XGBOD: Meningkatkan Pendeteksian *Outlier* Secara *Supervised* dengan Representasi *Learning Unsupervised*

Gde Agung Brahmana Suryanegara[#1], Muh. Alkahfi Khuzaimy Abdullah[#2], Muhammad Miftah Falah[#3]

[#]*Fakultas Informatika, Universitas Telkom*
Jl. Telekomunikasi No. 1 Bandung, Jawa Barat, Indonesia
[1]brahmasurya@student.telkomuniversity.ac.id
[2]alkahfi@student.telkomuniversity.ac.id
[3]wizzmf@student.telkomuniversity.ac.id

***Abstrak* — Algoritma *ensemble semi-supervised* terbaru bernama XGBOD (*Extreme Gradient Boosting Outlier Detection*) diusulkan, dideskripsikan, dan didemonstrasikan untuk meningkatkan pendeteksian *outlier* melalui pengamatan secara normal terhadap beberapa dataset. *Framework* yang diusulkan merupakan gabungan dari dua kekuatan metode, yaitu *supervised* dan *unsupervised* pada *machine learning* yang menciptakan sebuah pendekatan *hybrid* dengan memanfaatkan kemampuan kedua metode tersebut dalam mendeteksi *outlier*. XGBOD ini menggunakan beberapa algoritma *unsupervised* dalam menambang *outlier* untuk mengambil representasi yang berguna dari data yang mendasarinya, sehingga dapat meningkatkan kapabilitas prediksi sebuah pengklasifikasi *embedded supervised* pada sebuah *feature space* yang meningkat. Pendekatan yang unik diimplementasikan dan ditampilkan untuk menyajikan performansi unggul ketika ditandingkan dengan pendeteksi-pendeteksi yang individu (selain *ensemble*). Penelitian ini akan membandingkan *Full ensemble* dan dua representasi algoritma berbasis learning yang telah ada, dengan diujikan di atas tiga dataset yang memiliki data *outlier* di dalamnya.**

***Kata Kunci* — *Semi-supervised*, *machine learning*, metode *ensemble*, pendeteksian *outlier*, *unsupervised, supervised*.**

I. PENDAHULUAN

Metode pendeteksian *outlier* secara luas digunakan untuk mengidentifikasi pengamatan anomali pada data [1]. Namun, menggunakan pendeteksi *outlier* dengan algoritma *supervised* tidak dapat dianggap sepele, karena *outlier* pada data khususnya, hanya merupakan sebagian kecil dibandingkan dataset yang melingkupinya. Lagi pula, tidak seperti metode pengklasifikasian yang tradisional, *ground truth* sering tidak tersedia dalam pendeteksian *outlier* [2] – [4]. Bagi algoritma *supervised*, data yang sangat tidak seimbang, dan data nya kurang dilabeli dapat membatasi kemampuan generalisasi metode nya [2]. Selama bertahun-tahun, banyak sekali algoritma *unsupervised* untuk pendeteksian *outlier*, yang mana dikhususkan dalam menelusuri *outlier* yang memiliki keterkaitan informasi seperti kepadatan local, korelasi global, dan hubungan hierarki untuk data tanpa label.

Metode *ensemble* menggabungkan beberapa pengklasifikasi dasar untuk membuat algoritma-algoritma yang lebih kuat dibandingkan menggunakan pengklasifikasi dasar secara individu. Dalam beberapa decade terakhir, banyak *ensemble framework* yang diusulkan, seperti *bagging* [6[, *boosting* [7], dan *stacking* [8]. Meskipun metode-metode *ensemble* tersebut sudah ditelusuri pengakplikasian *unsupervised* dan *supervised* nya, teknik *ensemble* untuk *outlier* sudah jarang diteliti [3]. Algoritma pendeteksi *outlier* biasanya *unsupervised* dan pelabelan yang benar nya masih kurang, pembuatan mereka terbilang penting [2], [4]. Kebanyakan metode *ensemble* untuk *outlier* termasuk kedalam *unsupervised*, menggunakan antara pendekatan *bagging* seperti *feature bagging* [9] atau pendekatan *boosting* seperti SELECT [3]. Namun, kapabilitas prediksi daripada metode *supervised* sering terlalu bergantung pada data yang berlabel di dalam dataset. Oleh karena itu, *ensemble* untuk *outlier* yang berbasis *stacking* digunakan untuk mempengaruhi keduanya, label yang berkaitan dengan informasi menggunakan *supervised learning* dan representasi data yang kompleks menggunakan metode *outlier unsupervised*.

Penelitian yang disajikan di sini meluas dan meningkatkan penelitian yang telah dilakukan oleh Micenkova dkk. [10], [11] dan Aggarwal dkk. [12] untuk mengusulkan sebuah *framework ensemble semi-supervised* untuk pendeteksian *outlier*. *Feature space* asli ditambahkan dengan menerapkan beberapa fungsi pendeteksi *outlier unsupervised*. Fungsi pendeteksi *outlier unsupervised* dapat menghasilkan *Transformed Outlier Scores* (TOS) yang mana merepresentasikan data dengan begitu kaya nya. Penyeleksi algoritma Greedy TOS kemudian diterapkan untuk memangkas penambahan *feature space* dengan tujuan untuk mengendalikan kompleksitas komputasi dan meningkatkan akurasi prediksi. Akhirnya, metode *ensemble supervised* bernama XGBoost [13] digunakan sebagai *classier* hasil akhir pada *feature space* yang disempurnakan. Kombinasi fitur asli ini dengan keluaran dari beberapa algoritma *unsupervised* dasar (untuk mendeteksi *outlier*) memungkinkan representasi data yang lebih baik, mirip dengan *classification meta-framework stacking* [8].

