Terakreditasi SINTA Peringkat 2
Surat Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti No. 10/E/KPT/2019
masa berlaku mulai Vol. 1 No. 1 tahun 2017 s.d Vol. 5 No. 3 tahun 2021

Terbit online pada laman web jurnal: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 5 No. 2 (2021) 203 - 212 ISSN Media Elektronik: 2580-0760

Penerapan *Deep Learning* dalam Deteksi Penipuan Transaksi Keuangan Secara Elektronik

Faried Zamachsari[1], Niken Puspitasari[2]
[1]Magister Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Nusa Mandiri Jakarta
[2]RSUP Dr. Kariadi, Kementerian Kesehatan
[1]14002335@nusamandiri.ac.id, [2]niken.p83@gmail.com

Abstract

The rapid development of information technology coupled with an increase in public activity in electronic financial transactions has provided convenience but has been accompanied by the occurrence of fraudulent financial transactions. The purpose of this research is how to find the best model to be implemented in the banking payment system in detecting fraudulent electronic financial transactions so as to prevent losses for customers and banks. Fraud detection uses machine learning with ensemble and deep learning with SMOTE. Financial transaction data is taken from bank payment simulations built with the concept of Multi Agent-Based Simulation (MABS) by banks in Spain. To build the best model, not only pay attention to the accuracy value, but the value of precision is a value that needs attention. A precision score is very important for fraud prevention. Fraud detection gets the best results without the SMOTE process by using deep learning with an accuracy score of 99.602% and precision score of 90.574%. By adding SMOTE, it will increase the accuracy score and precision score with the best model produced in the Extra Trees Classification with an accuracy score of 99.835% and precision score of 99.786%.

Keywords: Fraud Detection, Deep Learning, Deep Neural Network, Machine Learning, Machine Learning with Ensemble.

Abstrak

Pesatnya perkembangan teknologi informasi yang disertai dengan peningkatan aktivitas masyarakat dalam transaksi keuangan secara elektronik disatu sisi memberikan kenyamanan bagi masyarakat akan tetapi disertai dengan terjadinya penipuan transaksi keuangan. Penipuan yang tersembunyi dan beragam jenis transaksi konsumen menyebabkan sulitnya mendeteksi transaksi yang akan merugikan konsumen dan perbankan. Tujuan dari penelitian ini adalah bagaimana mendapatkan model terbaik untuk kemudian disimulasikan ke dalam sistem pembayaran perbankan dengan tujuan melakukan deteksi penipuan transaksi keuangan secara elektronik sehingga dapat mencegah terjadinya kerugian kepada nasabah dan perbankan. Untuk melakukan deteksi penipuan digunakan *machine learning* yaitu *machine learning with ensemble* dan *deep learning*. Kedua jenis *machine learning* tersebut akan dikonfigurasi dan dioptimalisasi menggunakan *SMOTE* sehingga mendapatkan model dengan nilai *accuracy score* dan *precision score* terbaik. Data transaksi keuangan diambil dari simulasi pembayaran bank yang dibangun di atas konsep *Multi Agent-Based Simulation* (MABS) oleh bank di Spanyol dengan tujuan untuk mempromosikan pengembangan aplikasi *Big Data*. Penelitian sebelumnya menggunakan data yang sama dengan judul *A Bayesian Classifier based on Constraints of Ordering of Variables for Fraud Detection* mendapatkan *accuracy score* 99.272%. Untuk membangun model terbaik tidak hanya mempertimbangkan *accuracy score* akan tetapi *precision score* merupakan nilai yang perlu dipertimbangkan. *Precision score* sangat menentukan untuk pencegahan terjadinya penipuan. Deteksi penipuan mendapatkan hasil terbaik tanpa proses *SMOTE* dengan menggunakan *Deep Learning* dengan *accuracy score* yaitu 99.602% dan *precision score* 90.574%. Dengan melakukan penambahan *SMOTE* akan meningkatkan *accuracy score* dan *precision score* dengan model terbaik yang dihasilkan pada *Extra Trees Classification* dengan nilai *accuracy score* 99.835% *dan precision score* 99.786%.

Kata kunci: *Fraud Detection, Deep Learning, Deep Neural Network, Machine Learning, Machine Learning with Ensemble*.

1. Pendahuluan

Penipuan pembayaran adalah penipuan yang disengaja dengan tujuan untuk mendapatkan keuntungan finansial atau menyebabkan kerugian dengan trik implisit maupun eksplisit yang dilakukan oleh banyak pihak untuk mendapatkan keuntungan finansial yang signifikan [1]. Kemajuan dalam komunikasi dan teknologi di seluruh

Diterima Redaksi: 14-02-2021 | Selesai Revisi: 03-04-2021 | Diterbitkan Online: 30-04-2021

dunia secara tidak langsung meningkatkan kasus penipuan [2].

Deteksi penipuan transaksi keuangan secara elektronik merupakan sebuah upaya deteksi dengan menggunakan data transaksi keuangan secara elektronik yang telah diambil untuk kemudian dilakukan analisa menggunakan *machine learning* sehingga karakteristik dari transaksi penipuan yang telah terjadi dapat diidentifikasi untuk kemudian pola yang ada dapat digunakan untuk mendeteksi transaksi keuangan secara elektronik lainnya dalam mencegah terjadinya penipuan yang selanjutnya akan merugikan nasabah dan perbankan. Penggunaan *machine learning* dalam melakukan deteksi penipuan dapat menggunakan *supervised machine learning*.

Supervised machine learning telah digunakan untuk mendeteksi penipuan [3]. Dalam konteks ini, jumlah kasus normal lebih besar dari kasus penipuan sehingga menyebabkan ketidakseimbangan data [2]. Secara umum, teknik *discriminative Supervised* digunakan dalam pekerjaan sebelumnya seperti *Extra Trees Classification*, *Bagged Decision Trees*, *AdaBoost Classification*, *Stochastic Gradient Boosting*, *Random Forest Classification* dan *Neural Network (NN)*.

Extra Trees Classification berbeda dari pohon keputusan klasik dalam cara pembuatannya. Pemisahan terbaik untuk pemisahan sampel *node* menjadi dua grup, pemisahan acak diambil untuk masing-masing fitur *max features* yang dipilih secara acak dan pemisahan terbaik di antara yang dipilih [4].

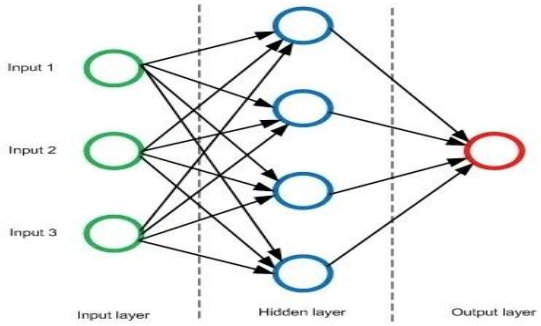
Bagged Decision Trees merupakan pengklasifikasi ansambel, memiliki kinerja tinggi dalam banyak tugas realistis. Itu adalah kombinasinya dari algoritma bagging dan pengklasifikasi pohon keputusan [5].

