TELEMATIKA, Vol. 14, No. 01, APRIL, 2017, Pp. 12 – 20
ISSN 1829-667X

PERBAIKAN PERFORMANSI KLASIFIKASI DENGAN PREPROCESSING ITERATIVE PARTITIONING FILTER ALGORITHM

Djoko Budiyanto Setyohadi, Felix Ade Kristiawan, Ernawati
[1]Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta, Yogyakarta, Indonesia
Jalan Babarsari 43, Yogyakarta 55281
e-mail : [1]djoko.bdy@gmail.com

Abstract

Abstract. Preprocessing data and preprocessing performance analysis are crucial in data mining. Those two points have great impact to data mining process success rate, because a quality decisions must be based on quality data. Preprocessing is useful to increase the quality of data and to reduce the noise data. Our experiment show that the performance iterative partitioning filter algorithm is tested by using some dataset from University of California, Irvine (UCI) Machine Learning Repository and is simulated by using modified iterative partitioning filter's parameter variation. This experiment also explained how to analyze classification result from a preprocessed dataset using Backpropagation, so that it can identify best accuracy from multiple datasets that have been tested. Final result from this experiment is table of data consist of training time, classification accurarcy, classification error, Kappa statistic, Mean Absolute Error (MAE) or average of iterations error, Root mean squared error and confusion matrix. This final result is presented in ratio chart between experiment result and modified iterative partitioning filter's parameter.
Keywords: Data mining, Iterative Partitioning Filter, Backpropagation, UCI Machine Learning Repository.

Abstrak

Preprocessing data dan analisis kinerja preprocessing data sangat penting dalam data mining. Kedua hal tersebut memiliki dampak besar pada keberhasilan proses data mining, karena keputusan-keputusan yang berkualitas harus didasarkan pada data yang berkualitas. Dengan preprocessing maka dapat membantu kita untuk meningkatkan kualitas data dan menghapus noise data. Pada penelitian ini, dibahas bagaimana kinerja dari algoritma noisy data filtering yaitu iterative partitioning filter dengan menggunakan berbagai dataset dari University of California, Irvine (UCI) Machine Learning Repository dengan variasi parameter iterative partitioning filter yang dirubah. Penelitian ini juga membahas bagaimana menganalisis hasil klasifikasi dari dataset yang telah di preprocessing menggunakan Backpropagation sehingga dapat mengidentifikasi akurasi terbaik dari berbagai dataset yang telah diuji. Hasil akhir dari penelitian ini berupa data tabel hasil eksperimen yang terdiri dari waktu pelatihan, akurasi klasifikasi, Kesalahan klasifikasi, Kappa statistic, Mean Absolute Error (MAE) atau rata-rata error per iterasi sesuai dengan data, Root mean squared error dan confusion matrix. Hasil akhir yang ditampilkan juga berupa grafik hasil eksperimen yang dibandingkan dengan parameter iterative partitioning filter yang divariasi.
Kata Kunci : Data mining, Iterative Partitioning Filter, Backpropagation, UCI, Machine Learning Repository.

1. PENDAHULUAN

Saat ini data mining merupakan teknik yang dibutuhkan dalam berbagai bidang. Data mining adalah sebuah proses ekstraksi data yang bervolume besar yang bertujuan mendapatkan suatu informasi yang diinginkan (Witten et.al., 2011). Dalam pelaksanaanya, proses yang ada data mining dapat dibagi menjadi tiga (3) tahapan yakni: preprocessing, data mining dan post processing. Tiga tahapan tersebut sangat penting, karena setiap taapan mampu mempengaruhi kualitas luaran data mining. Hal lain yang perlu dipertimbangakan adalah bahwa setiap tahapan mempunyai problematika yang berbeda. Lebih lanjuit dalam paper ini kami memfokuskan pada problema preprocessing.

Preprocessing merupakan proses awal yang akan mentransformasikan data masukan menjadi data dengan format yang sesuai dan siap untuk diproses. Beberapa contoh hal yang dilakukan dalam preprocessing meliputi berbagai proses yang diperlukan antara lain : penggabungan, pengubahan bentuk, ataupun pentransformasian data sebagai cara untuk membersihkan, mengintegrasikan, mereduksi dan mendiskritisasi. Lebih lanjut proses yang ada alam preprosesing dapat terdiri dari salah satu kegiatan proses ataupun gabungan dari beberapa proses diatas. Proses yang ada tergantung dari tujuan yang akan dicapai dalam preprocessing tersebut (Karthick & Malathi, 2015). Pemilihan proses yang tepat perlu dilakukan mengingat karena proses yang sesuai dalam tahapan preprocessing data akan meningkatkan performansi klasifikasi (Raviya & Gajjar, 2013).