Motivasi di balik penelitian ini ialah, algoritma *unsupervised* dalam mendeteksi *outlier* lebih baik dalam mempelajari pola yang kompleks pada dataset yang sangat tidak seimbang dibandingkan dengan metode *supervised*. Strategi pengambilan *output* dari metode *unsupervised* sebagai *input* untuk pengklasifikasi *supervised* dianggap sebagai proses daripada *representation learning* [10], [11] atau *unsupervised feature engineering* [12]. *Stacking* digunakan sebagai sebuah kombinasi *framework* untuk mempelajari *weight* dari fitur asli dan menghasilkan TOS terbaru secara otomatis. Dibandingkan dengan penelitian yang telah ada [10], [11], pendekatan yang direpresentasikan pada penelitian ini tidak bergantung pada metode EasyEnsemble yang mahal [14] untuk menangani ketidakseimbangan data dengan membangun beberapa sampel penyeimbang, sebaliknya, penelitian ini menggunakan XGBoost [13] sebagai gantinya. Lagi pula, beberapa metode untuk memilih TOS dirancang, dinilai, dan dibandingkan untuk mencapai suatu anggaran komputasi yang efisien. Bahkan, XGBOD tidak memerlukan pemrosesan data pada kombinasi fitur, seperti *feature*

scaling pada *logistic regression* di [10], [11], sehingga *setup* XGBOD ini lebih mudah. Terakhir, penjelasan teoritis mengenai XGBOD disediakan di usulan *framework* terbaru yang diusulkan oleh Aggrawal dan Sathe [15]. Secara keseluruhan, XGBOD mudah digunakan, efisien dalam pengimplementasiannya, dan terbukti efektif secara empiris dalam mendeteksi *outlier*.

II. STUDI TERKAIT

A. Representation Learning

Keefektifan algoritma *machine learning* sangat bergantung pada representasi data yang dipilih atau *feature* [16], yang berlimpah, dan representasi yang efektif cenderung menghasilkan hasil prediksi yang bagus. Beberapa algoritma *machine learning*, seperti *deep learning*, punya kapabilitas dalam mempelajari baik pemetaan representasi untuk *output* serta untuk algoritmanya sendiri. Namun, algoritma-algoritama itu memerlukan data yang berjumlah besar untuk mengambil representasi yang berguna, yang terkadang tidak tersedia ketika penambangan *outlier*. Namun, konsep nya mudah digunakan untuk pendeteksian *outlier*: metode pendeteksi *outlier unsupervised* dapat dipandang sebagai alat untuk mengambil representasi yang kaya dari data yang terbatas, yang biasa dikenal dengan *unsupervised feature engineering* [12]. Pendekatan ini telah terbukti efektif dalam memperkaya data dan meningkatkan *supervised learning* [17].

B. Ketidakseimbangan data dan Extreme Gradient Boosting

Ketidakseimbangan data terjadi ketika *class* yang diberikan (atau subset) di dalam dataset merepresentasikan hanya sebagian kecil dari keseluruhan populasi *class* [18]. Ketika ketidakseimbangan data ditemukan, performa dari pengklasifikasi akan terdegradasi biasanya. Pendeteksian *outlier* merupakan sebuah klasifikasi *binary* yang relatif tidak seimbang [15]. *Outlier* pada dasarnya berjumlah minoritas, sehingga pendeteksiannya menjadi sulit. Untuk menangani ketidakseimbangan data, *bootstrap aggregating* (*bagging)* atau *EasyEnsemble* akan dilibatkan dalam pendeteksian *outlier* [10], [11]. *EasyEnsemble* membuat beberapa subsampel yang seimbang dengan mengambil sampel secara menurun daripada *class* yang mayoritas, dan menggabungkan dasar *output* dari pengklasifikasi yang telah dilatih pada subsampel, seperti *majority vote*. Namun, metode-metode ini mahal untuk dieksekusi dan performa nya hanya bagus pada kasus masalah yang spesifik [19].

Extreme Gradient Boosting, biasanya disebut sebagai XGBoost, merupakan sebuah metode ensemble berbasis *Tree* dan dikembangkan oleh Chen [13]. XGBoost dapat diskalakan, dan implementasi yang akurat terhadap *gradient boosted tree*, secara eksplisit didesain untuk mengoptimalkan kecepatan komputasi dan performa model. Dibandingkan dengan algoritma *boosting* yang diakui seperti *gradient boosting*, XGBoost memanfaatkan regularisasi untuk mengurangi efek *overfitting*, menghasilkan prediksi yang lebih baik [13], dan waktu eksekusi yang lebih cepat [20]. Penelitian terbaru menunjukan bahwa metode *ensemble* dengan XGBoost memiliki kemampuan terbesar untuk menangani dataset tidak seimbang dibandingkan metode *ensemble* yang lainnya [18]. Hasilnya, XGBoost dipilih sebagai pengklasifikasian final *supervised* yang menggantikan *EasyEnsemble* pada studi ini. Selain itu, XGBoost dapat secara otomatis menghasilkan ranking daripada fitur penting ketika *fitting* data [20], hal ini berguna untuk mengimplemnetasikan skema pemangkasan sebuah fitur untuk meningkatkan efisiensi komputasi daripada algoritma yang disajikan disini.

C. Metode Deteksi Unsupervised Outlier

Metode *Unsupervised* merupakan metode yang tidak bergantung pada informasi label dan dapat mempelajari karakteristik pencicilan melalui berbagai pedekatan. Mengembangkan deteksi outlier menggunakan *Unsupervised* dikategorikan kedalam emap kelompok [21]. (I) Liniear Model yaitu analisis komponen utama, (II) *Proximity-Based Outlier Model* yaitu metode berbasis kepadatan, (III) *Statistical and Probabilitc models* yaitu mengandalkan analisis nilai, dan (IV) *High-dimensional outlier models* sebagai hutan isolasi. Model-model ini memiliki fungsi yang berbeda satu sama lainya, dengan asumsi yang berbeda, dan ketika asumsi terpenuhi dapat menghasilkan hasil superior pada dataset tertentu. Pada penelitian ini menggunakan berbagai macam metode deteksi outlier *unsupervised* sebagai deteksi dasar untuk membangun *ensemble* yang efektif.