AdaBoost merupakan pembelajaran yang lemah awalnya disediakan set data pelatihan berbobot sama. Setelah latihan, yang lemah pelajar diuji untuk tingkat kesalahan *pseudo loss*. Kesalahan dihitung sehubungan dengan distribusi atas contoh pelatihan dan label yang salah diperoleh dalam iterasi sebelumnya. Setiap hipotesis lemah atau anggota *ensemble* memiliki *pseudo loss* sedikit lebih baik dari 0,5. Bobot distribusi adalah diperbarui menggunakan aturan pembaruan. Bobot kesalahan klasifikasi contoh memiliki nilai mendekati 0 dan bobot dengan benar contoh yang diklasifikasikan disimpan di dekat 1 untuk iterasi berikutnya [6].

Distribusi *loss function* adalah kunci penerapan model SGB, dan algoritma SGB dapat diterapkan ke semua *loss function*. Untuk masalah *K-class*, *loss function* pengganti (*multi-class log-loss*) adalah *loss function* yang disarankan oleh Friedman dan telah diterapkan secara luas di banyak bidang. *Loss Function* dapat dinyatakan sebagai studi ini, ada tiga parameter kunci untuk algoritma SGB: jumlah pohon (meningkatkan interaksi, M), kedalaman interaksi (kedalaman pohon maksimum, J) dan penyusutan (kecepatan pembelajaran, v). Hyper-parameter ini harus dikombinasikan untuk meningkatkan generalisasi dan kinerja model SGB [7].

Random Forest Classification adalah pengklasifikasi *ensemble* yang terdiri sejumlah besar pohon keputusan, dan hasil klasifikasi akhir ditentukan oleh suara mayoritas dari semua pohon keputusan. Dapat menerapkan dua pendekatan untuk menangani kumpulan data yang tidak seimbang dengan mengatur bobot kelas dan tetapkan ambang probabilitas [8].

Deep learning (DL) memiliki hubungan yang erat dengan *neural networks*, yang merupakan sebuah jaringan syaraf tiruan layaknya saraf pada otak manusia. Pada tahap awal, ia didesain untuk bisa memodelkan bagaimana otak manusia bekerja. DL tidak hanya meniru bagaimana otak manusia bekerja, namun juga berusaha menciptakan jaringan saraf tiruan itu sendiri. Tentunya tidak akan mirip 100% seperti otak manusia dan cara kerjanya sangatlah identik.



Gambar 1. Ilustrasi Lapisan Neuron dalam *Deep Learning*

Melalui ilustrasi pada gambar 1 bisa melihat struktur sederhana dari DL. Struktur ini adalah sistem saraf tiruan yang dimodelkan dari saraf otak manusia. Ada 3 bagian, yaitu (1) *input layer* merupakan neuron yang menerima masukan (input) dari kita. Dalam ilustrasi di atas ada 3 neuron yang menerima input. Satu neuron di dalam arsitektur *neural networks* sering disebut dengan istilah *node*, (2) *hidden layer* merupakan neuron yang menjadi perantara antara input layer dengan output layer. Layaknya mata yang melihat sebuah objek, maka bisa dianalogikan bahwa mata adalah *input layer*, otak adalah *output layer* (tujuan akhir untuk memproses input objek oleh mata), dan *hidden layer* adalah sebuah neuron yang dilewati dari mata sampai ke otak dan (3) *output layer* merupakan nilai akhir (keputusan akhir/ kondisi akhir) yang ingin dihasilkan dari nilai-nilai awal yang dimasukkan di input layer [9].

Fleksibilitas kepada sebuah *neural networks* untuk menentukan apakah sebuah perceptron akan meneruskan informasi ke *layer* selanjutnya atau tidak. Nilainya ditentukan di awal, dan sama seperti *weights*

DOI: https://doi.org/10.29207/resti.v5i2.2952

akan terus diperbaharui melalui mekanisme yang disebut dengan *backpropagation* [10].

Dalam beberapa tahun terakhir, metode data mining telah banyak digunakan dalam deteksi penipuan untuk mengurangi kesalahan yang disebabkan oleh penilaian para ahli, termasuk deteksi penipuan keuangan [11]. Penipuan ini tersembunyi dalam informasi yang sangat besar dan analisa ahli terkadang gagal untuk memperhitungkan keseluruhannya dan penerapan metode *data mining* memecahkan masalah ini. Metode *data mining* yang digunakan termasuk *Support Vector Machine* [12].

Penggunaan metode NN mengungguli pendekatan yang ada pada pengelompokan *Auto Encoder* (AE), *Local Outlier Factor* (LOF), *Isolation Forest* (IF) dan *K-Means*. Metode deteksi penipuan berbasis NN yang diusulkan bekerja dengan akurasi 99,87% sedangkan metode yang ada AE, IF, LOF dan *K-Means* masing-masing memberikan akurasi 97%, 98%, 98% dan 99,75% [13].

Set data *fraud detection* yang tidak seimbang dapat diseimbangkan menggunakan metode *undersampling*, *oversampling* dan *SMOTE*. Hasil pengklasifikasi mesin menunjukkan hasil yang lebih baik dengan *area under curve* (AUC) sebesar 91,37% dihasilkan dengan metode *oversampling* [14].

Penggunaan ANN dengan *Backpropagation*, memberikan akurasi sebesar 99.96% baik pada titik data training maupun pengujian yaitu transaksi nilai riil. Model ini dapat mendeteksi transaksi secara *real time*, sehingga bank dapat mendeteksi penipuan dan menghentikan transaksi yang sedang berlangsung jika transaksi tersebut adalah penipuan [15].

Penggunaan *Random Forest* memperoleh hasil yang baik pada kumpulan data yang kecil, masih terdapat beberapa masalah seperti data yang tidak seimbang. Algoritma *Random Forest* itu sendiri harus ditingkatkan Misalnya, mekanisme pemungutan suara mengasumsikan bahwa masing-masing pengklasifikasi dasar memiliki bobot yang sama, beberapa di antaranya mungkin lebih penting daripada yang lain [16].

Algoritma *Decision Tree*, *Random Forest* dan *XGBoost* yang diterapkan pada kumpulan data memiliki data 284.808 kartu kredit. Performa algoritma *XGBoost* paling baik ditemukan dengan akurasi tertinggi 99.962%. Performa *Decision Tree* tergolong minimal dengan akurasi 99.923% dan performansi algoritma *Random Forest* 99.957% [17].

Penggunaan ANN menghasilkan akurasi lebih baik dibandingkan *Support Vector Machine* dan *Logistic Regression* dalam menangani deteksi penipuan [18].

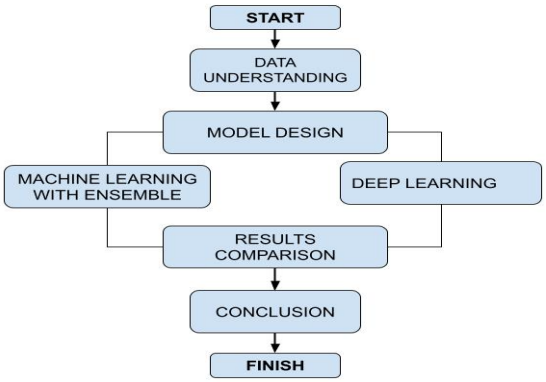
Pada arsitektur ANN pada deteksi kecurangan medis, penggunaan aktivasi SeLU cenderung menghasilkan hasil akurasi klasifikasi terbaik dibandingkan dengan fungsi aktivasi lainnya. Penting untuk dicatat bahwa perbedaan antara ReLU dan ReLU2x adalah pilihan *hyperparameter* [19].