Salah satu problem preprocessing yang krusial adalam proses pembershian data. Kondisi ini dapat dipahami karena banyak algoritma data mining bila diberi data yang bersifat korot (*noisy*) mampu mebuat algoritma menjadi bersifat tidak *robust*, kondisi tersebut membuat proses pembersihan data menjadi suatu hal sangat penting (Setyohadi, 2015). Hal itu menjelaskan mengapa algoritma pembersihan data (*Noisy Data Filtering*) berkembang. Salah satu algoritma yang populer adalah algoritma Iterative Partitioning Filter, sebuah metode *Noisy Data Filtering* yang dikembangkan oleh Khoshgoftaar & Rebours(2007). Pada penelitian *Improving Software Quality Prediction by Noise Filtering Techniques*, mereka mengembangkan metode yang memiliki dua buah skema penyaringan data yaitu *majority* dan *consensus*. Filter ini merupakan upaya untuk meningkatkan kualitas input data dengan menghapus potensi *noisy instance*. Instance yang salah, akan diklasifikasikan dan diidentifikasi sebagai noisy yang dihapus dari dataset pelatihan.

Dibandingkan dengan algoritma preprocessing yang lain, penggunaan konsep *majority* dan *consensus dalam Noisy Data Filtering* menjadi lebih menjanjikan karena hal tersebut membuat terkuranginya data *noise* maupun *outlier* secara signifikan *(*Anand, 2012; Karthick, 2015; Kotsiantis 2006). Sebagaimana dipahami dalam pengolahan data modern outlier maupun *noise* merupakan data ikutan yang selalu muncul secara acak. Nilai acak, yang juga merepresentasikan nilai ketidakberaturan, sering diukur sebagai sebuah nilai kompleksitas dataset (Setyohadi, 2015). Mengingat Preprocessing merupakan bagian dari sebuah proses dalam data mining, kami akan mengukur kualitas preprocessing dengan menggunakan sebuah algoritma klasifikasi. Secara singkat kami mengambil prinsip bahwa semakin baik kualitas data yang di preprocessing akan semakin baik performansi algoritma klasifikasi.

Terkait dengan kebutuhan algoritma klasifikasi, pada penelitian ini menggunakan algoritma yang mampu memodelkan masalah yang sangat kompleks. Untuk itulah kami memilih algoritma jaringan syaraf tiruan dengan algoritma Backpropagation. Pilihan ini didasarkan bahwa disamping algoritma tersebut mempunyai kapasitas untuk memodelkan masalah yang sangat kompleks di area Machine Learning, Data Mining dan Pengenalan Pola (Nawi et al., 2013; Maharani, 2009), akan tetapi proses pelatihan algoritma Backpropagation sangat terpengaruh akan kualiatas data pelatihan (Anand et al., 2012).

Memenuhi kebutuhan yang berhubungan problematika kompleksitas data, dataset adalah data yang memiliki karakteristik kompleksitas yang mempengaruhi performansi proses klasifikasi (Setyohadi, 2015). Data ini diambil dari University of California Irvine (UCI), Machine Learning Repository yakni : wine, iris, wisconsin, pima dan haberman. Hasil penelitian disajikan dalam bentuk tabel, ukuran kinerja data preprocessing yang digunakan untuk perbandingan adalah waktu pelatihan, akurasi, *kappa statistic, Mean Absolute Error (MAE), Root mean squared dan confusion matrix*. Hasil penelitian juga disajikan dalam bentuk grafik perbandingan hasil akurasi klasifikasi dengan variasi parameter dan skema penyaringan iterative partitioning filter.

