Seminar I

MENGKOMBINASIKAN TEKNIK RESAMPLING DENGAN ALGORITMA MACHINE LEARNING UNTUK MENGATASI DATASET TAK SEIMBANG

Oleh :

ZINEDINE KAHLIL GIBRAN ZIDANE

H13116304

Pembimbing Utama : **Dr. Amran, S.Si., M.Si.**

Pembimbing Pertama : **Supri Bin Hj Amir, S.Si., M.Eng.**

Penguji : **1. Dr. Anna Islamiyati, S.Si, M.Si.**

2. Nur Hilal A Syahrir, S.Si, M.Si.

PROGRAM STUDI ILMU KOMPUTER

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

2019

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam beberapa tahun terakhir, perkembangan yang pesat dalam sains dan teknologi telah berdampak pada pertumbuhan data mentah secara eksponensial [1]. Berdasarkan World Economic Forum [2], data digital di dunia akan mencapai 44 zettabyte, atau 44 triliun gigabyte pada tahun 2020. Jumlah ini akan terus berkembang hingga lebih dari dua kali lipat setiap dua tahunnya [3, 4].

Dari pertumbuhan data tersebut, kebutuhan untuk menganalisis data terus meningkat [5]. Menurut laporan dari LinkedIn pada tahun 2018, permintaan pekerjaan yang membutuhkan analisis data berkembang hingga 12 kali lipat lebih banyak dari tahun 2014 [6]. Ini disebabkan karena data merupakan kunci dari setiap organisasi, institusi atau perusahaan untuk mengambil keputusan yang lebih cermat dan efektif [7].

Namun data mentah atau *raw data* tidak dapat memberikan informasi yang berguna. Data tersebut perlu diolah dan dianalisis menjadi informasi yang berguna. *Raw data* yang dimaksud adalah data yang belum diproses, yang terstruktur dan yang tidak terstruktur. Data yang terstruktur adalah data yang disimpan dengan format yang telah ditentukan seperti database [8], di mana atribut data dapat dibedakan dengan jelas sehingga dapat langsung diproses oleh peralatan komputasi [9]. Sedangkan data tidak terstruktur adalah data yang disimpan dalam format yang tidak terstruktur, sehingga membutuhkan campur tangan manusia agar dapat diinterpretasi oleh mesin; seperti dokumen, gambar, video dan audio [10]. Menurut laporan Beal [8], 80% hingga 90% data di dunia tidak terstruktur.

Dengan jumlah data yang sangat banyak, tidak mungkin oleh manusia untuk menganalisis dan membuat perhitungan mengenai data secara manual. Maka dari itu diperlukan bidang khusus untuk mengolah dan menganalisis data. *Data Science* adalah

bidang yang mempelajari bagaimana meng-ekstraksi *raw data* menjadi *meaningful information* atau informasi yang berguna [11, 12]. Data Science melibatkan prinsip, proses, dan teknik untuk memahami fenomena melalui data [13]. Data Science merupakan bidang yang sangat luas dan sedang dikembangkan [13], namun salah satu bidang khusus dari data science adalah *machine learning* yang merupakan perpotongan antara *computer science* (ilmu komputer) dan *statistics* (statistika) [14]. Machine learning membahas mengenai bagaimana membangun sistem komputer yang dapat belajar melalui pengalaman tanpa harus diprogram secara spesifik dan manual [14] [15].

Untuk menyelesaikan suatu masalah pada komputer, dibutuhkan algoritma. Algoritma adalah kumpulan instruksi-instruksi yang berurutan yang digunakan untuk membawa suatu input menjadi output tertentu [16]. Namun untuk beberapa masalah, kita tidak mempunyai algoritmanya. Contohnya adalah bagaimana komputer mengenali angka dalam bentuk tulisan tangan [17] dan mengklasifikasi suatu email menjadi spam atau bukan spam [16]. Dalam hal ini, kita dapat mengumpulkan seluruh email yang berlabel spam dan yang bukan spam dan "mempelajari" apa yang membedakan mereka. Di dalam hal ini lah machine learning bekerja.

Data yang dikumpulkan biasanya dalam bentuk dataset atau tabel, di mana setiap kolomnya adalah atribut atau ciri-ciri dan setiap barisnya adalah instansi atau observasi. Dataset tersebut ada yang memiliki kolom label, atau kolom yang berisi informasi mengenai kategori dari setiap observasi (contohnya, spam atau bukan spam), dan ada juga dataset yang tidak memiliki kolom label, di mana isinya hanyalah atribut atau ciri-ciri dari setiap observasi, tanpa mengindikasikan kategori dari tiap observasi [17]. Pembelejaran pada dataset berlabel disebut *supervised learning*. Kasus di mana tujuannya adalah mengklasifkasikan input data ke suatu kategori diskrit tertentu disebut *klasifikasi*, dan kasus di mana outputnya adalah suatu variabel kontinu disebut *regresi*. Selain itu, pembelajaran pada dataset tanpa label atau acuan kategori yang benar disebut *unsupervised learning*. Kasus *unsupervised learning* di mana tujuannya

adalah mengelompokkan observasi-observasi yang mirip disebut *clustering*, jika menentukan distribusi data pada input disebut *estimasi kepadatan*. Dan yang terakhir, pembelajan di mana mesin dilatih untuk membuat keputusan tertentu dengan cara *trial and error* diseut *reinforcement learning* [18]. Masing-masing jenis pembelajaran memiliki banyak algoritma yang telah dikembangkan dengan berbagai pendekatan yang berbeda-beda [14]. Berdasarkan dokumentasi dari *scikit-learn* [19], terdapat lebih dari 100 algoritma machine learning yang ada.

Untuk distribusi data pada suatu dataset, terdapat istilah kelas yang terdistribusi secara seimbang (*balanced*) dan secara tak seimbang (*imbalanced*). Dataset dengan kelas yang seimbang berarti jumlah observasi untuk setiap kelas tidak jauh dari kelas-kelas yang lain [20]. Sedangkan untuk dataset dengan distribusi kelas yang tak seimbang, jumlah suatu observasi pada kelas tertentu sangat jauh berbeda dengan kelas yang lain. Hal ini berlaku pada dataset dengan kelas biner (dua kelas saja) dan juga *multiclass* (lebih dari dua kelas) [1]. Kelas dengan jumlah observasi sedikit disebut kelas minoritas (*minority class*) dan kelas dengan jumlah observasi yang sangat banyak disebut kelas mayoritas (*majority class*). Tidak jarang suatu dataset terdistribusi secara tak seimbang dengan proporsi antara kelas minoritas dan mayoritasnya adalah 1:100, 1:1000, atau 1:1000 [1]. Sebagian besar data asli di dunia terdistribusi secara tak seimbang [1, 21, 22, 23].