Metode *based-constraint* menghasilkan model *probabilistic graphical* untuk deteksi penipuan. Hasil kompetitif dibandingkan dengan algoritma standar untuk Akurasi yang didapat adalah 99,272%. Hasil ini lebih baik dibandingkan model *probabilistic graphical* seperti *Hill-climbing, K2, genetic search* dan *simulated annealing* [20].

Tujuan dari penelitian ini adalah bagaimana mendapatkan model terbaik untuk kemudian disimulasikan ke dalam sistem pembayaran perbankan dengan tujuan melakukan deteksi penipuan transaksi keuangan secara elektronik sehingga dapat mencegah terjadinya penipuan yang akan menimbulkan kerugian kepada nasabah dan perbankan. Untuk melakukan deteksi penipuan transaksi keuangan secara elektronik digunakan *machine learning* yaitu *machine learning with ensemble* dan *deep learning*. Kedua jenis *machine learning* tersebut akan dikonfigurasi dan dioptimalisasi menggunakan SMOTE sehingga mendapatkan model dengan nilai *accuracy score* dan *precision score* terbaik.

2. Metode Penelitian

Artificial Intelligence (AI) atau yang diartikan sebagai kecerdasan buatan merupakan topik yang sangat hangat. Aplikasinya pun sangat luas, mulai dari skala yang kecil, besar, bahkan hingga tingkat kenegaraan. Secara teknis, untuk bisa membuat sebuah program AI, maka tidak akan lepas dari *machine learning* (ML).



Gambar 2. Metode Penelitian

Pada gambar 2 menjelaskan metode penelitian yang akan digunakan. Proses dimulai dengan memahami data yang diambil pada https://www.kaggle.com, kemudian mendesain model dengan dua pendekatan *machine learning* yaitu *machine learning with ensemble* dan *deep learning*. Hasil eksperimen dari model akan dibandingkan untuk mendapatkan nilai terbaik dalam

DOI: https://doi.org/10.29207/resti.v5i2.2952

melakukan deteksi penipuan pada transaksi keuangan secara elektronik.

2.1 *Data Understanding*

Transaksi keuangan yang mengandung kecurangan banyak dilakukan analisa untuk mendeteksi kecurangan tersebut. Untuk membangun model *machine learning* ini menggunakan data dari https://www.kaggle.com/ntnu-testimon/banksim1/notebooks yang berisi 594,643 transaksi keuangan dengan 7,200 data merupakan data *fraud*.

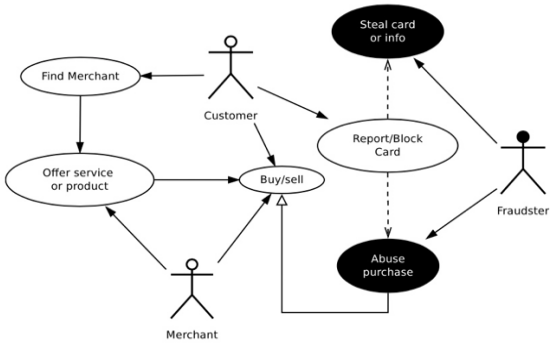
Tabel 1. Deskripsi *Dataset* BankSim

Kolom	Atribut	Jumlah Nilai
1	*Step*	180
2	*Customer*	4113
3	*Age*	8
4	*Gender*	4
5	*Zip Code Origin*	1
6	*Merchant*	50
7	*Zip Merchant*	1
8	*Category*	15
9	*Amount*	*continuous*
10	*Fraud*	2

Tabel 1 menjelaskan bahwa dataset memiliki 10 Variabel yaitu *Step, Customer, Age, Gender, zipcodeOri, Merchant, zipMerchant, Category, Amount* dan *Fraud*. Fitur Penipuan bersifat biner, di mana 0 dianggap sebagai "tidak ada penipuan" dan 1 dianggap sebagai "penipuan".

BankSim adalah Simulasi Pembayaran Bank, yang dibangun di atas konsep *Multi Agent-Based Simulation (MABS)*. BankSim didasarkan pada sampel data transaksi yang dikumpulkan yang disediakan oleh satu bank di Spanyol dengan tujuan untuk mempromosikan pengembangan aplikasi untuk *Big Data*. Data ini memuat catatan data transaksional selama enam bulan, dari November 2012 hingga April 2013 yang dibatasi oleh lokasi kode pos di Madrid dan Barcelona.

Prinsip Banksim adalah konsep transaksi komersial. Banksim mengamati jejaring sosial yang muncul dari hubungan antara pelanggan dan pedagang. Setiap pelanggan memiliki tujuan untuk membeli barang dari *merchant*. Tujuan pedagang adalah untuk melayani pelanggan dan melakukan pembayaran yang dihasilkan ke dalam pembelian barang. Di lingkungan virtual, interaksi antara agen selalu antara pedagang dan pelanggan. Membeli barang dari pelanggan lain atau menjual barang ke pedagang lain tidak termasuk dalam model. Pelanggan dapat mencari pedagang dalam arah radial apa pun dari posisi mereka saat ini di dunia maya dan mencari pedagang yang cocok dengan pilihan kategorinya. Jika tidak ada pedagang yang ditemukan maka transaksi tidak dapat dilakukan, dan langkah untuk pelanggan ini berakhir.



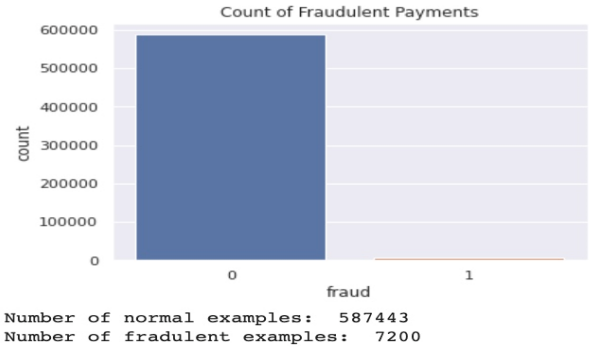
Gambar 3. Diagram *Use Case* Transaksi Pada BankSim