2.TINJAUAN PUSTAKA

Preprocessing data merupakan langkah penting dalam proses penemuan pengetahuan, karena keputusan-keputusan yang berkualitas harus didasarkan pada data yang berkualitas (Kumar & Chadha, 2012). *Preprocessing* data seringkali digunakan untuk mengurangi kesalahan data dan sistematis bias dalam data mentah sebelum analisis apapun terjadi (Tong et al., 2011). Ada banyak faktor yang menimbulkan problem performansi

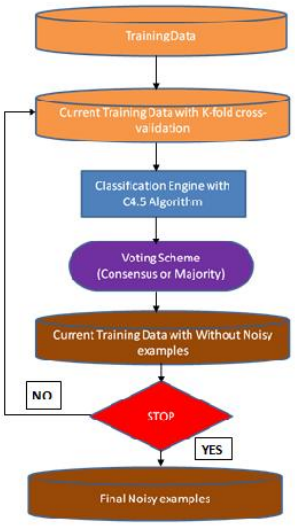
klasifikasi. Yang pertama dan terutama adalah bentuk dan kualitas data, bila data mengandung *noise,* redundansi dan mempunyai data yang tidak relevan, kondisi tersebut membuat proses pengekstrasian *feature* selama fase pelatihan lebih sulit (Kotsiantis et al., 2006). Dengan demikian, *preprocessing data* merupakan langkah yang penting dalam proses Machine Learning. Penggunaan algoritma *Future Subset Selection* akan melakukan pengidentifikasian dan penghapusan fitur yang tidak relevan dan *redundant*\. Implikasi dari kondisi tersebut akan mengurangi dimensi dari data dan memungkinkan algoritma pembelajaran untuk beroperasi lebih cepat dan efektif. Peningkatan efisiensi data dan menghapus *noise* data yang membantu untuk mengidentifikasi *survival of the fittest* juga bisa dilakuak (Karthick & Malathi, 2015). Mereka menyatakan bahwa dengan *preprocessing* akan membantu untuk meningkatkan efisiensi secara signifikan.

Penggalian manfaat preprocessing dalam klasifikasi juga dilakukan secara spesifik dalam algoritma jaringan syaraf tiruan (JST). Nawi et al. (2013) melaporkan manfaat pada *preprocessing* data menggunakan teknik yang berbeda-beda dalam rangka meningkatkan konvergensi JST. Dalam penelitiannya disebutkan bahwa *preprocessing* merupakan langkah penting dalam proses data mining, kualitas, keandalan, dan ketersediaan adalah beberapa faktor yang dapat menyebabkan kesuksesan interpretasi data di JST. Dengan mengolah dataset dari UCI repository yaitu Wine, Iris dan Haberman dengan teknik preprocesing *Min-Max Normalization, Z-Score Normalization dan Decimal Scaling Normalization.* Dalam kesimpulannya disebutkan bahwa penggunaan teknik preprocessing dapat meningkatkan keakuratan klasifikasi JST sedikitnya 95%. Kondisi tersebut berimplikasi pada pengingkatan performnasi algoritma JST.

Penelitian lain juga memperlihatkan bagaimana pentingnya *data preprocessing* agar peningkatkan akurasi prediksi dapat dilakukan khususnya tyerkait dengan problema *missing data*. Preprocessing dilakukan dengan cara mengganti *missing data* pada dataset pelatihan JST. Penggunaan skala dalam rentang nilai data yang sama untuk setiap fitur input terbukti disamping mampu menaikkan kecepatan pemrosesan juga meminimalkan bias dalam jaringan saraf tiruan dalam satu fitur dan fitur yang lainnya. (Anand et al., 2013).

3. ITERATIVE PARTITIONING FILTER

Iterative partitioning filter adalah salah satu algoritma *Noisy Data Filtering* pada *preprocesing data*. *Iterative partitioning filter* menghapus *noisy examples* di beberapa iterasi. Pada dasarnya algoritma ini menggunakan algoritma C4.5. Dalam setiap iterasi data pelatihan dibagi menjadi *n* bagian, dan algoritma C4.5 dibangun disetiap subset ini untuk mengevaluasi semua *examples*. Kemudian semua contoh kesalahan klasifikasi dihapus (menggunakan skema majority atau consensus) dan iterasi baru dimulai (Saez et al., 2016). Pada gambar 2.1 menunjukkan diagram algoritma *iterative partitioning filter*.