D. Outlier Ensemble

Metode *ensemble* merupakan metode yang sudah banyak diperkenalkan dan dibahas sebelumnya dalam konteks deteksi outlier [2], [4], [12]. Studi-studi ini telah menggabungkan outlier detector konsituen yang berbeda dengan potensi kesalahan independen [3]. Strategi kombinasi yang paling mudah yaitu rata-rata output dari berbagai detector dasar setelah normalisasi yang dikenal sebagai *Full ensemble.* Salah satu karya paling awal, Feature Bagging [9], menginduksi keragaman dengan membangun subset fitur yang dipilih secara acak. Rayana dan Akoglu mengadaptasi pendekatan peningkatan menjadi penambangan outlier [3]. Algoritma mereka, SELECT, menghasilkan informasi label semu untuk melakukan pembelajaran sekuensial [3]. Patut diketahui bahwa kerangka kerja ini *unsupervised* dan mirip dengan metode *bagging and boosting* dalam tugas klasifikasi tradisional. Pada penelitian ini, kami menggabungkan hasil dari berbagai detektor *unsupervised* melalui *supervised* yang serupa pendekatan yang disebut Stacking [8]. Penumpukan telah digunakan baru-baru ini dalam menyisir *ensemble* yang diawasi dan tidak terawasi dalam tugas populasi basis pengetahuan [22].

E. Ensemble Outlier Semi Supervised dengan Feature Learning

Penelitian sebelumnya Micenkova dkk. telah mengusulkan semi-*supervised* sebagai kerangka kerja yang disebut dengan BORE untuk memanfaatkan kekuatan kedua metode *supervised* dan *unsupervised* [10], [11]. BORE pertama kali digunakan untuk berbagai metode deteksi outlier dengan unsupervised untuk menghasilkan skor outlier pada *data training.* Metode *unsupervised* kemudian digabungkan dengan fitur asli untuk membangun ruang fitur baru. Untuk mengatasi ketidak seimbangan dari data pada data pencicilan peneliti menggunakan EasyEnsemble [14] untuk membuat banyak sampel pelatihan yang seimbang dan mereka kemudian melakukan rata-rata hasil dari sampel. *Logistic regression* dengan regulasi L2 diterapkan pada sub sample untuk mengidentifikasi outlier. L1 regression disarankan untuk mencegah overfitting dan melakukan pemilihan fitur sementara banyak detector disajikan.

III. DESAIN ALGORITMA

XGBOD adalah kerangka kerja tiga fase, seperti yang digambarkan pada Gambar. 1.

Pada fase pertama, representasi data baru dihasilkan. Pada fase kedua, seleksi proses dilakukan pada skor outlier yang baru dibuat untuk disimpan yang bermanfaat. Skor outlier yang dipilih kemudian digabungkan dengan fitur asli untuk menjadi ruang fitur baru. Akhirnya, classifier XGBoost dilatih pada fitur baru ruang, dan hasilnya dianggap sebagai hasil prediksi.

A. *Fase I: Pembelajaran Representasi Tanpa Pengawasan*

Pendekatan yang diusulkan didasarkan pada gagasan bahwa skor outlier tanpa pengawasan dapat dilihat sebagai bentuk pembelajaran representasi dari data asli [10] - [12]. Kalau tidak, ini juga dapat dipahami sebagai bentuk fitur yang tidak diawasi teknik, untuk menambah ruang fitur asli juga.

Biarkan ruang fitur asli dan $X \in R^{nxd}$ menunjukkan seperangkat *n* titik data dengan fitur *d*. Deteksi outlier adalah biner klasifikasi, vektor $y \in \{0,1\}$ memberikan label outlier, di mana 1 mewakili outlier dan 0 mewakili poin normal. Biarkan L menjadi set pengamatan berlabel X, sedemikian rupa sehingga:

$$L = \{(x_1, y_1), \dots, (x_n, y_n)\} \in R^{nxd} \quad (1)$$

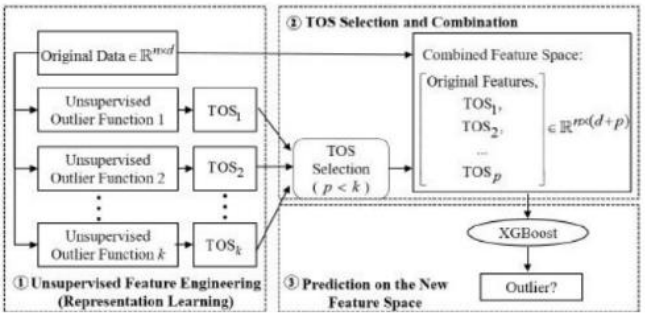
Fungsi penilaian pencilan didefinisikan sebagai pemetaan function $\Phi(\cdot)$, di mana setiap fungsi scoring akan menghasilkan vektor bernilai riil $\phi_i(x) \in R^{n\times 1}$ pada dataset X **Transformed Outlier Score (TOS)** untuk menggambarkan tingkat keterasingan. Fungsi penilaian outlier bisa berupa apa saja metode deteksi outlier tanpa pengawasan. Keluaran, TOS, adalah digunakan sebagai fitur baru untuk menambah ruang fitur asli. Menggabungkan fungsi *k* penilaian pencilan bersama-sama membangun matriks fungsi transformasi: $\Phi = [\Phi_1, \dots, \Phi_k]$ yang menghasilkan matriks skor outlier dari fungsi penilaian dasar *k* pada ruang fitur asli *X*. Menerapkan $\Phi(\cdot)$ pada dokumen asli data *X*, matriks skor pencilan $\Phi(X)$ kemudian diberikan sebagai:

$$\Phi(x) = [\Phi_1(x)^T, \dots, \phi_k(x)^T] \in \mathrm{R}^{n\times k} \quad (2)$$

Seperti disebutkan di atas, setiap deteksi outlier tanpa pengawasan metode dapat digunakan sebagai fungsi skor outlier dasar untuk transformasi fitur. Namun, basisnya heterogen fungsi cenderung menghasilkan hasil yang lebih baik, karena keluaran identik dari fungsi dasar tidak banyak berkontribusi pada ansambel [1], [12] Keragaman antara fungsi-fungsi dasar mendorong perbedaan karakteristik data yang harus dipelajari, mengarah pada peningkatan kemampuan generalisasi ansambel. Lebih jauh lagi, sangat estimator basis berkorelasi menghasilkan kesalahan yang sama dan tidak berkontribusi pada prediksi; sebaliknya, mereka membawa yang tidak perlu beban komputasi yang tinggi untuk solusi keseluruhan.