Gambar 3 menjelaskan diagram *use case* BankSim yang didalamnya terdapat juga kasus penyalahgunaan. Proses dapat dijelaskan sebagai berikut: (1) *Find Merchant*. Langkah pertama dalam simulasi untuk pelanggan adalah menemukan pedagang. Pencarian selanjutnya oleh pelanggan dimulai dari pelanggan berpindah dari satu pedagang ke pedagang lainnya. (2) *Offer Service Or Product*. Dilakukan oleh pedagang dan setelah pedagang didekati oleh pelanggan maka pedagang akan menawarkan produk atau layanan sesuai dengan spesifikasi permintaan. (3) *Buy/Sell*. Setelah pelanggan menemukan pedagang dan setelah pedagang menawarkan produk, transaksi terjadi dan menyimpan informasi yang diperlukan untuk pembuatan data sintetis transaksi. (4) *Steal Card or Info*. Penipu bergerak di sekitar lingkungan simulasi dan menemukan pelanggan untuk mencuri kartu fisik atau hanya informasi penting dari kartu kredit pelanggan. Informasi ini disimpan untuk digunakan nanti. (5) *Abuse Purcessing*. Kasus penyalahgunaan ini dilakukan oleh Penipu, mereka melakukan pembelian barang atau jasa di merchant fisik atau internet merchant yang menyembunyikan kehadiran fisik. (5) *Report/Block Card*. Kasus penggunaan ini dilakukan oleh Pelanggan, ketika mereka menyadari bahwa perilaku penyalahgunaan dilakukan di rekening mereka, mereka melaporkan kasus tersebut ke bank dan memblokir kartu untuk penyalahgunaan lebih lanjut. (6) *Log of transactions*. Setiap kali item atau layanan dibeli dari pedagang, transaksi dibuat. Sebuah log berisi informasi tentang pelanggan, pedagang, jumlah, lokasi, tanggal dan penipuan jika ada.

Pada gambar 4 menunjukkan data penipuan yang tidak seimbang dan sebagian besar transaksi adalah non-penipuan. Apabila dataset di sini selalu diprediksi tanpa penipuan, akurasi data awal 98.789 % untuk kumpulan data ini dan sebagian besar untuk yang lain juga karena persentase penipuan sangat rendah. Akurasi sangat tinggi tetapi tidak mendeteksi penipuan apapun sehingga ini adalah pengklasifikasi yang tidak berguna. Jadi skor akurasi dasar harus lebih baik setidaknya daripada selalu

DOI: https://doi.org/10.29207/resti.v5i2.2952

memprediksi non-penipuan untuk melakukan deteksi yaitu 98.789%.



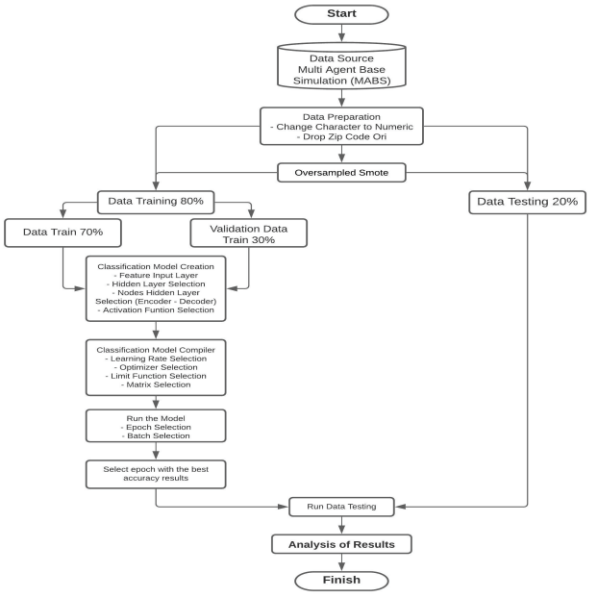
Gambar 4. Statistik Data Transaksi Berdasar Kategori Penipuan dan Non Penipuan

2.2 *Model Design*

Deteksi penipuan transaksi keuangan secara elektronik akan menggunakan 2 pendekatan *machine learning* yaitu *deep learning* dan *machine learning with ensemble* untuk mendapatkan model terbaik.

1) Deep Learning

Deep learning merupakan salah satu bagian dari berbagai macam metode *machine learning* yang menggunakan *artificial neural networks* (ANN). Jenis pembelajaran dalam deep learning dapat berupa *supervised*, *semi-supervised*, dan *unsupervised*. Dalam data yang dibahan adalah data *supervised*. Model yang yang dibangun menggunakan *deep learning* dengan *data supervised* digambarkan dalam gambar 5.



Gambar 5. Bagan Alir Eksperimen *Deep Learning*

Gambar 5 menjelaskan proses eksperimen dalam mencari model terbaik dalam menggunakan *Deep Learning*. Proses diatas dapat dijelaskan:

a. Data *Source*. Dataset yang digunakan dalam eksperimen ini adalah data BankSim sebuah Simulasi Pembayaran Bank, yang dibangun di atas konsep Multi Agent-Based Simulation (MABS).
b. Data *Preparation*. Persiapan data yang pertama adalah dengan mengubah data yang bersifat *character* menjadi *numeric* dan yang kedua dengan menghapus kode asal transaksi dengan tujuan agar model dapat diimplementasikan di negara atau kota lainnya.
c. *Smote*. *Smote* akan membuat poin data baru dari kelas minoritas menggunakan instans yang lain sehingga sampel yang dihasilkan bukanlah salinan persis tetapi serupa dengan instans yang dimiliki. Smote yang akan dipakai adalah *random oversampling* (ROS) dan *random undersampling* (RUS) dan satu algoritma sintesis yaitu FSMOTE.
d. *Split* Data *Training* dan Data *Testing*. Data *training* digunakan untuk melatih algoritma, sedangkan data *testing* dipakai untuk mengetahui performa algoritma yang sudah dilatih sebelumnya ketika menemukan data baru yang belum pernah dilihat sebelumnya. Data *training* sebesar 80% dan data testing sebesar 20%.
e. *Split* data *training*. Data *training* akan dibagi dua, yaitu (a) *Training set. Training set* adalah himpunan data yang digunakan untuk melatih atau membangun model. *Training set* sebesar 70 %. (b) *Validation set. Validation set* adalah himpunan data yang digunakan untuk mengoptimasi saat melatih model. Model dilatih menggunakan *training set* dan kinerja akan dievaluasi dengan *validation set*. Hal ini berguna untuk generalisasi agar model mampu mengenali pola secara umum. *Validation set* sebesar 30%.
f. Pembuatan Model Klasifikasi. Model klasifikasi yang digunakan adalah: (a) *Feature Input Layer*. Input layer yang digunakan adalah 8 *feature*. (b) Jumlah *Hidden Layer*. *Hidden Layer* dipilih dengan berkali-kali percobaan untuk menghasilkan model yang paling baik. Percobaan dimulai dari satu *hidden layer* sampai dengan 500 *hidden layer*. (c) Jumlah *Nodes Hidden Layer* (*Encoder - Decoder*). *Nodes Hidden Layer* dipilih dengan berkali-kali percobaan untuk menghasilkan model yang paling baik. *Representation learning* adalah metode untuk melakukan kompresi *feature vector* menggunakan *neural network*. Proses melakukan kompresi disebut *encoding*, hasil *feature vector* dalam bentuk terkompres disebut *coding*, proses mengembalikan hasil kompresi ke bentuk awal disebut *decoding*. Proses *Encoder Decoder* akan digunakan *autoencoder*. (d) Pemilihan Fungsi Aktivasi. Fungsi aktivasi dari sebuah *node* mendefinisikan keluaran dari simpul tersebut yang diberi masukan atau sekumpulan masukan. Fungsi aktivasi yang digunakan adalah *relu, softplus, softmax, sigmoid, tanh, selu dan elu*.