Gambar 1. Diagram Algoritma Iterative Partitioning Filter

Pada gambar 1 terlihat bahwa *Iterative Partitioning Filter* menggunakan pohon keputusan C4.5 pada *classification engine*. Sebagaimana dialorkan pada riset2 sebelumnya, algoritma ini bekerja dengan baik sebagai filter untuk *noisy* data dan umumnya menghasilkan hasil yang baik pada berbagai dataset yang besar (Quinlan, 1993). Algoritma C4.5 adalah algoritma yang tergabung dalam kelompok algoritma *Decision Tree*. Algoritma tersebut mempunyai input berupa *training samples* dan *samples* berupa data latihan pohon keputusan yang telah diuji kebenarannya. Selkanjutnya data latihan tersebut akan dipakai sebagai dasar pembentukan sebuah *tree*. *Decision tree* sendiri sebenarnya sama dengan struktur *tree*, dimana tiap internal *node* menunjukkan sebuah *test* pada sebuah atribut, tiap cabang menunjukkan hasil dari *test* dan *leaf node* menunjukkan *class-class* atau *class distribution* (Br Ginting et al., 2014). Secara detail tahapan-tahapan dalam Algoritma Decision Tree C4.5 adalah sebagai berikut :

1. Menyiapkan data training
2. Menentukan akar dari pohon
3. Menghitung nilai Gain :

$$Entropy(S) = \sum_{i=1}^{n} -pi * log_2 pi \quad .....(1)$$

4. Ulangi langkah ke-2 hingga semua tupel terpartisi

$$Gain(S,A) = S - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * S_i \quad .....(2)$$

Proses pembuatan partisi dalam pohon keputusan akan berhenti bila (Swastina, 2013).

1. saat semua tupel dalam node N mendapat kelas yang sama dan atau
2. tidak ada atribut di dalam tupel yang bisa dipartisi lagi dan atau
3. tidak ada tupel didalam cabang yang kosong.

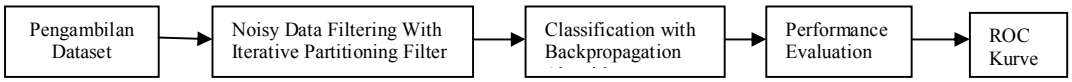
Selanjutnya pada *Iterative Partitioning Filter* terdapat dua buah skema untuk penyaringan noise, bila menggunakan skema majority maka menghilangkan sebuah *instance* bila kesalahan klasifikasi lebih dari 50 % dari pengklasifikasi, sedangkan bila menggunakan skema consensus maka menghilangkan *noisy example* jika kesalahan klasifikasi oleh semua pengklasifikasi. Proses iterasi berakhir setelah kriteria tercapai, sebagaimana yang telah didefinisikan (Zhu et al., August 2003).

4. METODOLOGI PENELITIAN

4.1.Diagram Penelitian

Penelitian ini diawali dengan pengambilan dataset dari University of California, Irvine (UCI) Machine Learning Repository, selanjutnya dataset diolah dengan software Keel menggunakan algoritma *Noisy Data Filtering* dengan metode *Iterative Partitioning Filter,* pada proses *preprocessing* keempat parameter *Iterative Partitioning Filter* diberi nilai pada range tertentu untuk mencari akurasi terbaik. Selanjutnya dataset yang telah diolah pada proses *preprocessing* diklasifikasikan dengan Jaringan syaraf tiruan. SEcara lengkap alur penelitian dapat dilihat pada gambar 2.

.



Gambar 2. Aliran Diagram Penelitian

Mengingat saat ini sudah tersedia perangkat lunak JST yang sudah establish khsusnya perangkat lunak Weka. Dalam penelitian ini kami mempergunakan JST dari WEKA sebagai sarana untuk menguji dan menganalisa kualitas *preprocessing*. Saran yang dimaksud adalah indek-indek yang terkait kualitas klasifikasi hasil klasifikasi JST.

4.2. Dataset Penelitian

Kompleksitas dataset pada penelitian ini terkait dengan masalah perbedaan. Kompleksitas dapat didefinisikan sebagai kesulitan algoritma klasifikasi untuk menentukan batas keputusan. Kinerja dari algoritma klasifikasi dipengaruhi oleh kompleksitas dataset. Struktur kelas juga bisa menjadi karakteristik penting bagi masalah perbedaan. Selain itu dapat merepresentasikan kompleksitas dataset karena sesuai dengan tingkat algoritma klasifikasi (Ho & Basu, 2002). Sehubungan dengan penelitian ini maka dipilih 5 buah dataset yang digunakan dalam penelitian, sesuai dengan tujuan penelitian kompleksitas dataset dipilih dalam berbagai tingkatan yaitu dari rendah ke tinggi (Lihat Tabel 1).