Pada umumnya, algoritma-algoritma machine learning, dalam hal ini pada masalah klasifikasi, bekerja dengan tujuan utama memaksimalkan akurasi [24]. Hal ini sangat masuk akal, karena akurasi yang tinggi menjelaskan bahwa model algoritma tersebut melaksanakan tugasnya dengan baik, mengklasifikasikan kelas data dengan benar dengan sedikit kesalahan. Namun, akurasi hanya memberikan informasi secara umum, bagaimana jika model algoritma tersebut bekerja pada dataset tak seimbang, dan hanya mampu mengklasifikasikan kelas mayoritas dengan benar tetapi tak mampu mengklasifikasikan kelas minoritas? Jika perbandingan antara kelas minoritas dan mayoritas saja satu berbanding seratus, maka kita akan mendapatkan akurasi lebih

besar dari 99%, dengan kesalahan lebih kecil dari 1% yang hampir seluruhnya adalah kelas minoritas. Masalah ini memberi bias terhadap performa algoritma-algoritma klasifikasi, terutama jika kelas yang lebih utama untuk diklasifikasikan dengan benar adalah kelas minoritas, seperti email spam, diagnosis penyakit di bidang kedokteran, deteksi kartu kredit palsu dan lain-lain [23, 25]. Hal ini menunjukkan bahwa pada kasus dataset tak seimbang, dibutuhkan perhatian lebih terhadap *preprocessing* data sebelum dimasukkan ke model.

Banyak cara yang telah ditemukan untuk mengatasi dataset tak seimbang ini, seperti melakukan *resampling* terhadap data yang ada. Resampling adalah teknik mengambil sampel secara berulang dari sampel data asli [26]. Teknik resampling terdiri dari *oversampling*, yaitu mengambil sampel berulang kali dari kelas minoritas; dan undersampling, yaitu mengambil sampel secara acak dari kelas mayoritas [27]. Kedua teknik ini dapat digunakan secara terpisah ataupun digabung [27, 28, 29, 30]. Masing-masing teknik resampling memiliki metode yang berbeda-beda, dengan performa yang berbeda pula [27, 29]. Begitu juga dengan algoritma klasifikasi pada machine learning, terdapat banyak metode yang berbeda dengan performa yang berbeda pula [31].

Dalam beberapa penelitian terkait [29, 32, 33, 27], telah dilakukan berbagai percobaan untuk mengatasi masalah dataset tak seimbang, namun metode-metode resampling maupun algoritma machine learning yang digunakan tidak beragam untuk mengetahui metode terbaik untuk mengatasi masalah ini. Seperti penelitian yang dilakukan Amin dkk. [33] hanya meneliti teknik oversampling, [27, 29, 32] meneliti teknik oversampling dan undersampling namun hanya menggunakan satu algoritma machine learning, sedangkan Diri [31] hanya meneliti beberapa algoritma machine learning tanpa pertimbangan dataset tak seimbang. Sedangkan untuk mengetahui metode resampling dan algoritma machine learning terbaikuntuk masalah ini, dibutuhkan kombinasi-kombinasi antar teknik resampling, dan juga antar algoritma machine learning. Setiap kombinasi (pasangan) ini, seperti SMOTE dengan Support

Vector Machine, atau Tomek Links dengan Naive Bayes Classifier akan diuji performanya terhadap dataset yang diberikan, kemudian dari kombinasi-kombinasi tersebut dapat ditarik kesimpulan mengenai kombinasi algoritma dan teknik resampling yang terbaik, dan algoritma machine learning dengan performa terbaik, dan teknik resampling dengan performa terbaik. Setiap kombinasi atau pasangan dievaluasi hasilnya dengan tidak hanya pada satu dataset tak seimbang saja, melainkan dengan beberapa dataset tambahan untuk mendapatkan hasil yang lebih umum dan tanpa bias.

Berdasarkan uraian di atas, penulis ingin melakukan penelitian mengenai dataset tak seimbang dengan cara "Mengkombinasikan Teknik Resampling Dan Algoritma Machine Learning Untuk Mengatasi Dataset Tak Seimbang".

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang masalah di atas, dapat dikemukakan pertanyaan penelitian sebagai berikut:

1. Algoritma Machine Learning yang mana kah yang memiliki performa terbaik dalam mengklasifikasi dataset tak seimbang?
2. Teknik Resampling yang mana kah yang memiliki performa terbaik pada dataset tak seimbang?
3. Kombinasi algoritma Machine Learning dan teknik Resampling yang mana kah yang memiliki performa terbaik untuk mengatasi dataset tak seimbang?
4. Teknik resampling yang manakah yang lebih baik digunakan? Oversampling, undersampling, atau gabungan keduanya?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Dataset yang digunakan adalah dataset tak seimbang.
2. Dataset hanya memiliki dua kelas (biner).

3. Teknik resampling yang digunakan adalah Random Undersampling, Tomek Links, SBC, Random Oversampling, SMOTE, MSMOTE, Borderline-SMOTE, dan ADASYN.
4. Algoritma machine learning yang digunakan adalah Regresi Logistik, Decision Tree, Random Forest, Neural Network, Nearest Neighbor, dan Support Vector Machine.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan dari penelitian ini adalah:

1. Mengetahui algoritma machine learning yang memiliki performa terbaik dalam mengklasifikasi dataset tak seimbang.
2. Mengetahui teknik resampling yang memiliki performa terbaik pada dataset tak seimbang.
3. Mengetahui kombinasi algoritma machine learning dan teknik resampling yang memiliki performa terbaik dalam mengatasi dataset tak seimbang.
4. Mengetahui apakah oversampling, undersampling, atau gabungan keduanya yang lebih baik digunakan untuk masalah dataset tak seimbang.

1.5 Manfaat Penelitian

Hasil penelitian ini diharapkan dapat bermanfaat:

1. Sebagai rujukan untuk mengatasi dataset tak seimbang yang sangat sering dijumpai.
2. Menjadi sumber informasi mengenai performa beberapa teknik resampling.
3. Menjadi sumber informasi mengenai performa dari beberapa algoritma machine learning.

BAB II
TINJAUAN PUSTAKA

2.1 Machine Learning

Machine learning adalah memprogram komputer untuk mengoptimalkan suatu ukuran kinerja menggunakan sampel data atau berdasarkan pengalaman [16]. Machine learning menggunakan suatu algoritma untuk menganalisis data.

Dalam pembelajaran yang terawasi atau *supervised learning*, pengklasifikasi akan diberikan suatu input tertentu dan menghubungkannya dengan suatu output. Kasus di mana tujuannya adalah mengklasifkasikan input data ke suatu kategori diskrit tertentu disebut *klasifikasi*, dan kasus di mana outputnya adalah suatu variabel kontinu disebut *regresi*. Dalam pembelajaran tanpa pengawasan atau *unsupervised learning*, pengklasifikasi diberi input dan dibiarkan sendiri untuk menemukan pola pada data tersebut. Kasus *unsupervised learning* di mana tujuannya adalah mengelompokkan observasi-observasi yang mirip disebut *clustering*, jika menentukan distribusi data pada input disebut *estimasi kepadatan*. Dalam *reinforcement learning*, sistem komputer menerima input secara terus menerus dan mencoba memilih keputusan-keputusan yang paling optimal berdasarkan kondisi lingkungannya.

Masing-masing jenis pembelajaran memiliki banyak algoritma yang telah dikembangkan dengan berbagai pendekatan yang berbeda-beda [14].