DOI: https://doi.org/10.29207/resti.v5i2.2952

g. *Compile* Model Klasifikasi. *Compile* model klasifikasi yang digunakan adalah: (a) Pemilihan *Learning Rate*. *Learning Rate* adalah parameter penyetelan dalam algoritma pengoptimalan yang menentukan ukuran langkah pada setiap iterasi sambil bergerak menuju fungsi *loss function*. Karena memengaruhi sejauh mana informasi yang baru diperoleh menggantikan informasi lama, secara metaforis mewakili kecepatan model pembelajaran mesin "belajar". *Learning Rate* yang digunakanan mulai 0.1 sampai dengan 0.0001 untuk mendapatkan performa terbaik dari model yang dibangun. (b) Pemilihan *Optimizer.* Untuk menghasilkan model yang optimal akan digunakan beberapa *optimizer SGD, RMSprop, Adam, Adadelta, Adagrad, Adamax, Nadam* dan *Ftrl.* (c) Pemilihan *Loss Function*. *Loss* adalah ukuran seberapa dekat atau berbeda model yang dihasilkan dengan konsep asli, sementara *error* adalah salah satu fungsi untuk mengukur *loss*. Untuk menghasilkan model optimal beberapa model digunakan adalah *Cross-Entropy, Hinge, Huber, Kullback-Leibler, MAE (L1)* dan *MSE (L2).* (d) Pemilihan *Matrics*. Pemilihan *Matrics* untuk digunakan sebagai Analisa dalam menentukan epoch tertentu agar model tidak *overfitting*.

h. Menjalankan model klasifikasi untuk data *training* Model dijalankan dengan: (a) Pemilihan *Epoch*. *Epoch* adalah berapa kali seluruh *training dataset* diperintahkan ke *network* ketika menjalanakan *training*. *Epoch* dinaikkan sampai *validation accuracy* mulai turun meskipun training *accuracy* masih meningkat (*overfitting*). (b) Pemilihan *BatchSize*. *BatchSize* adalah jumlah *sub-samples* yang diberikan pada *network*, untuk kemudian dilakukan *update parameters*. Semakin besar semakin cepat *learning*-nya, tapi perlu memori lebih besar.

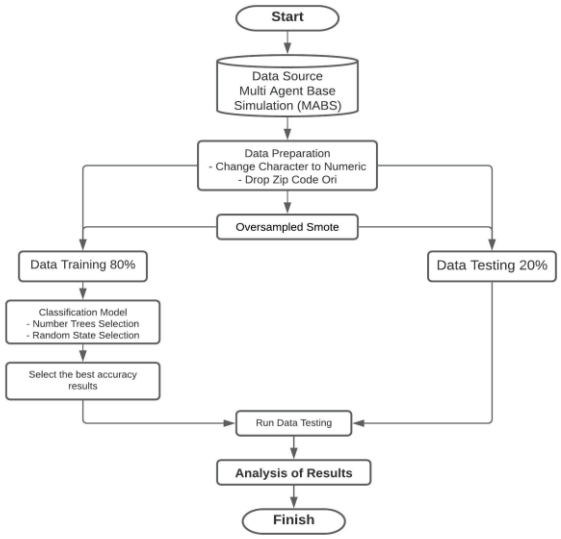
i. Penentuan *Epoch* agar tidak *overfitting.* Dengan menggunakan grafik matriks yang dihasilkan akan ditentukan nilai *epoch* pada titik dimana *validation loss* sudah mulai turun walaupun nilai akurasi yang dihasilkan masih terus naik. Hal ini karena model sudah *overfitting*.

j. Menjalankan model klasifikasi untuk data tes. Berdasarkan model dan nilai *epoch* yang telah ditentukan sehingga menghasilkan nilai yang maksimal, model dijalankan dengan menggunakan data tes yang telah disiapkan sebelumnya.

k. Perolehan Hasil. Analisa hasil sehingga model yang ditetapkan menghasilkan nilai yang paling optimal.

2) Machine Learning With Ensemble

Dalam mencari model terbaik *Machine Learning* yang akan digunakan adalah *Machine Learning* dengan *ensemble*.



Gambar 6. Bagan Alur Eksperimen *Machine Learning*

Dalam *Machine Learning*, metode *ensemble* menggunakan beberapa algoritma pembelajaran untuk mendapatkan kinerja prediksi yang lebih baik daripada yang bisa diperoleh dari algoritma pembelajaran. Tidak seperti *ensemble* statistik dalam mekanika statistic yang biasanya tidak terbatas. *Ensemble Machine Learning* hanya terdiri dari sekumpulan model alternatif terbatas yang konkret, tetapi biasanya memungkinkan struktur yang jauh lebih fleksibel untuk ada di antara alternatif tersebut. Untuk mendapatkan model dengan akurasi terbaik dengan *machine learning* dijelaskan dengan gambar 6. Proses pada gambar 6 dapat dijelaskan sebaga:

a. Data *Source.* Dataset yang digunakan dalam eksperimen ini adalah data BankSim sebuah Simulasi Pembayaran Bank, yang dibangun di atas konsep *Multi Agent-Based Simulation* (MABS).

b. Data *Preparation.* Persiapan data yang pertama adalah dengan mengubah data yang bersifat *character* menjadi *numeric* dan yang kedua dengan menghapus kode asal transaksi dengan tujuan agar model dapat diimplementasikan di negara atau kota lainnya.

c. *Smote. Smote* akan membuat poin data baru dari kelas minoritas menggunakan instans yang lain sehingga sampel yang dihasilkan bukanlah salinan persis tetapi serupa dengan instans yang dimiliki. *Smote* yang akan dipakai adalah *random oversampling (ROS)* dan *random undersampling (RUS)* dan satu algoritma sintesis yaitu *FSMOTE*.

d. *Split* Data *Training* dan Data *Testing.* Data *training* digunakan untuk melatih algoritma, sedangkan data *testing* dipakai untuk mengetahui performa algoritma yang sudah dilatih sebelumnya ketika menemukan data baru yang belum pernah dilihat sebelumnya. Data *training* sebesar 80% dan data *testing* sebesar 20%.

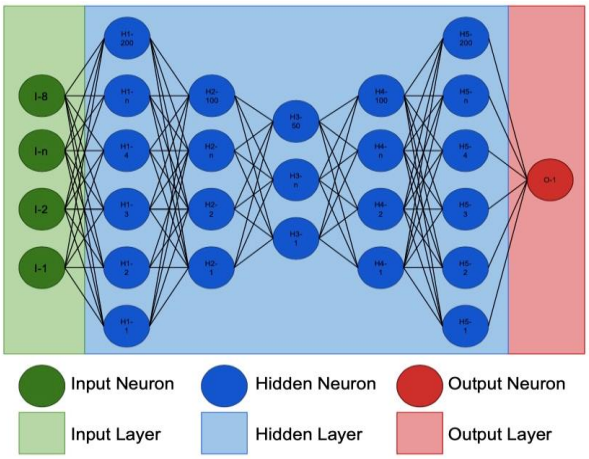
DOI: https://doi.org/10.29207/resti.v5i2.2952

e. *Classification Model.* Algoritma yang akan digunakan untuk mencari model terbaik adalah *Extra Trees Classification, Random Forest Classification, AdaBoost Classification, Stochastic Gradient Boosting dan Bagged Decision Trees.* Model juga akan disimulasikan dengan beberapa jumlah *tree* dan *random state.*
f. Pemilihan akurasi terbaik. Dari percobaan yang dilakukan akan dianalisa dan dipilih model dengan akurasi terbaik yang selanjutkan akan digunakan untuk menjalankan data *testing*.
g. Jalankan model pada data *testing.* Model dengan akurasi terbaik dijalankan pada data *testing*. Hasil yang didapatkan akan dilakukan analisa untuk selanjutnya akan diimplementasikan.