Tabel 1. Karakteristik dataset dan Pengukuran Kompleksitas untuk Pengujian Overlap Clustering (Ho, 2006)

Dataset	#Attributes	#Clusters	#Data	F2	F3
Wine	13	3	178	0.001	0.564
Iris	4	3	150	0.114	0.500
Wisconsin	9	2	683	0.217	0.350
Pima	8	2	768	0.251	0.651
Haberman	3	2	306	0.718	0.029

Data yang digunakan dalam penelitian ini adalah data yang sudah tersedia dalam perangkat keel, data ini diambil dari Machine Learning Repository dataset di University of California Irvine(UCI).Pada tabel 1 diketahui bahwa nilai semakin besar nilai *volume of overlap region* (F2) maka kompleksitas data semakin besar, hal ini berbanding terbalik dengan nilai *overlap feature efficiency*(F3), yaitu semakin kecil nilai *overlap feature efficiency*(F3), maka kompleksitas data semakin besar.

4.3. *Preprocessing*

Pada proses *preprocessing* keempat parameter pada algoritma *iterative partitioning filter* akan disimulasikan sebagai sebuah proses experimental. Luaran yang diukur adalah luaran performansi klasifikasi JST yang mendapatkan masukkan dari luaran algoritma *iterative partitioning filter.* JST yang dipakai sebagai sarana pengukuran implikasi hasil simulasi luaran *iterative partitioning filter* dilakukan dengan menggunakan *software* weka 3.6 dengan metode *Multilayer Perceptron Backpropagation*. Pengaturan parameter pada klasifikasi *Backpropagation* ini dibiarkan dalam nilai *default* yang sudah diberikan dari aplikasi.

4.4. Evaluasi

Proses dilakukan dengan melihat data hasil klasifikasi. Setelah dilakukan klasifikasi dengan *Multilayer Perceptron Backpropagation* maka hasil klasifikasi dievaluasi. Disajikan pula dalam bentuk grafik dan perbandingan akurasi terbaik masing-masing dataset. Selain itu disajikan juga *curva* ROC dari *true positive rate* dan *false positive rate* masing-masing dataset dengan akurasi terbaik.

5. PEMBAHASAN

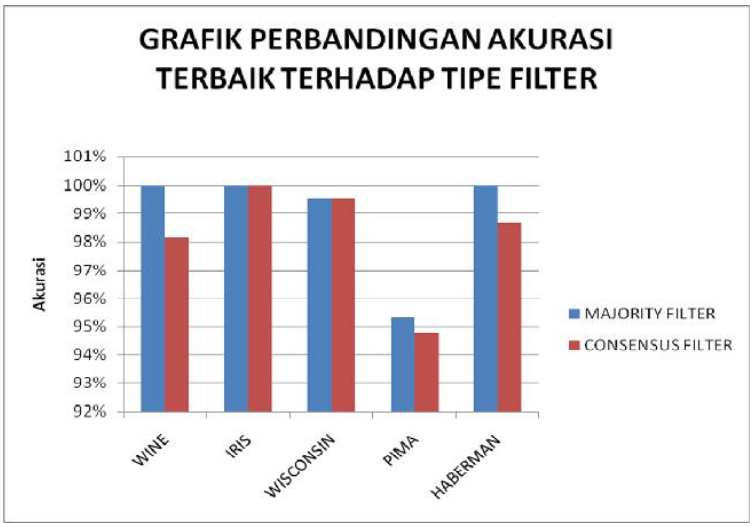
Hasil eksperimen dilakukan dengan simulasi setiap variable di algoritma preprocessing. Luaran performansi diukur berdasarkan index validasi. Hasil yang ada dianalisa berdasarkan pengaruh dataset tingkat noise atau kompleksitas dari dataset. Dengan cara tersebut diharapkan hasil experimen setiap simulasi variable dan keterkjaitannya dengan hasil performansi JST. Kondisi tersebut dipergunakan untuk mengetahui pengaruh algoritma *Iterative Partitioning Filter* dalam memberi masukan (*feeding*) terbaik dan dianggap mampu mempengaruhi algoritma JST sebagai algoritma pengukuran kualitas preprocessing.