2.2 Dataset

Dataset adalah kumpulan data yang berbentuk tabel, di mana setiap kolomnya merepresentasikan suatu ciri-ciri, atribut atau fitur. Setiap barisnya menyatakan observasi suatu individu, record atau sampel [34]. Suatu dataset biasanya memiliki satu kolom tambahan yang merepresentasikan kelas dari observasi tersebut, kolom ini disebut kolom kelas. Kolom kelas ini juga disebut sebagai variabel dependen terhadap variabel-variabel independen yang merupakan ciri-ciri (atribut) dari suatu observasi tertentu.

Dalam machine learning dikenal istilah dataset yang tak seimbang. Istilah ini berlaku ketika kelas dari dataset tersebut bersifat kategorik diskrit. Dataset yang tak seimbang (*imbalanced dataset*) adalah dataset yang frekuensi kejadian dari kelas tertentu sangat jauh berbeda dengan kelas yang lain. Contohnya seperti suatu dataset dengan jumlah pasien yang berkelas "diabetes" jumlahnya jauh lebih sedikit dibanding pasien yang "tidak diabetes".

Masalah ketidakseimbangan ini akan memberi bias terhadap performa pengklasifikasi sebab jumlah sampel pada kelas tertentu tidak dapat memberi informasi yang cukup kepada pengklasifikasi berdasarkan ciri-ciri yang diberikan [18, 16, 15].

2.3 ROC Curve

Di dalam machine learning, mengukur kinerja atau performa dari suatu model adalah hal yang esensial. Model yang diperoleh dari pelatihan melalui data training perlu diuji melalui data testing. Kinerja diukur berdasarkan seberapa baik model tersebut memprediksi dengan benar data yang ada.

Pada klasifikasi biner, kelas positif yang berhasil diprediksi dengan benar disebut true positive, jika kelas positif tersebut diprediksi negatif (salah) disebut false negative. Kelas negatif yang berhasil diprediksi negatif (benar) disebut true negative, dan kelas negatif yang diprediksi positif disebut false positive. Jumlah dari kasus-kasus tersebut direpresentasikan dalam suatu tabel kontingensi yang disebut *confusion matrix* [35].

	Kelas asli		
Hasil prediksi		Positif	Negatif
	Positif	TP	FP
	Negatif	FN	TN

Table 1 Confusion Matrix

Akurasi adalah ukuran kinerja yang menunjukkan seberapa baik suatu pengklasifikasi dalam mengklasifikasikan seluruh data. Akurasi adalah rasio antara observasi yang diklasifikasikan secara benar dengan total observasi:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (1)$$

Presisi adalah ukuran kinerja yang menunjukkan seberapa besar kebenaran suatu pengklasfikasi dari seluruh kelas positif yang diprediksi. Presisi adalah rasio antara jumlah kelas positif yang diklasifikasikan secara benar dengan jumlah observasi yang diklasifikasikan positif:

$$Precision = \frac{TP}{(TP + FP)} \quad (2)$$

Recall atau sensitivitas adalah ukuran kinerja yang menunjukkan seberapa baik suatu pengklasifikasi dalam mengklasifikasikan kelas positif. Recall adalah rasio antara jumlah observasi positif yang diklasifikasikan secara benar dengan jumlah observasi positif asli:

$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

ROC (Receiver Operating Characteristic) Curve atau Kurva ROC adalah teknik yang diproposalkan oleh Swets [35] untuk meringkas kinerja pengklasifikasi berdasarkan rasio positif asli (true positive) dengan positif palsu (false positive). Kurva ROC menunjukkan kinerja suatu model berdasarkan threshold atau nilai ambang batas tertentu yang digunakan pada model tersebut.

Nilai AUC (*Area Under Curve*) menunjukkan nilai peluang suatu model memprediksi kelas positif sebagai positif dan kelas negatif sebagai negatif, atau seberapa baik suatu model membedakan antara kelas positif dan negatif. Semakin tinggi nilai AUC, maka semakin baik model tersebut dalam membedakan atara kelas positif dan negatif. [35,

36, 37]. Nilai AUC 1 menunjukkan bahwa model tersebut dapat dengan sempurna membedakan antara kelas positif dengan kelas negaitf, dan nilai AUC 0,5 menunjukkan bahwa model tersebut sama sekali tidak dapat membedakan antara kelas positif dan negatif [37].

2.4 Jarak Euklid

Jarak Euklid adalah jarak antara suatu vektor $X = (x_1, \dots, x_n)$ ke suatu vektor $Y = (y_1, \dots, y_n)$ pada ruang euklid berdimensi n:

$$d(X, Y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2} \quad (4)$$

Pada machine learning, jarak Euklid digunakan untuk menghitung jarak antar dua observasi berdasarkan vektor fitur yang bersifat kontinu [38].

2.5 Teknik Resampling

2.5.1 Random Oversampling

Random oversampling, atau oversampling secara acak adalah teknik oversampling di mana anggota dari kelas minoritas dipilih secara acak dan diduplikasi ke dataset yang baru hingga tercapai keseimbangan [39]. Data minoritas tersebut dapat diduplikasi beberapa kali. Teknik ini biasanya menyebabkan overfitting pada model [33, 39].

2.5.2 Synthetic Minority Oversampling Technique (SMOTE)

SMOTE atau Synthetic Minority Oversampling Technique adalah teknik oversampling terpopuler yang diproposalkan oleh Chawla [40] pada tahun 2002. Teknik ini membuat data tiruan atau sintetik berdasarkan tetangga-tetangga terdekat dari sampel kelas minoritas.

Teknik ini dimulai dengan menentukan N data yang akan dibuat untuk setiap data pada kelas minoritas dalam dataset D. Kemudian untuk setiap data kelas minoritas $M_{i,c}$, pilih N tetangga secara acak dari k tetangga terdekat data tersebut, di mana $i = \{1,2, \dots m\}$ dengan m adalah jumlah data pada kelas minoritas dan $c \in C$, di mana $C =$

$\{c_1, c_2, \dots, c_z\}$ adalah fitur-fitur pada D. Lalu untuk setiap fitur c pada $M_{i,c}$, hitung jarak d antara $M_{i,c}$ dengan setiap tetangga tersebut:

$$d = M_{i,c} - T_{s,c}$$

di mana s adalah bilangan acak $s = \{1,2,..,k\}$ dari k tetangga terdekat $M_{i,c}$ dengan T adalah salah satu tetangga terdekat M. Kemudian suatu bilangan acak $g = [0,1]$ ditentukan. Data tiruan dibuat berdasarkan:

$$S_{j,c} = M_{i,c} + (g * d) \quad (5)$$

di mana $j = \{1,2,..,N * k\}$ bersifat inkremental dan S adalah data kelas minoritas tiruan.

Teknik ini membuat data sintetik S pada suatu titik dari jarak antara setiap fitur dari M dengan T [40].

2.5.3 Borderline – Synthetic Minority Oversampling Technique (Borderline-SMOTE)

Terinspirasi oleh SMOTE, Han memproposalkan teknik oversampling baru yang menyerupai SMOTE namun hanya dengan membuat data tiruan di sekitar data kelas minoritas yang berada di *borderline* atau perbatasan antara kelas mayoritas dan kelas minoritas saja di mana SMOTE membuat N data tiruan pada setiap sampel kelas minoritas [41].