3. Hasil dan Pembahasan

Eksperimen yang telah dilakukan untuk mendapatkan hasil model terbaik dalam setiap algoritma yang digunakan dapat dijelaskan sebagai berikut.

3.1. Pemodelan *Deep Learning*



Gambar 7. Hasil Model *Deep Learning*

Gambar 7 menunjukkan model *deep learning* yang paling optimal untuk melakukan deteksi penipuan transaksi keuangan. *Deep learning* yang memiliki banyak *hidden layer* disebut juga dengan *Deep Neural Network.* Konfigurasi model *deep learning* pada gambar 7 adalah sebagai berikut :

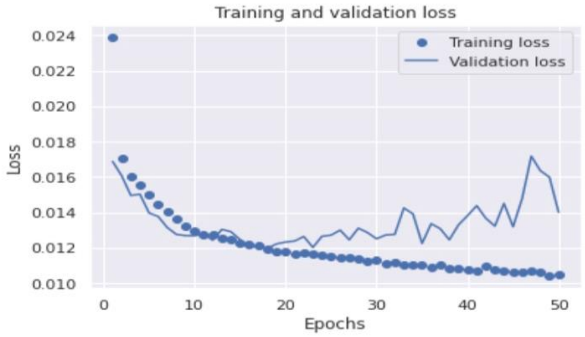
1. Model Klasifikasi. Model Klasifikasi terdiri dari *Feature Input Layer* (8), Jumlah *Hidden Layer* (5), Jumlah *Nodes Hidden Layer (Encoder - Decoder)* (200,100,50,100,200), Jumlah *Output Layer* (1) dan Pemilihan Fungsi Aktivasi (*Hidden Layer Relu, Selu dan Output Layer Sigmoid*)
2. *Compile* Model Klasifikasi. *Compile* model klasifikasi dengan *learning rate* (0.001), *optimizer* (*Nadam*), *loss Function (Binary Crossentropy*) dan *matrics (Accuracy*).
3. Menjalankan model klasifikasi untuk data training dengan *epoch* (18 dan 19) dan *batch size* (50)
4. Sintesis algoritma FSMOTE

3.2. Model *Machine Learning*

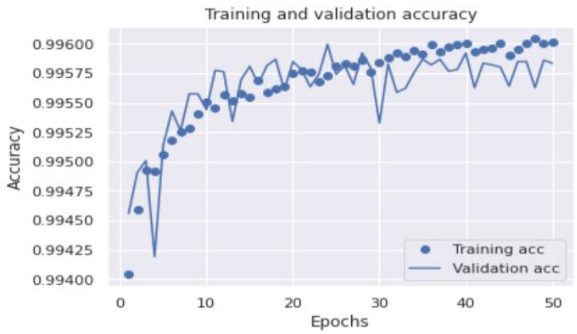
Konfigurasi *machine learning* dengan metode *ensemble* yaitu dengan menggunakan beberapa algoritma pembelajaran untuk mendapatkan kinerja prediksi yang lebih baik. Konfigurasi metode ini akan menggunakan *Num_trees (30), Random_state (7) Max_features (3)* dan *Sintesis* algoritma FSMOTE.

3.3. Hasil Implementasi Model

Model *deep learning* dengan menggunakan *activation function selu* pada *hidden layer* ditunjukkan pada Gambar 8. Pada *epoch* ke 18 nilai *validation loss* sudah sama dengan *training loss*. Pada *epoch* ke 19 nilai *validation loss* sudah naik sehingga walaupun *training loss* masih terus berkurang model sudah menunjukkan bahwa *validation loss* terus meningkat. Grafik ini akan digunakan untuk menentukan titik dimana model sudah mulai *overfitting.*



Gambar 8. Nilai *Training Loss* dan *Validation Loss* dengan *Activation Function Selu*



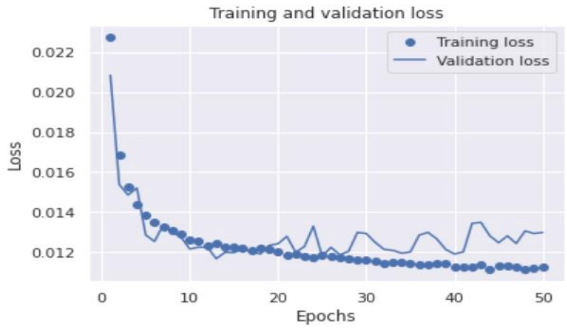
Gambar 9. Nilai *Training Accuracy* dan *Validation Accuracy* dengan *Activation Function Selu*

Gambar 9 menunjukkan pada pada *epoch* ke-18 nilai *validation Accuracy* sudah mulai turun dibawah nilai *training accuracy*. Pada grafik *training loss* dan *validation loss* pada *epoch* ke-19 nilai *validation loss* sudah naik kembali, sehingga dapat disimpulkan bahwa akurasi terbaik dari model adalah *epoch* ke-18 walaupun pada *epoch* ke-19 nilai akurasi masih tetap naik pada *training accuracy* akan tetapi pada titik ini model sudah

DOI: https://doi.org/10.29207/resti.v5i2.2952

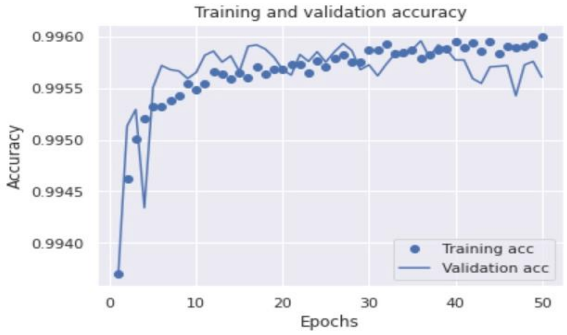
overfitting sehingga jika model diimplementasikan pada data test akan mendapatkan akurasi yang tidak optimal.