Dari hasil yang ada terlihat (lihat gambar 3) bahwa filter dengan skema majority lebih meningkatkan performansi akurasi hasil klasifikasi, dibandingkan skema consensus. Hal ini diperkuat dengan penelitian sebelumnya oleh (Khoshgoftaar & Rebours, 2007) tentang *improving software quality prediction by noise filtering techniques,* yang menyatakan *Iterative Partitioning Filter* dengan skema majority adalah filter yang paling baik. Berikut adalah hasil akurasi terbaik masing-masing dataset setelah preprocessing dengan Iterative Partitioning Filter.

Bila diamati berdasarkan problema data set kualitas problema nosie dalam dataset yang diambil adalah dataset yang memprunyai problem noise type 2. Noise yang diakibatkan kurang kuatnya keterikatan antara nilai atribut dengan klas/kelompok yang ada. Penggunaan sekma majority menjadi suatu jawaban atas hal ini yakni entitas yang atributnya kurang homogen dengan class akan diarakan ke konteks majority yang berada disekeliling entitas tersebut. Fenomena ini terlihat sekali dalam perbandingan akhir, dimana walaupun tinggi tingkat kompleksitas noises tetapi hasil optimum akan tetap dicapai dan tidak terlalu berpengaruh bila dibandingkan dengan dataset yang mempunyai kompleksitas yang rendah. (Lihat tabel 2 dan Tabel 3.). Bila dikaji keterkaitan perbandingan kualitas klasifikasi komponen majority terlihat secara signifikan pengaruhnya (Lihat gambar 2). Detail hasil experimen dapat dilihat pada tabel maupun gambar dibawah ini.

Tabel 2. Hasil akurasi terbaik masing masing dataset berdasarkan tipe filter dengan IPF

DATASET	Filter Scheme	Time Taken	*Correctly Classified*	*Incorrectly Classified*	*Kappa statistic*	MAE	*Root mean squared error*	*Confusion Matrix*
Wine	Majority	**1.65**	100	0	1	0.0253	0.0853	(19 0 0) (0 21 0) (0 0 13)
	Consensus	**1.76**	98.1481	1.8519	0.9717	0.0143	0.071	(19 0 0) (0 21 0) (0 1 13)
IRIS	Majority	**0.42**	100	0	1	0.0133	0.0278	(16 0 0) (0 13 0) (0 0 16)
	Consensus	**0.45**	100	0	1	0.0133	0.0278	(16 0 0) (0 13 0) (0 0 16)
Wisconsin	Majority	**3.79**	99.5238	0.4762	0.9892	0.0089	0.0733	(140 1) (0 69)
	Consensus	**2.67**	99.5215	0.4785	0.9898	0.0071	0.0705	(130 1) (0 78)
Pima	Majority	**2.34**	95.3125	4.6875	0.8914	0.048	0.1891	(127 3) (6 56)
	Consensus	**2.74**	94.7644	5.2356	0.8679	0.0606	0.2084	(134 5) (5 47)
Haberman	Majority	**0.38**	100	0	1	0.0027	0.0028	(0 0) (0 69)
	Consensus	**0.47**	98.6486	1.3514	0.9445	0.0542	0.13	(10 0) (1 63)



Gambar 3. Grafik Perbandingan hasil akurasi terbaik masing-masing dataset terhadap tipe filter

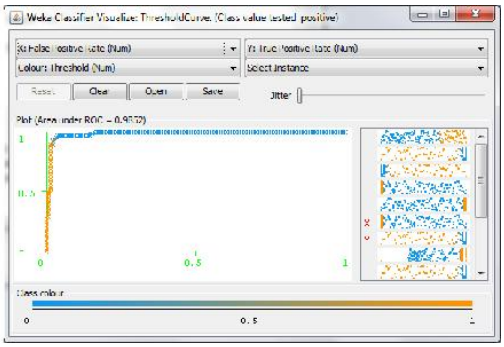
Tabel 3. Hasil perbandingan masing-masing dataset sebelum di *preprocessing* dan setelah di *preprocessing* dengan dua buah skema penyaringan *Iterative Partitioning Filter*.