Misalkan untuk data training D terdapat kelas minoritas P dan kelas mayoritas N, diekspresikan:

$$P = \{p_1, p_2, \dots, p_{pnum}\}, N = \{n_1, n_2, \dots, n_{nnum}\}$$

di mana $pnum$ dan $nnum$ adalah jumlah sampel kelas minoritas dan jumlah sampel kelas mayoritas. Borderline-SMOTE bekerja sebagai berikut.

Untuk setiap $p_i(i = 1,2, \dots, pnum)$ pada kelas minoritas P, hitung m tetangga terdekat dari seluruh data training D. Jumlah kelas mayoritas dari m tetangga terdekat p_i disimbolkan $m'(0 \leq m' \leq m)$.

Jika $m' = m$, atau seluruh tetangga terdekat dari p_i adalah sampel kelas mayoritas, maka p_i dinyatakan sebagai noise dan bukan merupakan sampel perbatasan. Jika $m/2 \leq m' < m$, atau tetangga terdekat p_i lebih banyak merupakan sampel kelas mayoritas dibanding minoritas, maka p_i dianggap sampel perbatasan dan dimasukkan ke dalam himpunan sampel perbatasan B. Jika $0 \leq m' < m/2$, maka p_i dianggap aman dan bukan merupakan sampel perbatasan.

Untuk himpunan sampel perbatasan B, di mana $B \subseteq P$. Maka:

$$B = \{p'_1, p'_2, \dots, p'_{dnum}\}, 0 \leq dnum \leq pnum$$

di mana $dnum$ adalah jumlah sampel perbatasan. Untuk setiap sampel perbatasan di B, tentukan k tetangga terdekat dari P.

Data tiruan berkelas minoritas dibuat sebanyak $s * dnum$ $(s = 1, 2, .., k)$. Untuk setiap p'_i, Pilih sebanyak s tetangga terdekat secara acak dari P, lalu hitung jarak $d_j(j = 1,2, \dots, s)$ antara p'_i dengan s tetangga terdekatnya dari P di mana:

$$d_j = p'_i - p_j$$

adalah jarak antara p'_i dengan salah satu tetangga terdekatnya. Kemudian suatu bilangan acak g $(g = [0,1])$ ditentukan dan data tiruan sebanyak s dari p'_i dibuat.

$$S_j = p'_i + (g * d_j) \quad (6)$$

Proses di atas dilakukan untuk setiap p'_i, maka data tiruan akan dibuat sebanyak $s * dnum$ kali [41].

2.5.4 Adaptive Synthetic (ADASYN)

Adaptive Synthetic adalah teknik oversampling yang diproposalkan oleh He di mana data tiruan dibuat berdasarkan tingkat kesulitan suatu sampel kelas minoritas untuk dipelajari. Setiap sampel kelas minoritas memiliki alokasi data tiruan sesuai dengan banyaknya tetangga sampel kelas mayoritas dari k tetangga terdekat sampel kelas minoritas tersebut [42].

Misalkan untuk data training D dengan m sampel $\{x_i, y_i\}$, $i = 1, \dots, m$, di mana x_i adalah instansi dari n dimensi dari ruang fitur X dan $y_i \in \{1, -1\}$ adalah label kelas dari x_i. Didefinisikan m_s dan m_l sebagai jumlah sampel kelas minoritas, dan jumlah sampel kelas mayoritas secara berurutan. Maka $m_s \leq m_l$ dan $m_s + m_l = m$.

Tentukan derajat ketidakseimbangan d:

$$d = \frac{m_s}{m_l} \tag{7}$$

di mana $d \in (0, 1]$. Jika $d < d_{th}$ di mana d_{th} adalah nilai maksimum toleransi derajat ketidakseimbangan, maka hitung jumlah data kelas minoritas tiruan:

$$G = (m_l - m_s) * \beta \tag{8}$$

di mana $\beta \in [0, 1]$ adalah parameter yang menunjukkan rasio ketidakseimbangan yang diinginkan setelah data tiruan dibuat. $\beta = 1$ menunjukkan dataset akan seimbang sepenuhnya.

Untuk setiap sampel $x_i \in P$ di mana P adalah kelas minoritas, tentukan K tetangga terdekat dengan jarak Euklid pada n dimensi ruang fitur. Kemudian hitung rasio r_i, yang didefinisikan sebagai:'

$$r_i = \frac{\Delta_i}{K}, i = 1, \dots, m_s \tag{9}$$

di mana Δ_i adalah jumlah sampel pada K tetangga terdekat x_i yang merupakan sampel kelas mayoritas, maka $r_i \in [0, 1]$.

Kemudian normalisasikan r_i menjadi:

$$\hat{r}_i = \frac{r_i}{\sum_{i=1}^{m_s} r_i} \tag{10}$$

agar $\hat{r}_i$ menjadi distribusi kepadatan ($\sum_i \hat{r}_i = 1$).

Hitung data tiruan yang akan dibuat untuk setiap sampel x_i:

$$g_i = \hat{r}_i * G \tag{11}$$

Untuk setiap sampel $x_i \in P$, buat g_i data tiruan dengan persamaan:

$$s_i = x_i + (x_{zi} - x_i) * \lambda \tag{12}$$

di mana $(x_{zi} - x_i)$ adalah selisih vektor pada n dimensi dan λ adalah bilangan acak $\lambda \in [0, 1]$.

2.5.5 Random Undersampling

Random undersampling, atau undersampling secara acak adalah teknik undersampling di mana anggota dari kelas mayoritas dipilih secara acak dan dihapus dari dataset training hingga tercapai keseimbangan. Kekurangan dari teknik ini adalah kita tidak dapat mengatur informasi apa saja yang dihilangkan dari dataset tersebut, informasi yang berguna bisa saja hilang [33, 39, 30, 29].

2.5.6 Tomek Links

Tomek Link atau tautan Tomek adalah metode undersampling yang diproposalkan oleh Tomek [43] untuk memodifikasi CNN (Condensed Nearest Neighbor). Teknik ini menghapus sampel kelas mayoritas jika dan hanya tetangga kelas mayoritas tersebut berasal dari kelas berbeda dan merupakan tetangga terdekat satu sama lain. Jika x dan y adalah sampel dengan kelas berbeda maka Tomek Link didefinisikan untuk setiap sampel z:

$$d(x, y) < d(x, z) \wedge d(x, y) < d(y, z)$$

di mana fungsi $d(.)$ adalah jarak Euklid (4).

2.6 Algoritma Klasifikasi Machine Learning

2.6.1 Logistic Regression

Regresi logistik adalah algoritma klasifikasi yang menggunakan kelas untuk membangun dan menggunakan model regresi logistik multinomial tunggal dengan *estimator* tunggal. Regresi logistik menyatakan probabilitas kelas (antara 0 dan 1) tergantung pada jarak dari batas, dengan sutu pendekatan. Hasil regresi logistik adalah suatu bilangan antara 0 dan 1 yang menyatakan probabilitas kelas.

Nilai atribut pada data observasi dari regresi logistik dapat berupa nominal, ordinal, interval atau skala rasio, sedangkan untuk atribut kelas harus merupakan kelas biner. Hubungan antara atribut dengan kelas bersifat non-linear. Distribusi data tidak tersebar dalam bentuk distribusi Gauss, melainkan distribusi Bernoulli, yang dikarenakan kelasnya berbentuk biner.