Sementara itu, fungsi penilaian pencilan harus akurat juga, karena yang tidak akurat menurunkan prediksi. Hasil dari, ada tradeoff yang melekat antara keragaman dan akurasi:

GAMBAR 1. ILUSTRASI XGBOD



menggunakan detektor yang berbeda tetapi tidak akurat meningkatkan keragaman di risiko penurunan kemampuan prediksi keseluruhan. Karena itu, keseimbangan antara keragaman dan akurasi harus dipertahankan untuk mendapatkan hasil prediksi yang ditingkatkan [1], [4]. Di studi ini, berbagai jenis metode outlier tanpa pengawasan adalah digunakan sebagai fungsi penilaian outlier dasar, dan parameternya juga di-tweak untuk menghasilkan variasi lebih lanjut. Desain ini menghasilkan beragam koleksi TOS yang akurat dan tidak akurat.

B. *Fase II : TOS Selection*

Setelah matriks skor outlier $\Phi(X)$ dihasilkan, itu siap dipadukan dengan fitur asli X. Sedang bekerja dari Micenková et al. [11], X secara langsung dikombinasikan dengan seluruh representasi data yang baru dibuat $\Phi(X)$ sebagai:

$$Feature\ Space_{new} = [x, \Phi(x)] \in R^{n\ x\ l} \quad (3)$$

di mana $l = (d + k)$ adalah dimensi dari fitur gabungan ruang dengan *d* fitur asli dan *k* TOS yang baru dibuat.

Dibandingkan dengan pendekatan yang disajikan di sini, beberapa TOS metode seleksi dirancang untuk memilih hanya $p = (p \leq k)$ TOS dari $\Phi(X)$ untuk menggabungkan dengan fitur asli. Itu Alasannya terkait erat dengan pemilihan fitur dalam mesin belajar: tidak semua KL akan berkontribusi pada prediksi. Selain itu, mengurangi jumlah TOS yang dipilih akan mempercepat eksekusi karena (i) ada lebih se dikit transformasi ke berlaku pada data asli dan (ii) ruang fitur gabungan lebih kecil untuk dipelajari. Tiga metode pemilihan didefinisikan, dan set S kosong diinisialisasi untuk menyimpan TOS yang dipilih.

Random Section mengambil p TOS dari $\Phi(X)$ secara acak dan menambah *S* tanpa penggantian.

Accurate Selection memilih p top TOS paling akurat. Ukuran akurasi dapat berupa evaluasi yang sesuai metrik, seperti area di bawah karakteristik operasi penerima curve (ROC), antara lain. Biarkan $Acc_i(\cdot)$ menunjukkan ROC dari $\Phi_i(x)^T$ diukur dengan kebenaran dasar y. Kemudian iterative memilih TOS paling akurat dalam $\Phi(X)$ berdasarkan nilainya dari $Acc_i(\cdot)$ menggunakan:

$$Acc_i = ROC(\Phi_i(x)^T, y) \quad (4)$$

Balance Selection menjaga keseimbangan antara keragaman dan akurasi dengan memilih TOS yang akurat dan berbeda. Untuk setiap $\Phi i(X) \in \Phi(X)$, pilihan serakah adalah dieksekusi berdasarkan akurasi TOS yang dihitung oleh Persamaan. (4) Untuk meningkatkan keragaman dalam S pada saat yang sama, akurasi diskon fungsi $\Psi(\Phi_i)$ dievolasikan berdasarkan Persamaan. (4) sebagai:

$$\Psi(\Phi i) = \frac{Acc_i}{\sum_{j=1}^{\#(s)} |p(\Phi_i, \Phi_j)|}$$
$$subject\ to\ \Phi_j \in \{S\}, Acc_i \geq 0 \quad (5)$$

Korelasi Pearson $p(\Phi_i, \Phi_j)$ digunakan untuk mengukur korelasi antara sepasang TOS. Korelasi Pearson antara TOS dan semua TOS yang dipilih dalam *S* digabungkan sebagai $\sum_{j=1}^{\#(s)} |p(\Phi_i, \Phi_j)|$. Jika TOS sangat berkorelasi dengan TOS
yang telah dipilih dalam S, penyebut yang lebih besar akan ditugaskan dalam Persamaan. (5) untuk diskon akurasinya. Itu fungsi akurasi diskon lebih memilih TOS akurat yang dimiliki korelasi rendah dengan TOS yang sudah dipilih di *S*, dengan demikian mencegah kemiripan with-in set di *S*. Sampai ukuran set *S* sama dengan p, TOS dalam $\Phi(X)$ akan dievaluasi secara iteratif oleh persamaan (5) Setiap kali, dengan diskon terbesar akurasi $(\Psi(X)_i)$ ditambahkan ke *S* dan dihapus dari kandidat pool $\Phi(X)$. Alur kerja diberikan dalam Algoritma 1.