DATASET	SEBELUM IPF	IPF MAJORITY	IPF CONSENSUS
WINE	98.361%	100%	98.148%
IRIS	96.078%	100%	100%
WISCONSIN	94.118%	99.524%	99.522%
PIMA	79.694%	95.313%	94.764%
HABERMAN	72.115%	100%	98.649%

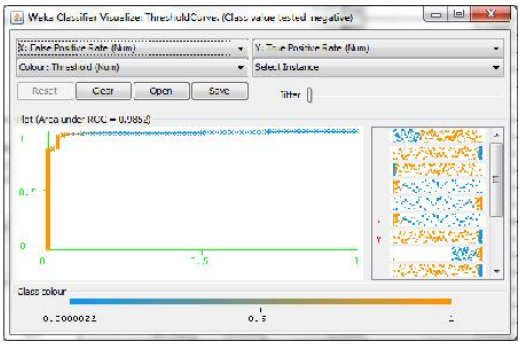
Pada tabel 3 diketahui bahwa sebelum preproccessing akurasi hasil klasifikasi semakin kebawah semakin menurun. KOndisi ini sesuai dengan problema performansi terkait dengan kompleksitas data. Semakin besar kompleksitasnya semakin besar sehingga performansi klasifikasi semakin menurun. Signifikansi perlakuan preprocessing terlihat pada table yang sama. Setelah dilakukan *preprocessing* dengan *Iterative Partitioning Filter* maka performansi klasifikasi meningkat. Peningkatan tersebut terlihat dari naiknya akurasi klasifikasi Hal ini disebabkan karena kualitas data yang meningkat. Bahkan pada dataset haberman, *iterative partitioning filter* dengan skema majority dapat meningkatkan akurasi klasifikasi hingga 27.885 persen.

5.1. Implikasi perlakuan preprocessing pada sensitivitas dan performansi

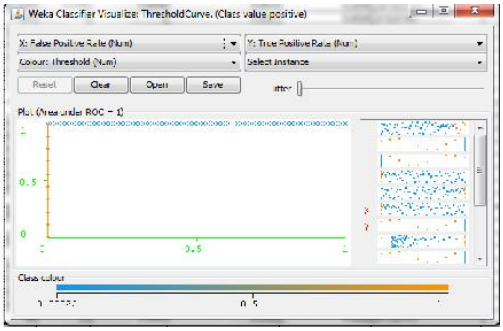
Untuk lebih mempertajam analisa kurva ROC dipakai sebagai sarana melihat sensitivitas maupun kualitas hasil preprocessing. Nilai Luas area dalam ROC merupakan nilai yang dipakai untuk mengetahui akurasi uji diagnostic. Luas area di bawah kurva merupakan representasi rata-rata sensifitas untuk semua nilai spesifitas yang bisa terjadi dalam algoritma klasifikasi.



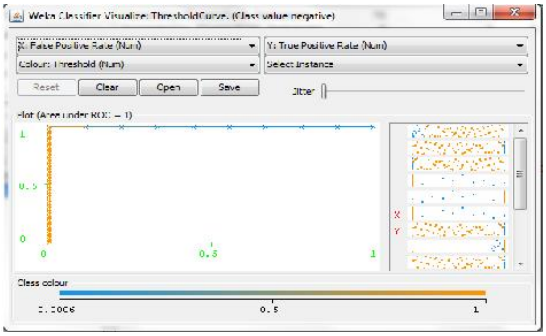
Gambar 4.Kurva ROC dataset pima kelas tested_positive



Gambar 5.Kurva ROC dataset pima kelas tested_negative



Gambar 6.Kurva ROC dataset haberman kelas positif



Gambar 7.Kurva ROC dataset haberman kelas negatif

Gambar 4 s/d Gambar 7 adalah kurva ROC pada masing masing kelas dataset haberman dan pima yang menunjukkan performansi klasifikasi JST yang mendapatkan umpan dari preprocessing *Iterative Partitioning Filte. Dari gambar-gambar tersebut* dapat dilihat bahwa *Area Under Curve (AUC)* dari hasil klasifikasi berada pada nilai 1 dan mendekati 1 yang menunjukkan bahwa hasil klasifikasi JST yang datasetnya mendapatkan perlakuan preprocessing *Iterative Partitioning Filter* performansinya sangat baik.