Pada analisis regresi logistik, hubungan antara suatu kejadian berdasarkan atribut atau vektor fiturnya diekspresikan:

$$p = \frac{1}{1 + e^{-z}} \tag{13}$$

di mana p adalah peluang terjadinya suatu kejadian. Peluang tersebut bernilai nol hingga satu pada kurva berbentuk S dan z adalah kombinasi linear dari vektor fitur kejadian tersebut:

$$z = b_o + b_1 x_1 + b_2 x_2 + \cdots + b_n x_n$$

di mana b_0 adalah nilai *intercept* dari model, dan $b_i (i = 0,1, \dots, n)$ adalah koefisien kemiringan atau gradien dari model regresi logistik tersebut, dan $x_i (i = 1,2, \dots, n)$ adalah nilai atribut ke-i dari vektor fitur kejadian tersebut [44, 45, 46].

2.6.2 Naive Bayes

Naive Bayes adalah algoritma klasifikasi yang memprediksi kelas suatu data berdasarkan vektor fitur atau atribut-atributnya. Pembelajaran pengklasifikasi ini mengasumsikan bahwa setiap fitur adalah kelas yang independen, yaitu:

$$P(X|C) = \prod_{i=1}^{n} P(X_i|C) \tag{14}$$

di mana $X = (X_1, \dots, X_n)$ adalah fitur vektor dan C adalah kelas [47].

Berdasarkan aturan Bayes, peluang sampel $E = (x_1, x_2, \dots, x_n)$ berlabel suatu kelas $C = \{1, -1\}$ biner adalah:

$$P(C|E) = \frac{P(E|C) * P(C)}{P(E)} \tag{15}$$

dan E diklasifikasikan $C = 1$ jika dan hanya jika:

$$B(E) = \frac{P(C = 1|E)}{P(C = -1|E)} \geq 1 \tag{16}$$

di mana $B(E)$ adalah pengklasifikasi Bayes. Asumsikan seluruh atribut bersifat independen terhadap variabel kelas, yaitu:

$$P(E|C) = P(x_1, x_2, \dots, x_n|C) = \prod_{i=1}^{n} P(x_i|C) \tag{17}$$

maka hasil pengklasifikasinya adalah:

$$B_{nb}(E) = \frac{P(C = 1)}{P(C = -1)} \prod_{i=1}^{n} \frac{P(x_i|C = 1)}{P(x_i|C = -1)} \tag{18}$$

B_{nb} disebut pengklasifikasi naive Bayes [48, 49, 50].

2.6.3 Decision Tree

Decision Tree atau pohon keputusan adalah pengklasifikasi yang menentukan output kelas dari suatu sampel berdasarkan keputusan yang diambil dari setiap nilai atribut dari sampel tersebut. Pohon keputusan adalah suatu fungsi Boolean di mana inputnya adalah suatu sampel E dengan vektor fitur X dan outputnya adalah 0 atau 1. Di pohon keputusan, setiap node pohon yang bukan node daun adalah suatu uji atribut atau suatu ekspresi boolean, setiap node daun adalah nilai Boolean, dan setiap cabang mewakili salah satu nilai yang mungkin dari atribut yang diuji. Ada beberapa metode pohon keputusan seperti ID3, C4.5, dan CART.

Algoritma ID3 menggunakan teori entropi informasi yang menghitung entropi dari masing-masing atribut. Kemudian menghitung Information Gain dari suatu atribut berdasarkan entropinya terhadap target kelas. Di mana entropi adalah ukuran ketidapastian suatu atribut yang terkait dengan suatu kejadian. Semakin kecil entropi dari suatu atribut, semakin tinggi kemurnian informasi yang ada. Semakin besar entropi dari suatu atribut, semakin besar ketidakpastian informasi tersebut. Information Gain adalah jumlah ketidakpastian informasi yang berkurang berdasarkan informasi yang diterima terkait dengan suatu atribut.

Misalkan S adalah data Training dengan jumlah sampel s, dengan m jumlah nilai atribut kelas yang berbeda dari $C_i (i = 1,2, \dots, m)$. Misalkan S_i adalah jumlah sampel pada C_i. Informasi yang dibutuhkan dari suatu sampel klasifikasi adalah:

$$I(S_1, \dots, S_m) = -\sum_{i=1}^{m} p_i \, log_2(p_i) \tag{19}$$

di mana p_i adalah peluang suatu sampel berasal dari kelas C_i yang diperoleh dari s_i/s. Suatu himpunan atribut A memiliki n nilai berbeda $\{a_1, \dots, a_n\}$. Bagi S menjadi n subset $\{S_1, \dots, S_n\}$, di mana S_j memiliki beberapa sampel di S yang memiliki nilai A_j di A.

Misalkan s_{ij} adalah jumlah sampel di kelas C_i dari subset S_j, entropi subset yang dibagi oleh A adalah:

$$E(A) = -\sum_{j=1}^{n} \frac{s_{1j} + s_{2j} + \cdots + s_{mj}}{s} * I(s_{1j}, s_{2j}, \ldots, s_{mj)} \quad (20)$$

Berdasarkan persamaan di atas, informasi yang dibutuhkan dari suatu subset S_j dihitung berdasarkan:

$$I(s_{1j}, s_{2j}, \ldots, s_{mj}) = -\sum_{i=1}^{m} p_{ij}\, log_2 p_{ij} \quad (21)$$

Maka Information Gain dari A dapat dihitung dengan:

$$Gain(A) = I(s_1, s_2, \ldots, s_m) - E(A) \quad (22)$$

Algoritma ID3 membangun decision tree berdasarkan urutan atribut yang memiliki information gain tertinggi sebagai root node. Node tersebut kemudian bercabang sesuai dengan jumlah nilai berbeda pada atribut tersebut. Cabang dengan entropi nol adalah node daun, dan cabang dengan entropi yang lebih dari nol membutuhkan pecabangan lebih lanjut. Proses tersebut dilakukan secara rekursif pada semua node yang bukan merupakan node daun hingga seluruh data terklasifikasi [46, 51, 52, 53].

2.6.4 Support Vector Machine

Support Vector Machine adalah algoritma pengklasifikasi yang membangun suatu hyperplane yang memisahkan data-data dengan kelas berbeda dengan margin sebesar mungkin.

Misalkan suatu data training pada sampel $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ dipisahkan menjadi dua kelas, di mana $x_i \in R^n$ adalah vektor fitur dan $y_i \in \{-1, 1\}$ adalah label kelasnya. Jika diasumsikan dua kelas tersebut dapat dipisahkan dengan hyperplane $w * x + b = 0$ pada suatu bidang H, dan kita tidak memiliki informasi mengenai distribusi datanya, maka hyperplane yang optimal adalah hyperplane yang memaksimalkan

margin. Nilai optimal dari w dan b dapat dicari menggunakan fungsi Lagrange $\alpha_i (i = 1, \dots, m)$:

$$f(x) = sgn\left(\sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b\right) \tag{23}$$

di mana α_i dan b diperoleh dari algoritma pembelajaran SVC. x_i di mana nilai α_i tidak nol adalah "support vector". Untuk $K(x_i, x)$ adalah inner product atau hasil kali dalam antara dua observasi $i \in \{1, \dots, n\}$ dan $i' \in \{1, \dots, n\}$ yang diekspresikan:

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j} \tag{24}$$

yang disebut kernel Linear.