Dengan menjalankan salah satu algoritma pemilihan TOS di atas, hlm TOS dipilih sebagai $S \in \mathbb{R}^{n \times p}$. kemudian ruang fitur yang disempurnakan, $Feature\ Space_{comb}$, dibuat dengan menggabungkan yang asli fitur *X* oleh *S*, seperti $Feature\ Space_{comb} = [X, S] \epsilon \mathbb{R}^{n \times (d+p)}$. Itu dicatat bahwa ($k - p$) TOS dibuang untuk meningkatkan efisiensi dan prediksi algoritma.

C. *Fase III : Prediksi menggunakan XGBoost*

Klasifikasi XGBoost diterapkan pada $Feature\ Space_{comb}$ menghasilkan hasil akhir. Memanfaatkan XGBoost, waktu berjalan efisiensi dan kemampuan prediksi algoritma ditingkatkan karena ketahanannya terhadap ketidakseimbangan data dan overfitting. Selain itu, proses pemangkasan pasca mungkin dilakukan oleh kepentingan fitur internal XGBoost. Itu Pentingnya fitur dihitung pada jumlah fitur dalam node pemisahan, ketika model dipasang. TOS lebih agresif pemangkasan dengan demikian dimungkinkan, mis. memilih top *q* paling penting TOS dari *S* oleh peringkat fitur internal.

D. *Theoretical Foundations*

Pengorbanan Bias-Variance banyak digunakan untuk memahami kesalahan generalisasi dari suatu algoritma klasifikasi. Baru saja, Aggarwal dan Sathe telah menunjukkan teori yang serupa framework juga berlaku untuk ensemble outlier [15]. Di dalam lihat, ansambel outlier memiliki dua jenis kesalahan yang dapat direduksi: (i) bias kuadrat, disebabkan oleh kemampuan terbatas untuk mencocokkan data dan (ii)

Algorithm 1 Balance Selection

Input: $\Phi = \{\Phi_i, \ldots, \Phi_k\}$, ground truth *y* , # of TOS = *p*
Output: The Set of Selected TOS: *S*
Initialize: Selected TOS: *S* ={}

1. $\Phi(X)_{max} = \max\left(ACC\left(\Phi(X)\right)\right)$ /* most accurate*/
2. $S \leftarrow S \cup \Phi(X)_{max}$ /*add selected TOS to set S */
3. $\Phi(X) \leftarrow \Phi(X) \backslash \Phi(X)_{max}$ /*remove from the pool*/
4: **while** #(S) < p **do**
5: **for** $\Phi(X)_i \epsilon\ \Phi(X)$ **do**
6. $\Psi(\Phi_i) \leftarrow$ Eq. (5) /*discounted accuracy*/
7. $\Phi\ (X)_{max}$ max $(\Psi(X)_i)$
8. $S \leftarrow S \cup \Phi(X)$ /*add the current best to set S */
9. $\Phi(X) \leftarrow \Phi(X)/\Phi(X)_{max}$ /*remove from the pool*/
10. **end for**
11. **end while**
12. **return S**

varians, yang disebabkan oleh sensitivitas terhadap data pelatihan. Sebuah ansambel outlier yang efektif harus berhasil mengendalikan reducible error, mengingat bahwa mengurangi bias dapat meningkatkan varians dan sebaliknya.

Dalam penelitian ini, berbagai pendeteksi outlier tanpa pengawasan algoritma digunakan untuk memperkaya ruang fitur, yang menyuntikkan keragaman ke dalam model dan kemudian menggabungkan hasilnya. Ini adalah dianggap sebagai pendekatan pengurangan varians, sebagai menggabungkan beragam detektor dasar mengurangi varian ensemble outlier [3], [12], [15]. Namun, ini dapat menyebabkan TOS menjadi tidak akurat termasuk dalam ansambel, menimbulkan bias model yang lebih tinggi. Ini menjelaskan mengapa ensemble lengkap (rata-rata semua TOS) tidak berkinerja baik — mungkin termasuk beberapa detektor basis yang tidak akurat dengan bias tinggi [3]. Jadi, algoritma pemilihan desain TOS hanya menyimpan yang bermanfaat untuk mengurangi bias. Apalagi itu mekanisme ensemble dan regularisasi di XGBoost bisa mencapai varian rendah tanpa menimbulkan banyak bias [20]. Dengan berbagai instrumen untuk mengurangi bias dan varians, XGBOD adalah dianggap dapat meningkatkan kemampuan generalisasi di semua tahap. Namun, kinerja XGBOD mungkin heuristik dan tidak dapat diprediksi dengan set data patologis atau pilihan yang buruk fungsi deteksi outlier dasar tanpa pengawasan.

IV. DESAIN EKSPERIMEN

Analisi perbandingan dengan berbagai metode termasuk. ROC [1], [10], [11] banyak digunakan untuk melakukan evaluasi terhadap deteksi outlier pada data. Pada eksperimen ini akan melakukan 10 kali iterasi independen yang nantinya hasi dari rata-rata iterasi tersebut digunakan sebagai skor akhir perhitungan. Dataset yang digunakan pada penelitian ini yaitu sebanyak 3 data, berikut tabel 1 merupakan gambaran dari dataset yang digunakan.

TABEL I. DATASET PENELITIAN

Dataset	Points (n)	Features (d)	Outlier
Arrhythmia	452	274	66 (15%)
Cardio	1.831	21	176 (9.6%)
Letter	1.600	32	100 (6.25%)

A. *Dataset Outlier*

Tabel 1 merupakan dataset yang digunakan pada penelitian kali ini. Dataset tersebut sebelumnya sudah banyak digunakan dalam melakukan pengamatan oleh berbagai peneliti dalam mendeteksi outlier [2], [11], [15], [25], dan

dapat diakses secara publik dalam *Outlier detection respository* yang disebut odds [26]. Penelitian ini membagi dataset awal kedalam dua bagian yaitu *data training* dan data uji dengan perbandingan 60% sebagai *data training* dan 40% sebagai data uji.

