

1. Pendahuluan

1.1 Latar belakang

Perkembangan dunia industri yang menghasilkan data dalam jumlah yang besar mendorong manusia untuk mencari informasi yang berguna di balik data tersebut. Salah satu bentuk informasi yang bisa dicari adalah klasifikasi suatu data menjadi kelas-kelas tertentu. Jaringan Syaraf Tiruan (JST) sebagai salah satu teknik klasifikasi dalam *data mining* menawarkan berbagai keunggulan seperti mampu menangani klasifikasi dimana kelas targetnya lebih dari dua kelas (*non binary classification*), mampu menangani masalah *non linear*, mampu menangani atribut data yang *redundant*, dan mampu mengklasifikasikan suatu obyek dengan cepat apabila model dari hasil pembelajaran telah didapat.

Akan tetapi seringkali data yang disediakan dalam dunia nyata memiliki distribusi yang tidak seimbang antar kelas. Dalam *data mining*, hal ini disebut dengan masalah *imbalance class*. Hal ini menimbulkan masalah tersendiri dalam teknik klasifikasi yang digunakan dalam *data mining*. JST yang pada awalnya bertujuan untuk memaksimalkan akurasi dalam mengenali pola menjadi cenderung untuk mengenali pola yang dominan dan mengabaikan pola minoritas. Padahal terkadang dalam suatu data, kelas minoritas justru memiliki informasi yang bernilai lebih daripada kelas mayoritas.

Oleh karena itu dibutuhkan suatu pendekatan yang mampu mengoptimalkan kemampuan JST untuk lebih sensitif terhadap kelas minoritas. Pendekatan ini dapat berupa pendekatan *sampling*. Ide dasar dari *sampling* adalah untuk memodifikasi distribusi dari data sehingga kelas minoritas dapat direpresentasikan dengan baik dalam data *training* [7]. Beberapa jenis *sampling* yang ada antara lain *undersampling*, *oversampling*, dan gabungan dari *undersampling* dan *oversampling*.

Untuk menemukan struktur dan bobot JST yang optimal tidaklah mudah. *Evolutionary Algorithms* (EA) dapat menjadi solusi bagi masalah ini. Ketertarikan penggunaan EA untuk mendesain arsitektur JST telah berkembang dalam tahun-tahun belakangan karena EA mampu berevolusi mencari arsitektur optimal tanpa intervensi dari luar dan menghilangkan pencarian struktur optimal secara manual yang membosankan dan melelahkan [1]. Algoritma-algoritma yang termasuk dalam EA antara lain Algoritma Genetika, *Evolution Strategies*, *Evolutionary Programming*, dan *Genetic Programming*. *Evolutionary Programming* (EP) sebagai salah satu EA memiliki keunggulan dibandingkan EA lainnya dalam pencarian struktur dan bobot JST. Dengan karakter EP yang hanya menggunakan mutasi tanpa rekombinasi dalam menghasilkan anak, membuat EP dapat mengurangi efek negatif dari masalah permutasi [12].

1.2 Perumusan masalah

Masalah pokok yang akan diteliti adalah :

1. Bagaimana menerapkan EP untuk mencari bobot dan arsitektur JST.
2. Bagaimana menerapkan metode *sampling* untuk menangani masalah *imbalance class*.

Dalam penyusunan tugas akhir ini permasalahan dibatasi dalam beberapa hal, yaitu :

1. Perangkat lunak yang dihasilkan untuk menangani studi kasus Data Mining Cup 2007 (DMC 2007).
2. JST yang digunakan adalah *Feed Forward Neural Networks* dengan *supervised learning*.
3. EP digunakan untuk menentukan bobot dan arsitektur JST saja.

1.3 Tujuan

Tujuan dari penelitian tugas akhir ini adalah sebagai berikut :

1. Mengimplementasikan EANN dengan menggunakan EP untuk mencari struktur dan bobot JST yang optimal.
2. Menerapkan *sampling* untuk memodifikasi data *training* agar EANN dapat mengklasifikasikan kelas *imbalance* dengan baik.
3. Menghitung performansi EANN menggunakan *recall*, *precision*, *F-measure*, akurasi, dan parameter yang telah ditetapkan dalam DMC 2007.

1.4 Metodologi penyelesaian masalah

Metodologi yang akan digunakan dalam merealisasikan tujuan dan pemecahan masalah di atas adalah dengan menggunakan langkah-langkah berikut.

1. Studi Literatur :
 - a. Pencarian referensi, mencari referensi dan sumber-sumber lain yang layak dan berhubungan dengan *data mining*, JST, EP, dan *sampling*.
 - b. Pendalaman materi, mempelajari dan memahami materi yang berhubungan dengan tugas akhir.
2. Mempelajari konsep dari JST dan EP serta *sampling* yang akan digunakan dalam implementasi perangkat lunak.
3. Melakukan analisis terhadap parameter-parameter JST, EP, dan *sampling*.
4. Melakukan implementasi perancangan perangkat lunak.
5. Melakukan pengujian perangkat lunak dengan memasukkan data yang akan dievaluasi serta mencatat hasil keluaran program.
6. Pengambilan kesimpulan dan penyusunan laporan tugas akhir.