Metode SVM yang menggunakan kernel linear dapat memisahkan kelas-kelas secara linear. Namun secara praktis dataset yang dijumpai sering tidak dapat dipisahkan secara linear. Kernel non-linear lebih fleksibel dikarenakan kernel ini memetakan variabel p dan x ke dimensi yang lebih tinggi. Salah satu kernel non-linear yang populer digunakan adalah kernel radial atau kernel Gauss:

$$K(x_i, x_{i'}) = exp\left\{-\gamma \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2\right\} \tag{25}$$

mana $\gamma > 0$ adalah parameter tuning tambahan. Ketika suatu sampel test terletak jauh dari sampel training, eksponensial tersebut akan menjadi negatif, dan $K(x_i, x_{i'})$ mendekati nol. Pada kasus ini, sampel training yang terletak jauh dari sampel test hampir tidak memiliki efek atau kontribusi kepada keputusan yang digunakan untuk mengklasifikasikan sampel test. Parameter γ mempengaruhi seberapa jauh letak suatu observasi agar dapat berkontribusi terhadap keputusan klasifikasi. Kernel ini lebih fleksibel dari kernel linear [54, 55, 56].

2.6.5 Multilayer Perceptron (MLP)

Artificial Neural Network (ANN) atau Jaringan Saraf Tiruan adalah salah satu algoritma machine learning yang mencari suatu fungsi f yang tidak diketahui berdasarkan input vektor X ke suatu output y:

$$y = f(X)$$

Pada tahap pelatihan, fungsi f dioptimasikan sedemikian sehingga hasil yang didapat untuk setiap vektor X yang diberikan sedekat mungkin dengan nilai y.

Istilah neuron atau node digunakan pada ANN untuk menandakan suatu atribut (pada layer input), suatu jenis output (pada layer output) atau suatu ekspresi matematika (pada layer tersembunyi). Neuron atau node ini masing-masing memiliki *weight* (bobot) yang berbeda-beda yang mempengaruhi nilai output dari node tersebut berdasarkan suatu fungsi aktivasi. Untuk node pada layer non-output, output node tersebut akan dikirim ke seluruh node pada layer selanjutnya. Koneksi atau sambungan antara node memiliki nilai bias.

Multilayer Perceptron (MLP) adalah salah satu jenis Artificial Neural Network yang memiliki layer tesembunyi atau hidden layer. MLP terdiri dari setidaknya 3 layer, yaitu input layer, hidden layer dan output layer. Secara praktis, MLP biasanya memiliki lebih dari satu hidden layer.

MLP dapat diekspresikan sebagai berikut:

$$\hat{y} = v_0 + \sum_{j=1}^{NH} v_j g\left(w_j^T x'\right) \tag{26}$$

di mana x' adalah vektor input x yang diaugmentasikan dengan 1, atau $x' = (1, x^T)^T$, dan w_j adalah vektor bobot dari node tersembunyi ke-j, dan $v_0, v_1, \dots, v_{NH}$ adalah bobot untuk node output, dan $\hat{y}$ adalah output jaringannya. Fungsi g adalah fungsi aktivasi pada node tersembunyi [57, 58, 59, 60, 18, 17, 61].

Fungsi aktivasi adalah fungsi yang memberi output dari suatu node berdasarkan input yang diberikan.

Fungsi aktivasi Threshold atau Unit Step adalah fungsi aktivasi yang memberi nilai 0 atau 1 sesuai dengan threshold atau nilai batas yang ditentukan:

$$f(x) = \begin{cases} 0, & 0 > x \\ 1, & x \geq 0 \end{cases}$$

Fungsi aktivasi Sigmoid logistik adalah fungsi aktivasi yang memberi nilai antara 0 dan 1:

$$f(x) = \frac{1}{1 + e^{-\beta x}} \tag{27}$$

Fungsi aktivasi Sigmoid tangensial adalah fungsi aktivasi yang memberi nilai antara -1 dan 1

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{28}$$

2.6.6 K-Nearest Neighbor

K-Nearest Neighbor adalah algoritma prediksi non-parametrik di mana hasil prediksi kelas dari suatu titik didasarkan oleh mayoritas kelas dari k tetangga terdekatnya. Diberikan suatu titik, hitung jarak Euklid antara titik tersebut dengan semua titik pada data training. Kemudian pilih k tetangga terdekat berdasarkan jarak Euklid tersebut, prediksi kelas dari titik tersebut adalah modus kelas dari k tetangga terdekatnya.

Untuk suatu sampel x dengan k tetangga terdekat B, prediksi kelas ditentukan oleh:

$$\hat{y} = \frac{1}{k} \sum_{m \in B} y_m \tag{29}$$

di mana y_m adalah label kelas dari x_m [57, 46].

BAB III
METODE PENELITIAN

3.1 Waktu dan Tempat

Penelitian ini dilaksanakan dari bulan Agustus 2019 sampai dengan bulan Oktober 2019. Lokasi penelitian dilakukan di Laboratorium Rekayasa Perangkat Lunak Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin

3.2 Tahapan Penelitian

Untuk menyelesaikan penelitian ini, peneliti akan melewati beberapa tahap penelitian, yaitu: Pra-penelitian, eksplorasi data, model *tuning and fitting*, dan analisis hasil.

Pada tahap pra-penelitian, peneliti menentukan tema penelitian, masalah yang akan diteliti, mengumpulkan sumber referensi atau literatur seperti jurnal dan buku yang mendukung dalam penelitian, dan menentukan metode yang digunakan beserta batasan masalahnya. Kemudian peneliti mencari data yang sesuai dengan tema penelitian sebagai objek penelitian.

Pada tahap eksplorasi data, peneliti mencoba menguraikan karakteristik-karakteristik data yang digunakan untuk menentukan fungsi-fungsi tertentu yang akan digunakan. Setiap dataset akan memberikan parameter-parameter optimal yang berbeda terhadap algoritma klasifikasi machine learning.

Pada tahap model *tuning and fitting*, peneliti akan mencari parameter-parameter terbaik untuk model yang akan digunakan berdasarkan hasil eksplorasi data dan *trial and error* untuk mendapatkan hasil terbaik. Tuning juga dilakukan terhadap beberapa teknik resampling yang membutuhkan parameter. Kemudian model akan memberi hasil prediksi yang akan dianalisis pada tahap selanjutnya.

Pada tahap analisis hasil, peneliti akan merangkum hasil yang diperoleh dari metode-metode yang digunakan ke dalam bentuk tabel dan diagram, kemudian menyimpulkan hasilnya sebagai output dari penelitian ini.