B. Fungsi Penilaian Outlier Dasar dan Pengaturan Parameter

Efektivitas XGBOD bergantung pada akurasi dan keanekaragaman dari fungsi penilaian outlier dasar. Berbagai macam jenis algoritma deteksi outlier yang termasuk kedalam kelompok unsupervised termasuk pada XGBOD. Proses pemangkasan dan kemudian memilih yang paling berguna. Pada penelitian ini menggunakan lima fungsi score dasar outlier yaitu (I) KNN (jarak Euclidean dari tetangga terdekat k sebagai skor outlier), (II) K-Median, (III) Rata-rata KNN (rata-rata k jarak tetangga terdekat sebagai skor outlierness), (IV) LOF [27], dan (V) LoOP [28]. Perlu diketahui bahwa KNN, K-Median, rata-rata KNN merupakan kelompok algoritma *unsupervised* tanpa persyaratan dari kebenaran dasar. Pada penelitian ini untuk mendorong keragaman, fungsi skor dasar yang digunakan bervariasi. Untuk algoritma dengan basis kelompok tetangga terdekat seperti KNN, K-Median, Avg-KNN, LOF, dan LoOP nilai k kisaran yang digunakan yaitu 1, 2, 3, 4, 5, 10, 15, 20, 30, 40, dan 50. Sedangkan untuk LoOP mengingat bahwa algoritma LoOP secara komputasi mahal pada dataset besar, rentang k yang digunakan yaitu lebih sempit dengan nilai 1, 3, 5, dan 10. Proses ini dilakukan dengan cara yang sama terhadap 3 dataset yang digunakan pada penelitian ini dalam melakukan fungsi score outlier dasar dan pengaturan parameter.

C. Pengaturan Eksperimen

Eksperimen 1 yaitu membandingkan performansi antar *framework* yang berbeda. Diantaranya terdapat semua TOS (menggunakan TOS) dan tanpa TOS (tidak menggunakan TOS, atau *original*). Beberapa poin dasar yang dibandingkan,

(i) *Best_TOS* : *score* tertinggi diantara semua TOS (*unsupervised*)
(ii) *Full_TOS* : rata-rata *score* dari semua *score* semua TOS
(iii) *XGB_Orig* : XGBoost no TOS pada data asli
(iv) *XGB_New* : XGBoost dengan TOS yang baru saja dihasilkan
(v) *XGB_Comb* : XGBoost dengan penggabungan fitur asli dan semua TOS pada data yang terkombinasi
(vi) *L1_Comb* : *L*1 *logistic regression* pada data yang terkombinasi, menggunakan *EasyEnsemble* (50 bagging)
(vii) *L2_Comb* : *L2 logistic regression* pada data yang terkombinasi, menggunakan *EasyEnsemble* (50 bagging)

Keterangan: *XGBoost* pada eksperimen I dan II menggunakan 100 estimator dasar dengan default maksimal kedalaman *tree* yaitu 3.

Eksperimen II menganalisa efek dari pemilihan TOS (TOS *selection*). *XGBoost* hanya digunakan sebagai pengklasifikasi. Oleh karena itu, memilih nol TOS sama saja dengan *XGB_Orig* dan memilih all TOS sama saja dengan *XGB_Comb* pada eksperimen I. Pemilihan TOS akan menghasilkan *feature space* yang berbeda. Hasil dari pemilihan jumlah TOS yang berbeda dianalisa. Disamping jumlah nya, algoritma dalam memilih TOS (*Random* Selection, *Accurate* Selection, dan *Balance* Selection) juga dibandingkan. Perlu diketahui bahwa memilih satu TOS saja dengan *Accurate Selection* sama sama dengan menggunakan *Balance Selection*.

V. HASIL DAN DISKUSI

A. Analisa Performansi Prediksi

Tabel II menunjukan hasil dari **Eksperimen I** yang telah dilakukan, yang mana secara langsung membandingkan metode klasifikasi tanpa adanya pemilihan TOS. Pada tabel, Semua pendekatan *unsupervised* ditandai dengan *****, dan semua metode yang diterapkan pada *feature space* terkombinasi ditandai dengan #. Hasil performansi terbaik dari tiap dataset ditandai dengan huruf tebal. Dapat dilihat bahwa, *XGB_Comb* mendapatkan nilai terbaik pada ketiga dataset yaitu Arrhythmia, Cardio, Letter. Pada dataset Cardio nilai performansi *XGB_Comb* hampir mendekati nilai performansi *XGB_Orig*, walaupun memang masih lebih unggul. Diambil kesimpulan bahwa semakin kecil *feature space* pada dataset, maka bisa jadi tidak semua metode *unsupervised* yang dipilih dapat mengambil representasi yang berguna karena *feature space* yang terbatas. Hal ini dapat mengakibatkan hasil performansi *XGB_Comb* dapat berada di bawah *XGB_Orig*. Sebaliknya, *unsupervised representation learning* meningkatkan pengklasifikasi *supervised* pada dataset yang memiliki *feature space* besar. Oleh karena itu, *unsupervised representation learning* amat berguna untuk dataset dengan dimensi *feature* yang besar.

Perlu diketahui bahwa, keluaran akhir pengklasifikasi harus dipilih dengan hati-hati saat menggunakan *unsupervised representation learning*. Melihat dari hasilnya, mengatakan bahwa kombinasi antara model sederhana dengan *EasyEnsemble* bisa jadi dapat digantikan oleh algoritma *Ensemble* yang beregulasi kuat, seperti *XGBoost*, untuk hasil yang lebih baik. Walaupun diketahui bahwa, data yang tidak seimbang akan cocok jika menggunakan pengklasifikasi akhir *logistic* regression karena penerapan *EasyEnsemble* nya yang dapat membuat subsampel yang seimbang, namun ternyata ketika dibandingkan dengan *XGB_Comb* dengan pada *featue space* gabungan (fitur asli + TOS yang baru dibuat) hasilnya lebih bagus *XGB_Comb*, tidak hanya itu, jika dibandingkan juga dengan *XGB_Orig* (tanpa TOS), hasil performansi L1_*Comb* dan L2_*Comb* juga kalah pada ketiga dataset tersebut.