6. KESIMPULAN

Preprocessing pada dataset wine, iris, wisconsin, pima dan haberman dapat dilakukan dengan metode *iterative partitioning filter.* Dalam melakukan *preprocessing* berbagai data *Iterative Partitioning Filter.* Selanjutnya untuk melakukan validasi pada preprocessing data ini digunakan klasifikasi menggunakan *Backpropagation* menunjukkan bahwa *Iterative Partitioning Filter* performansinya sangat baik. Lebih lanjut hasil eksperimen menunjukkan bahwa skema filter majority lebih efisien meningkatkan akurasi dibandingkan skema konsensus.

Referensi

Anand, R., Kirar, V.P.S. & Burse, K., 2013. K-Fold Cross Validation and Classification Accuracy of PIMA Indian Diabetes Data Set Using Higher Order Neural Network and PCA. *International Journal of Soft Computing and Engineering (IJSCE*, II(6), pp.436-38.

Anand, R., Kirar, V.P.S. & Kavita, B., 2012. Data Pre-processing and Neural Network Algorithms for Diagnosis of Type II Diabetes: A Survey. *International Journal of Engineering and Advanced Technology (IJEAT)*, II(1), pp.49-52.

Br Ginting, S.L., Zarman, W. & Hamidah, I., 2014. ANALISIS DAN PENERAPAN ALGORITMA C4.5 DALAM DATA MINING UNTUK MEMPREDIKSI MASA STUDI MAHASISWA BERDASARKAN DATA NILAI AKADEMIK. *Prosiding Semianar Nasional Aplikasi Sains & Teknologi (SNAST)*, pp.263-72.

Haryati, D.F., Abdillah, G. & Hadiana, A.I., 2016. KLASIFIKASI JENIS BATUBARA MENGGUNAKAN JARINGAN SYARAF TIRUAN DENGAN ALGORITMA BACKPROPAGATION. *Seminar Nasional Teknologi Informasi (SENTIKA)*, pp.557-62.

Ho, T.K. & Basu, M., 2002. Complexity Measures of Supervised Classification Problems. *IEEE Transaction on Pattern Analysis and Machine Intelligence*.

Junaedi, H., Herman, B., Maryati, I. & Melani, Y., 2011. Data Transformatiom Pada Data Mining. *IDeaTech*, pp.93-99.

Karthick, R. & Malathi, D.A., 2015. Preprocessing of Various Data Sets Using Different Classification Algorithms for Evolutionary Programming. *International Journal of Science and Research (IJSR)*, IV(4), pp.2730-33.

Khoshgoftaar, T.M. & Rebours, P., 2007. Improving Software Quality Prediction by Noise Filtering Techniques. *JOURNAL OF COMPUTER AND TECHNOLOGY* , 22(3),

Kotsiantis, S.B., Kanellopoulos, D. & Pintelas, P.E., 2006. Data Preprocessing for Supervised Leaning. *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE*, I(2), pp.111-17.

Maharani, W., 2009. KLASIFIKASI DATA MENGGUNAKAN JST BACKPROPAGATION MOMENTUM DENGAN ADAPTIVE LEARNING RATE. *Seminar Nasional Informatika (semnasIF)* , pp.25-31.

Nawi, N.M., Atomi, W.H. & Rehman, M.Z., 2013. The Effect of Data Pre-Processing on Optimized Training of Artificial Neural Networks. *Procedia Technology*, 8(C), pp.33-40.

Quinlan, J.R., 1993. *C4.5 : Programs for Machine Learning*. San Mateo: Morgan Kaufmann, CA.

Saez, J.A., Galar, M., Luengo, J. & Harrera, F., 2016. An iterative class noise filter based on the fusion of classifiers with noise sensitivity control. *Information Fusion*, 27, pp.19-32.

Setyohadi, D. B., Bakar, A. A., & Othman, Z. A. (2015). Optimization overlap clustering based on the hybrid rough discernibility concept and rough K-Means. Intelligent Data Analysis, 19(4), 795-823. DOI: 10.3233/IDA-150746

Swastina, L., 2013. Penarapan Algoritma C4.5 Untuk Penentuan Jurusan Mahasiswa. *GEMA AKTUALITA*, II(1), pp.93-98.

Witten, I.H., Frank, E. & Hall, M.A., 2011. *Data Mining : Practical Machine Learning Tools and Techniques*. 3rd ed. Elsevier.

Zhu, X., Wu, X. & Chen, Q., August 2003. Eliminating class noise in large datasets. *Proc. the 20th Int. Conf. Machine Learning, Washington DC*, pp.920-27.