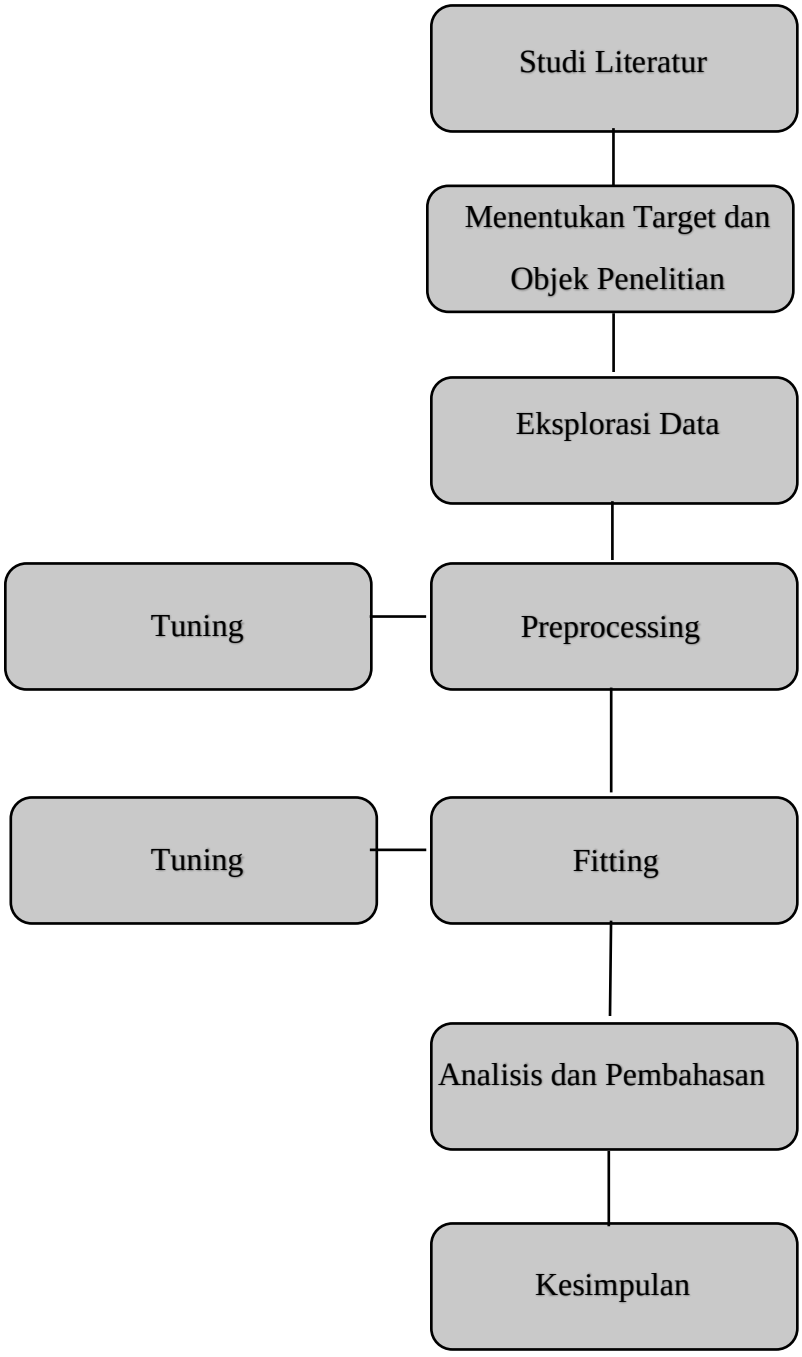
3.3 Deskripsi Data

Data diambil dari Website resmi Kaggle (kaggle.com), UCI Machine Learning Repository (archive.ics.uci.edu/ml/) dan KEEL (sci2s.ugr.es/keel/imbalanced.php). Data tersebut berupa tiga dataset, yaitu:

1. Credit Card Fraud Dataset (Kaggle), yang terdiri dari 30 kolom attribut dengan 1 kolom kelas, 284.807 baris. 284.315 jumlah sampel kelas mayoritas dan 492 jumlah sampel kelas minoritas dengan *imbalanced ratio* sebesar 577:1. Dataset ini merupakan dataset terpopuler di Kaggle sebab jumlah data yang besar dengan imbalanced ratio yang sangat tinggi, yang
2. Spambase Dataset (UCI), yang terdiri dari 57 kolom atribut dengan 1 kolom kelas, 4.601 baris. 2788 jumlah sampel kelas mayoritas dan 1813 jumlah sampel kelas minoritas dengan *imbalanced ratio* sebesar 1,5:1
3. Image Segmentation Dataset (KEEL), yang terdiri dari 19 kolom atribut dengan 1 kolom kelas, 2308 baris. 1962 jumlah sampel kelas mayoritas dan 346 jumlah sampel kelas minoritas dengan *imbalanced ratio* sebesar 6:1.

Seluruh dataset hanya memiliki atribut kontinu dengan label kelas biner yang sesuai dengan tema penelitian dan metode-metode yang digunakan.

3.4 Alur Penelitian



Daftar Pustaka

[1] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge & Data Engineering,* no. 9, pp. 1263-1284, 2008.

[2] J. Desjardins, "How much data is generated each day?," World Economic Forum, 2019.

[3] M. Chen, S. Mao and Y. & Liu, "Big Data: A survey," *Mobile networks and applications,* vol. 19, no. 2, pp. 171-209, 2014.

[4] S. Lohr, "The age of big data," New York Times, New York, 2012.

[5] N. Elgendy and A. Elragal, "Big data analytics: a literature review paper," *Industrial Conference on Data Mining,* pp. 214-227, 2014.

[6] LinkedIn Economic Graph Team, "Linkedin 2018 Emerging Jobs Report," LinkedIn, 2018.

[7] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. J. Patil and D. Barton, "Big data: the management revolution," *Harvard business review,* vol. 90, no. 10, pp. 60-68, 2012.

[8] V. Beal, "Unstructured Data," 2019. [Online]. Available: https://www.webopedia.com/TERM/U/unstructured_data.html. [Accessed 20 6 2019].

[9] H. Baars and H. G. Kemper, "Management support with structured and unstructured data—an integrated business intelligence framework," *Information Systems Management,* vol. 25, no. 2, pp. 132-148, 2008.

[10] G. Weglarz, "Two Worlds of Data - Unstructured and Structured," *DM Review,* vol. 14, pp. 19-23, 2004.

[11] F. Berman, R. Rutenbar, B. Hailpern, H. Christensen, S. Davidson, D. Estrin, M. Franklin, M. Martonosi, P. Raghavan, V. Stodden and A. S. Szalay, "Realizing the Potential of Data Science," *Communications Of The Acm,* vol. 61, no. 4, pp. 67-72, 2018.

[12] V. Dhar, "Data Science and Prediction," *Communications of the ACM,* vol. 56, no. 12, pp. 64-73, 2012.

[13] F. Provost and T. Fawcett, "Data Science and its Relationship to Big Data and Data-Driven Decision Making," *Big data,* vol. 1, no. 1, pp. 51-59, 2013.

[14] J. M. I. and M. T. M., "Machine learning: Trends, perspectives, and prospects," *Science,* vol. 349, no. 6245, pp. 255-260, 2015.

[15] P. Domingos, "A Few Useful Things to Know about Machine Learning," pp. 78-87, 2011.

[16] A. Ethem, Introduction to Machine Learning, MIT press, 2009.

[17] C. M. Bishop, Neural Networks for Pattern Recognition, Oxford university press, 1995.

[18] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Muller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and M. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research 12,* 2011.

[20] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 42, no. 4, pp. 463-484, 2011.

