. Tiwari, 2019. Cerebral Stroke Prediction-Imbalanced Dataset. [Online]. Available: https://www.kaggle.com/shashwatwork/cerebral-stroke-predictionimbalaced-dataset [Accessed: Oct 30, 2021].

Cynthia Caroline, menerima gelar Sarjana Komputer pada program studi Informatika di Institut Teknologi Harapan Bangsa (ITHB). Saat ini bekerja sebagai *UI/UX designer* di salah satu perusahaan startup.

Ventje Jeremias Lewi Engel, menerima Sarjana Teknik dari Insitut Teknologi Bandung (ITB) pada tahun 2012 dan Magister Teknik dari Institut Teknologi Bandung pada tahun 2013. Aktif sebagai dosen di Prodi Informatika Institut Teknologi Harapan Bangsa (ITHB). Minat penelitian pada bidang *Deep Learning, Cybersecurity*, dan *Malware Analysis.*