Untuk mengoptimalkan hasil dari model, digunakan kombinasi *activation function* pada *hidden layer*. Penggunaan *activation function* pada *hidden layer* ke 1, 2 dan 3 menggunakan *activation function relu* sedangkan penggunaan *activation function* untuk *hidden layer* 4 dan 5 menggunakan *activation function selu.*



Gambar 10. Nilai *Training Loss* dan *Validation Loss* dengan kombinasi *Activation Function Selu* dan *Relu*

Gambar 10 menunjukkan pada pada *epoch* ke 19 nilai *validation loss* sudah sama dengan *training loss*. Pada *epoch* ke 20 nilai *validation loss* sudah naik kembali sehingga walaupun *training loss* masih terus berkurang akan tetapi model sudah menunjukkan bahwa *validation loss* terus meningkat. Grafik ini akan digunakan untuk menentukan titik dimana model sudah mulai *overfitting.*



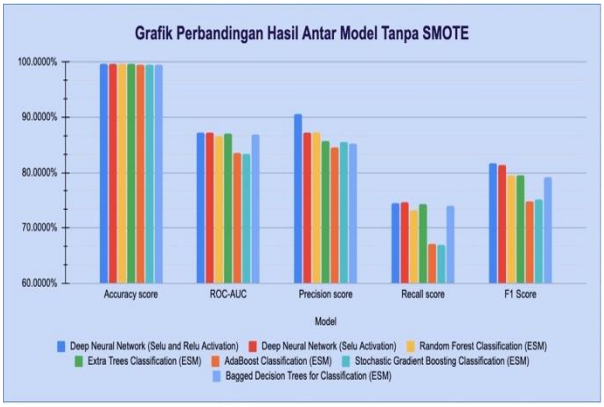
Gambar 11. Nilai *Training Accuracy* dan *Validation Accuracy* dengan *Activation Function Selu* dan *Relu*

Gambar 11 menunjukkan pada pada *epoch* ke-19 nilai *validation Accuracy* sudah mulai turun dibawah nilai *training accuracy.* Pada grafik *training loss* dan *validation loss* pada epoch ke-20 nilai *validation loss* sudah naik kembali, sehingga dapat disimpulkan bahwa akurasi terbaik dari model adalah *epoch* ke-19 walaupun pada *epoch* ke-19 nilai akurasi masih tetap naik pada *training accuracy* akan tetapi pada titik ini model sudah *overfitting* sehingga jika nanti model diimplementasikan pada data tes akan mendapatkan akurasi yang tidak akan optimal.

Pada tabel 2 dan gambar 12 menunjukkan perbandingan antar model. Nilai akurasi *Deep Learning (Deep Neural Network)* yang dihasilkan mendapatkan hasil paling baik yaitu sebesar 99.602% dibandingkan dengan model lainnya. Untuk meningkatkan performa dari sistem dilakukan penggabungan *activation function selu* dan *relu* dalam *hidden layer.* Penggunaan konsep ini dapat meningkatkan nilai *precision score* dan *accuracy score.* Metode yang digunakan pada model yang dibangun dengan satu *activation function* menghasilkan nilai akurasi sebesar 99.589% dan dengan menggunakan dua *activation function* menghasilkan nilai 99.602%.

Tabel 2. Perbandingan Antar Model Bentuk Angka Tanpa *SMOTE*

Model	*Accuracy score*	*ROC-AUC*	*Precision score*	*Recall score*	*F1 Score*
Deep Neural Network (Selu and Relu Activation)	99.602%	87.172%	90.574%	74.437%	81.716%
Deep Neural Network (Selu Activation)	99.589%	87.269%	87.269%	74.648%	81.257%
Random Forest Classification	99.537%	86.511%	87.144%	73.158%	79.540%
Extra Trees Classification	99.530%	87.047%	85.658%	74.249%	79.546%
AdaBoost Classification	99.443%	83.461%	84.510%	67.077%	74.791%
Stochastic Gradient Boosting Classification	99.455%	83.434%	85.602%	67.008%	75.172%
Bagged Decision Trees for Classification	99.522%	86.908%	85.276%	73.975%	79.225%



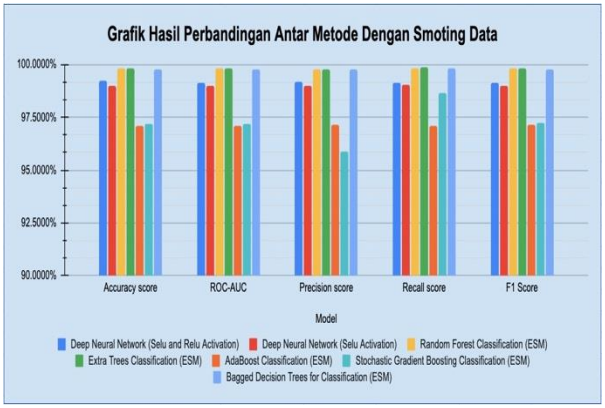
Gambar 12. Perbandingan Antar Model Bentuk Grafik Tanpa *SMOTE*

Data penipuan yang ada adalah data yang tidak seimbang. Untuk menyeimbangkan *dataset* dan mendapatkan nilai yang optimal dapat menggunakan *FSMOTE*. Dengan menggunakan *FSMOTE* mendapatkan nilai sebagai berikut:

DOI: https://doi.org/10.29207/resti.v5i2.2952

Tabel 3. Perbandingan Antara Model Bentuk Angka Dengan *SMOTE*

Model	*Accuracy score*	*ROC-AUC*	*Precision score*	*Recall score*	*F1 Score*
Deep Neural Network (Selu and Relu Activation)	99.217%	99.157%	99.121%	99.127%	
Deep Neural Network (Selu Activation)	99.007%	99.007%	98.993%	99.021%	99.007%
Random Forest Classification	99.802%	99.802%	99.776%	99.830%	99.803%
Extra Trees Classification	**99.835%**	**99.834%**	**99.786%**	**99.885%**	**99.835%**
AdaBoost Classification	97.116%	97.116%	97.133%	97.1155%	97.124%
Stochastic Gradient Boosting Classification	97.196%	97.192%	95.880%	98.650%	97.245%
Bagged Decision Trees for Classification	99.781%	99.781%	99.758%	99.807%	99.782%



Gambar 13. Perbandingan Antara Model Bentuk Grafik Dengan *SMOTE*

Pada tabel 3 dan gambar 13 menunjukkan perbandingan antara model. Nilai akurasi *Extra Trees Classification* yang dihasilkan mendapatkan hasil paling baik yaitu sebesar 99.835% dibandingkan dengan model lainnya. Nilai akurasi ini merupakan nilai yang lebih baik dari jurnal sebelumnya dengan judul *A Bayesian Classifier based on Constraints of Ordering of Variables for Fraud Detection* mendapatkan akurasi 99.272%. Penggunaan *FSMOTE* ini dapat meningkatkan nilai *precision score* dan *accuracy score.* Pada data tanpa *FSMOTE* model DNN memiliki *precision score* dan *accuracy score* sedangkan data dengan *FSMOTE precision score* dan *accuracy score* terbaik menggunakan model *Extra Trees Classification.*

Dalam deteksi penipuan ini tidak hanya memperhatikan nilai akurasi dalam menentukan model paling baik. *Precision score* merupakan faktor yang perlu dipertimbangkan karena nilai ini menunjukkan berapa nilai akurasi model dalam mendeteksi suatu transaksi *fraud* dibandingkan dengan seluruh *fraud* yang nyata. Semakin *precision score* tinggi maka model akan sangat *secure* jika diimplementasikan dalam pencegahan penipuan ini. Sebuah transaksi *fraud* dinyatakan sebagai transaksi normal jauh lebih berbahaya dibandingkan dengan transaksi normal dinyatakan sebagai transaksi *fraud*. Transaksi normal dinyatakan sebagai transaksi *fraud* masih bisa dilakukan konfirmasi sehingga transaksi masih bisa dilanjutkan nantinya akan tetapi suatu transaksi *fraud* tetapi dilakukan nyatakan normal ini akan sangat merugikan dari pengguna dan kerugian keuangan akan diderita oleh nasabah. *Precision score* dalam eksperimen dari semula tanpa *FSMOTE* dengan menggunakan dua *activation function* menghasilkan nilai terbaik 90.574 % untuk model DNN setelah dilakukan *SMOTE* menghasilkan nilai terbaik 99.786 % untuk model *Extra Trees Classification*.