[21] S. Kotsiantis, D. Kanellopoulos and P. Pintelas, "Handling imbalanced datasets: A review," *GESTS International Transactions on Computer Science and Engineering,* vol. 30, no. 1, pp. 25-36, 2006.

[22] A. Kumar and H. Sheshadri, "On the Classification of Imbalanced Datasets," *International Journal of Computer Applications (0975 – 8887),* vol. 44, no. 8, pp. 1-7, 2012.

[23] S. Visa and A. Ralescu, "Issues in Mining Imbalanced Data Sets - A Review Paper," *Proceedings of the sixteen midwest artificial intelligence and cognitive science conference,* vol. 2005, pp. 67-73, April 2005.

[24] F. Provost, "Machine Learning from Imbalanced Data Sets 101," *Proceedings of the AAAI'2000 workshop on imbalanced data sets,* vol. 68, pp. 1-3, July 2000.

[25] M. M. Rahman and D. N. Davis, "Addressing the Class Imbalance Problem in Medical Datasets," *International Journal of Machine Learning and Computing,* vol. 3, no. 2, p. 224, 2013.

[26] Statistics Solution, "Statistics Solution," 2016. [Online]. Available: https://www.statisticssolutions.com/sample-size-calculation-and-sample-size-justification-resampling/. [Accessed 14 August 2019].

[27] E. Burnaev, P. Erofeev and A. Papanov, "Influence of Resampling on Accuracy of Imbalanced Classification," *In Eighth International Conference on Machine Vision (ICMV 2015),* vol. 9875, p. 987521, 2015.

[28] A. Anand, G. Pugalenthi, G. B. Fogel and P. N. Suganthan, "An approach for classification of highly imbalanced data using weighting and undersampling," *Amino acids,* vol. 39, no. 5, pp. 1385-1391, 2010.

[29] A. More, "Survey of resampling techniques for improving classification performance in unbalanced datasets," *arXiv preprint arXiv:1608.06048,* 2016.

[30] S. J. Yen and Y. S. Lee, "Under-Sampling Approaches for Improving Prediction of the Minority Class in an Imbalanced Dataset," *Intelligent Control and Automation,* pp. 731-740, 2006.

[31] B. Diri and S. Albayrak, "Visualization and analysis of classifiers performance in multi-class medical data," *Expert Systems with Applications,* vol. 34, no. 1, pp. 628-634, 2008.

[32] G. E. Batista, R. C. Prati and M. C. Monard, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data," *ACM SIGKDD explorations newsletter,* vol. 6, no. 1, pp. 20-29, 2004.

[33] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir and A. Hawalah, "Comparing Oversampling Techniques to Handle the Class Imbalance Problem: A Customer Churn Prediction Case Study," *IEEE Access,* vol. 4, pp. 7940-7957, 2016.

[34] C. Snijders, U. Matzat and U.-D. Reips, "'Big Data'. Big gaps of knowledge in the field of Internet," *International Journal of Internet Science,* vol. 7, pp. 1-5, 2012.

[35] J. A. Swets, "Measuring the Accuracy of Diagnostic Systems," *Science,* vol. 240, no. 4857, pp. 1285-1293, 1988.

[36] E. LeDell, M. Petersen and M. v. d. Laan, "Computationally efficient confidence intervals for cross-validated area under the ROC curve estimates," *Electronic Journal of Statistics,* vol. 9, pp. 1583-1607, 2015.

[37] A. P. Bradley, "The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms," *Pattern recognition,* vol. 30, no. 7, pp. 1145-1159, 1997.

[38] A. Howard, Elementary Linear Algebra, Binder Ready Version: Applications Version, John Wiley & Sons., 2013.

[39] A. Y.-c. Liu, "The effect of oversampling and undersampling on classifying imbalanced text datasets," *The University of Texas at Austin,* 2004.

[40] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of artificial intelligence research,* vol. 16, pp. 321-357, 2002.

[41] H. Han, W. Y. Wang and B. H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," *International conference on intelligent computing,* pp. 878-887, August 2005.

[42] H. He, Y. Bai, E. A. Garcia and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence),* pp. 1322-1328, June 2008.

[43] I. Tomek, "Two Modifications of CNN," *IEEE Trans. Systems, Man and Cybernetics,* vol. 6, pp. 769-772, 1976.

[44] P. Tsangaratos and I. Ilia, "Comparison of a logistic regression and Naïve Bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size," *Catena,* vol. 145, pp. 164-179, 2016.

[45] D. W. Hosmer and S. Lemeshow, Applied Logistic Regression (Second Edition), Canada: Wiley-Interscience Publication, 2000.

[46] O. F.Y., A. J.E.T., A. O., Hinmikaiye, Olakanmi and Akinjobi, "Supervised Machine Learning Algorithms: Classification and Comparison," *International Journal of Computer Trends and Technology (IJCTT),* vol. 48, no. 3, pp. 128-138, 2017.

[47] I. Rish, "An empirical study of the naive Bayes classifier," *Watson Research Center,* 2011.

[48] D. D. Lewis, "Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval," *European conference on machine learning,* pp. 4-15, 1998.

[49] I. Rish, "An Empirical Study of the Naive Bayes Classifier," *IJCAI 2001 workshop on empirical methods in artificial intelligence,* pp. 41-46, 2001.

[50] H. Zhang, "The Optimality of Naive Bayes," *American Association for Artificial Intelligence,* 2004.

[51] J. Quinlan, "Induction of Decision Trees," *Machine Learning,* vol. 1, pp. 81-106, 1986.

[52] Q.-y. Dai, C.-p. Zhang and H. Wu, "Research of Decision Tree Classification Algorithm in Data Mining," *International Journal of Database Theory and Application,* vol. 9, no. 5, pp. 1-8, 2016.

[53] T. Vafeiadis, K. Diamantaras, G. Sarigiannidis and K. Chatzisavvas, "A comparison of machine learning techniques for customer," *Simulation Modelling Practice and Theory,* vol. 55, no. 1, pp. 1-9, 55.

[54] N. Guenther and M. Schonlau, "Support vector machines," *The Stata Journal,* vol. 16, no. 4, pp. 917-937, 2016.

[55] C. Schuldt, I. Laptev and B. Caputo, "Recognizing human actions: a local SVM approach," *Proceedings of the 17th International Conference on Pattern Recognition,* vol. 3, pp. 32-36, 2004.

[56] V. Vapnik and C. Cortes, "Support-vector networks," *Machine Learning,* vol. 20, no. 3, pp. 273-297, 1995.

[57] N. K. Ahmed, A. F. Atiya, N. E. Gayar and H. El-Shishiny, "An Empirical Comparison of Machine Learning Models for Time Series Forecasting," *Econometric Reviews,* vol. 29, no. 5-6, pp. 594-621, 2010.

[58] J. M. Zurada, Introduction to artificial neural systems, St. Paul: West publishing company, 1992.

[59] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall PTR, 1994.

[60] Z. H. Zhou and X. Y. Liu, "Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem," *IEEE Transactions on Knowledge & Data Engineering,* no. 1, pp. 63-77, 2006.

[61] A. N. N. (. MULTILAYER, "Artificial Neural Networks (The Multilayer Perceptron) - A Review of Applications in the Atmospheric Sciences," *Atmospheric Environment,* vol. 32, no. 14-15, pp. 2627-2636, 1998.