TABEL II. PERFORMANSI TIAP MODEL

Datasets	ROC						
	Best_TOS*	**Full_TOS***	**L1_Comb#**	**L2_Comb#**	**XGB_Orig**	**XGB_New**	**XGB_Comb#**
Arrhythmia	0.8278	0.7937	0.83392	0.8341	0.84259	0.77088	**0.86426**
Cardio	0.578	0.309	0.99328	0.9944	0.99548	0.92596	**0.99761**
Letter	0.9463	0.9446	0.92587	0.91189	0.934	0.92165	**0.96832**

Ditemukan bahwa *L1_Comb* menghasilkan *score* yang tidak jauh berbeda dengan *L2_Comb*, dan telah dikonfirmasi oleh *Wilcoxon rank-sum test*, bahwa secara statistik memang tidak jauh berbeda.

B. Jumlah TOS yang Dipilih

Umumnya, menggunakan sebuah subset dari TOS biasanya menghasilkan hasil yang lebih baik dibanding menggunakan semua TOS, memilih sejumlah kecil TOS untuk digabungkan dengan fitur asli dapat meningkatkan hasil yang signifikan. Pengamatan ini dapat dijelaskan dari desain *XGBoost* sendiri, yang mana dapat mempelajari fitur penting secara otomatis dengan mengidentifikasi fitur yang paling sering terpecah dari dasar *tree* [20].

Selain menggunakan gabungan *feature space* (fitur asli + semua TOS), menggunakan TOS yang baru dihasilkan juga terkadang memberikan hasil yang sangat baik, bahkan lebih baik dari *feature space* asli (tanpa TOS), ini membuktikan bahwa penggunaan TOS mampu merepresentasikan data dan juga menyiratkan bahwa fitur asli mungkin tidak diperlukan untuk pengklasifikasi akhir, namun perlu diketahui bahwa hasil ini juga tidak konsisten, karena terkadang hasilnya rendah pada sebagian dataset. Fenomena yang sama ini juga telah diamati dengan baik pada penelitian [10], [11]. Dengan demikian, fitur asli tidak sepenuhnya dapat digantikan oleh TOS, maka solusi terbaiknya adalah dengan mengecek hasil dari ketiga penerapan *XGBoost* ini.

C. Metode Pemilihan TOS

Random Selection lebih banyak memberi hasil yang tidak pasti. Hasil performansi yang diberikan jika dibandingkan metode lain terkadang menurun ketika menggunakan satu TOS, dibandingkan tanpa TOS, namun juga terkadang menghasilkan performansi yang tinggi, metode ini kurang dapat diprediksi atau kurang stabil. Penulis menganggap ketidak konsistenan ini disebabkan oleh *EasyEnsemble* pada [10], [11] kurang stabil dibandingkan *XGBoost* dan juga dataset yang berbeda, menciptakan karakteristik yang berbeda juga.

Sedangkan *Balance* dan *Accurate selection* sendiri mungkin bergantung pada dimensi pada *feature space* yang asli. Secara empiris, *Balance selection* menghasilkan hasil yang bagus pada dataset dengan jumlah fitur yang banyak, dan sebaliknya, untuk fitur yang berjumlah sedikit *Accurate selection* lebih unggul. Asumsi nya ialah, *Accurate selection* cenderung memilih TOS dihasilkan dengan tipe spesifik dari metode pendeteksi *outlier* dengan parameter berbeda, namun berat jika harus mengambil representasi yang berguna dari data berdimensi besar. Sebagai alternatifnya, *Balance selection* mendukung keberagaman, yang mengarah ke hasil yang lebih baik dengan memilih berbagai jenis TOS dan bebas dari kesalahan.

Ketika jumlah TOS yang dipilih meningkat, ketiga metode pemilihan ini menjadi lebih sebanding karena jumlah TOS yang bertumpuk juga banyak, sehingga menggunakan *XGB_Comb* bisa menjadi pilihan yang aman di sebagian besar kasus.

D. Keterbatasan dan Rencana Kedepan

Sejumlah penelitian sedang dilakukan. Pertama, TOS diambil dari fitur asli secara langsung. Namun, memilih fitur pada data asli akan menghilangkan beberapa data yang tidak penting, dan TOS dapat dibangun pada fitur yang dipilih. Kedua, banyak metode pemilihan TOS dapat digabungkan pada studi selanjutnya. Selain itu, pemilihan TOS dapat digantikan dengan metode preduksi dimensi 2018 *International Joint Conference on Neural Network* (IJCNN), seperti PCA, intinya TOS dapat digabungkan dibandingkan dipilih.

VI. K E S I M P U L A N

Algoritma *ensemble semi-supervised* terbaru bernama XGBOD (*Extreme Gradient Boosting Outlier Detection*), diusulkan untuk pendeteksian *outlier* pada beberapa dataset. XGBOD merupakan hasil dari 3 tahap sistem yaitu (i) menggunakan algoritma pendeteksi *outlier unsupervised*, (ii) memanfaatkan *greedy selection* untuk menghasilkan representasi yang berguna, dan (iii) menerapkan sebuah pengklasifikasi *XGBoost* untuk memprediksi pada *feature space* yang ditingkatkan. Eksperimen pada 3 dataset ber *outlier* menunjukan bahwa XGBOD memberi hasil performansi yang lebih baik dibandingkan pesaingnya, yang mana didukung oleh teori pertimbangan indikasi dalam pengurangan variasi dan bias.

Desain XGBOD ini termotivasi dari penelitian sebelumnya oleh Micenková dkk. [10], [11] dan Aggarwal dkk. [12], yang mana mengusulkan bahwa metode pendeteksi *outlier unsupervised* dapat menghasilkan representasi data *outlier* yang besar dibandingkan *feature space* asli. Lebih spesifiknya, menerapkan beberapa algoritma *unsupervised* yang sudah diakui untuk mendeteksi *outlier* pada data asli dapat menghasilkan TOS yang lebih baik dalam merepresentasikan data nya. Selanjutnya, menggabungkan TOS dengan *feature space* asli dapat meningkatkan prediksi *outlier* secara keseluruhan.