4. Kesimpulan

Nilai akurasi transaksi normal dalam *dataset* adalah 98.789% yang merupakan nilai akurasi dasar untuk membangun model. Data penipuan adalah data yang tidak seimbang, untuk menyeimbangkan data digunakan *SMOTE*. Penggunakan *SMOTE* paling optimal pada data penipuan ini adalah *FSMOTE*. Penggunaan DNN sebagaimana model yang dibangun memiliki akurasi terbaik dengan nilai 99.602% untuk data tanpa *FSMOTE* dibandingkan dengan model *machine learning* dengan *ensemble*. Model ini juga lebih baik dibandingkan model dari penelitian sebelumnya yang menggunakan *Bayesian Classifier based on Constraints of Ordering of Variables* dengan nilai akurasi 99.272%. Penggunaan *FSMOTE* akan meningkatkan nilai akurasi dengan nilai akurasi terbaik menggunakan model *Extra Trees Classification* sebesar 99.835%.

Nilai *Precision Score* merupakan nilai yang perlu dipertimbangkan untuk mebangun model, karena nilai ini yang sangat menentukan untuk pencegahan terjadinya penipuan. Penggunaan *FSMOTE* dapat meningkatkan *Precision Score* dari semula 90.574% pada model DNN dan setelah menggunakan menjadi 99.786% pada model *Extra Trees Classification*.

Penggunaan *FSMOTE* dalam menyeimbangkan data telah merubah model yang paling optimal dari model DNN ke model *Extra Trees Classification*. Model *Extra Trees Classification* dengan *FSMOTE* dapat diimplementasikan dalam sistem pembayaran bank untuk melakukan deteksi penipuan transaksi keuangan secara elektronik sehingga dapat mencegah terjadinya kerugian.

Untuk penelitian selanjutnya dapat disarankan dengan penggunaan *Recurrent Neural Network* (RNN) dengan *Long Short-Term Memory* (LSTM). *Recurrent Neural*

DOI: https://doi.org/10.29207/resti.v5i2.2952

Network (RNN) mampu mengenali dependensi yang panjang dan *Long Short-Term Memory* (LSTM) adalah bagian dari arsitektur RNN yang sangat ampuh untuk memproses data berurut.

Daftar Rujukan

[1] H. M. M. H. Vidanelage, T. Tasnavijitvong, P. Suwimonsatein and P. Meesad, "Study on Machine Learning Techniques with Conventional Tools for Payment Fraud Detection", 2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE), Pattaya, Thailand, pp. 1-5, 2019.

[2] J. Yao, J. Zhang and L. Wang, "A financial statement fraud detection model based on hybrid data mining methods", 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, pp. 57-61, 2018.

[3] D. S. Sisodia, N. K. Reddy and S. Bhandari, "Performance evaluation of class balancing techniques for credit card fraud detection", 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), Chennai, India, pp. 2747-2752, 2017.

[4] L. Abhishek, "Optical Character Recognition using Ensemble of SVM, MLP and Extra Trees Classifier", 2020 International Conference for Emerging Technology (INCET), Belgaum, India, pp. 1-4, 2020.

[5] L. Zhang, J. Xiong, H. Zhao, H. Hong, X. Zhu and C. Li, "Sleep stages classification by CW Doppler radar using bagged trees algorithm", 2017 IEEE Radar Conference (RadarConf), Seattle, WA, USA, pp. 0788-0791, 2017.

[6] L. N. Eeti and K. M. Buddhiraju, "Comparison of AdaBoost.M2 and perspective based model ensemble in multispectral image classification", 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, pp. 3684-3687, 2017.

[7] H. Ding, G. Li, X. Dong and Y. Lin, "Prediction of Pillar Stability for Underground Mines Using the Stochastic Gradient Boosting Technique", in IEEE Access, vol. 6, pp. 69253-69264, 2018.

[8] H. Luo, X. Pan, Q. Wang, S. Ye and Y. Qian, "Logistic Regression and Random Forest for Effective Imbalanced Classification", 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Milwaukee, WI, USA, pp. 916-917, 2019.

[9] Aries Saifudin, Romi Satria Wahono, "Pendekatan Level Data untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software", Journal of Software Engineering, Vol. 1, No. 2, Dec. 2015.

[10] Herlambang, MB. Deep Learning. [Online] Available: https://www.megabagus.id/artificial-intelligence/, Accessed Jan.01, 2021.

[11] Wang, L. and C. Wu, "Business failure prediction based on two-stage selective ensemble with manifold learning algorithm and kernel- based fuzzy self-organizing map", Knowledge-Based Systems. 121: p. 99-110, 2017.

[12] Ching-Chiang Yeh, Der-Jang Chi, Tzu-Yu Lin & Sheng-Hsiung Chiu, "A Hybrid Detecting Fraudulent Financial Statements Model Using Rough Set Theory and Support Vector Machines", Cybernetics and Systems, 47:4, 261-276, 2016.

[13] A. K. Rai and R. K. Dwivedi, "Fraud Detection in Credit Card Data using Unsupervised Machine Learning Based Scheme", 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, pp. 421-426, 2020.

[14] H. Najadat, O. Altiti, A. A. Aqouleh and M. Younes, "Credit Card Fraud Detection Based on Machine and Deep Learning," 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, pp. 204-208, 2020.

[15] S. C. Dubey, K. S. Mundhe and A. A. Kadam, "Credit Card Fraud Detection using Artificial Neural Network and BackPropagation," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, pp. 268-273, 2020.

[16] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang and C. Jiang, "Random forest for credit card fraud detection," 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, China, pp. 1-6, 2018.

[17] V. Jain, M. Agrawal and A. Kumar, "Performance Analysis of Machine Learning Algorithms in Credit Cards Fraud Detection," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, pp. 86-88, 2020.

[18] X. Yu, X. Li, Y. Dong and R. Zheng, "A Deep Neural Network Algorithm for Detecting Credit Card Fraud," 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Fuzhou, China, , pp. 181-183, 2020.

[19] G. Castaneda, P. Morris and T. M. Khoshgoftaar, "Maxout Neural Network for Big Data Medical Fraud Detection," 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService), Newark, CA, USA, pp. 357-362, 2019.

[20] P. Shiguihara-Juárez and N. Murrugarra-Llerena, "A Bayesian Classifier Based on Constraints of Ordering of Variables for Fraud Detection," 2018 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI), Bogota, Colombia, pp. 1-6, 2018.

DOI: https://doi.org/10.29207/resti.v5i2.2952