Penelitian ini memperluas daripada penelitian sebelumnya, yang mana bahkan menggunakan sedikit TOS saja dapat meningkatkan hasil pendeteksian *outlier* secara signifikan. Untuk mengontrol proses komputasi, dirancang dan diujilah tiga algoritma *selection*. Rekomendasi pemilihan, penggunaan, serta interpretasi nya juga telah disediakan. Umumnya, *balance selection* diusulkan untuk dataset berdimensi *feature space* yang tinggi, *accurate selection* disarankan untuk dataset dengan fitur yang lebih sedikit, dan *random selection* mungkin berguna dalam beberapa kasus, karena hasilnya yang sulit diprediksi.

Dibandingkan dengan metode *ensemble outlier semi-supervised* yang lainnya, XGBOD memberi hasil prediksi yang lebih baik, menghilangkan ketergantungan,

meningkatkan efisiensi waktu, lebih stabil, tidak memerlukan *feature scaling* atau *imputer* dalam *data preprocessing*. XGBOD merupakan *framework* pertama yang menggabungkan representasi *outlier unsupervised* dengan *supervised machine learning* dan menggunakan *ensemble tree*. Catatan akhir, semua *source code*, *dataset*, dan *gambar* yang diuji disini, dibagikan secara terbuka dan tersedia[1].

REFERENCES

[1] J. R. Pasillas-Díaz and S. Ratté, "An Unsupervised Approach for Combining Scores of Outlier Detection Techniques, Based on Similarity Measures," *Electron. Notes Theor. Comput. Sci.*, vol. 329, pp. 61–77, 2016.
[2] C. C. Aggarwal, "Outlier ensembles: position paper," *SIGKDD Explorations*, vol. 14, no. 2, pp. 49–58, 2013.
[3] S. Rayana and L. Akoglu, "Less is More: Building Selective Anomaly Ensembles," *TKDD*, vol. 10, no. 4, pp. 1–33, 2016.
[4] A. Zimek, R. J. G. B. Campello, and J. Sander, "Ensembles for unsupervised outlier detection: Challenges and research questions," *SIGKDD Explorations*, vol. 15, no. 1, pp. 11–22, 2014.
[5] T. G. Dietterich, "Ensemble Methods in Machine Learning," *Mult. Classif. Syst.*, vol. 1857, pp. 1–15, 2000.
[6] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[7] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.
[8] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.
[9] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," *KDD*, p. 157, 2005.
[10] B. Micenková, B. McWilliams, and I. Assent, "Learning Representations for Outlier Detection on a Budget." 29-Jul-2015.
[11] B. Micenková, B. McWilliams, and I. Assent, "Learning Outlier Ensembles : The Best of Both Worlds – Supervised and Unsupervised," *ODD Workshop on SIGKDD*, pp. 1–4, 2014.
[12] C. C. Aggarwal and S. Sathe, *Outlier ensembles: An introduction*. 2017.
[13] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *KDD*, pp. 785–794, 2016.
[14] T. Y. Liu, "EasyEnsemble and feature selection for imbalance data sets," in *Proceedings - 2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, IJCBS 2009*, 2009, pp. 517–520.
[15] C. C. Aggarwal and S. Sathe, "Theoretical Foundations and Algorithms for Outlier Ensembles?," *SIGKDD Explorations*, vol. 17, no. 1, pp. 24– 47, 2015.
[16] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *PAMI*, vol. 35, no. 8, pp. 1798–1828, 2013.
[17] G. Forestier and C. Wemmert, "Semi-supervised learning using multiple clusterings with limited labeled data," *Inf. Sci.*, vol. 361–362, pp. 48–65, Sep. 2016.
[18] N. Moniz and P. Branco, "Evaluation of Ensemble Methods in Imbalanced Regression Tasks," *Proc. First Int. Work. Learn. with Imbalanced Domains Theory Appl.*, vol. 74, pp. 129–140, 2017.
[19] N. García-Pedrajas, J. Pérez-Rodríguez, M. García-Pedrajas, D. OrtizBoyer, and C. Fyfe, "Class imbalance methods for translation initiation site recognition in DNA sequences," *Knowledge-Based Syst.*, vol. 25, no. 1, pp. 22–34, Feb. 2012.
[20] D. Nielsen, "Tree Boosting With XGBoost Why Does XGBoost Win 'Every' Machine Learning Competition?," 2016.
[21] C. C. Aggarwal, *Outlier analysis*, vol. 9781461463. 2013.
[22] N. F. Rajani and R. J. Mooney, "Supervised and Unsupervised Ensembling for Knowledge Base Population," *Proc. 2016 Conf. Empir. Methods Nat. Lang. Process.*, pp. 1943–1948, 2016.
[23] M. Friedman, "The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance," *J. Am. Stat. Assoc.*, vol. 32, no. 200, pp. 675–701, 1937.
[24] P. Nemenyi, "Distribution-free Multiple Comparisons," Princeton University, 1963.
[25] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," *ICDM*, pp. 413–422, 2008.
[26] S. Rayana, "ODDS Library," 2016. [Online]. Available: http://odds.cs.stonybrook.edu.
[27] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying Density-Based Local Outliers," *SIGMOD*, pp. 1–12, 2000.
[28] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "LoOP: local outlier probabilities," *CIKM*, pp. 1649–1652, 2009.
[29] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines," *IJCNN*, vol. 3, pp. 1741–1745, 2003.
[30] L. Van Der Maaten and G. Hinton, "Visualizing Data using t-SNE," *JMLR*, vol. 9, pp. 2579–2605, 2008.