

The 26th Annual Meeting in Mathematics
and
The 1st International Annual Meeting in Mathematics 2022
(AMM 2022)

Conference Proceedings

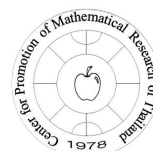


“ Frontiers in Mathematics for Smart and Sustainable Development ”

คณิตศาสตร์แนวหน้าสำหรับการพัฒนาแบบฉลาดและยั่งยืน

May 18-20, 2022

การจัดประชุมวิชาการนานาชาติ ครั้งที่ 1
จัดโดย สมาคมคณิตศาสตร์แห่งประเทศไทยในพระบรมราชูปถัมภ์
ร่วมกับ สาขาวิชาคณิตศาสตร์ สำนักวิชาวิทยาศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
และสาขาวิชาคณิตศาสตร์และสถิติประยุกต์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครราชสีมา

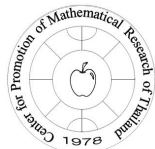




AMM
The 26th Annual Meeting
in Mathematics **2022**

AMM 2022

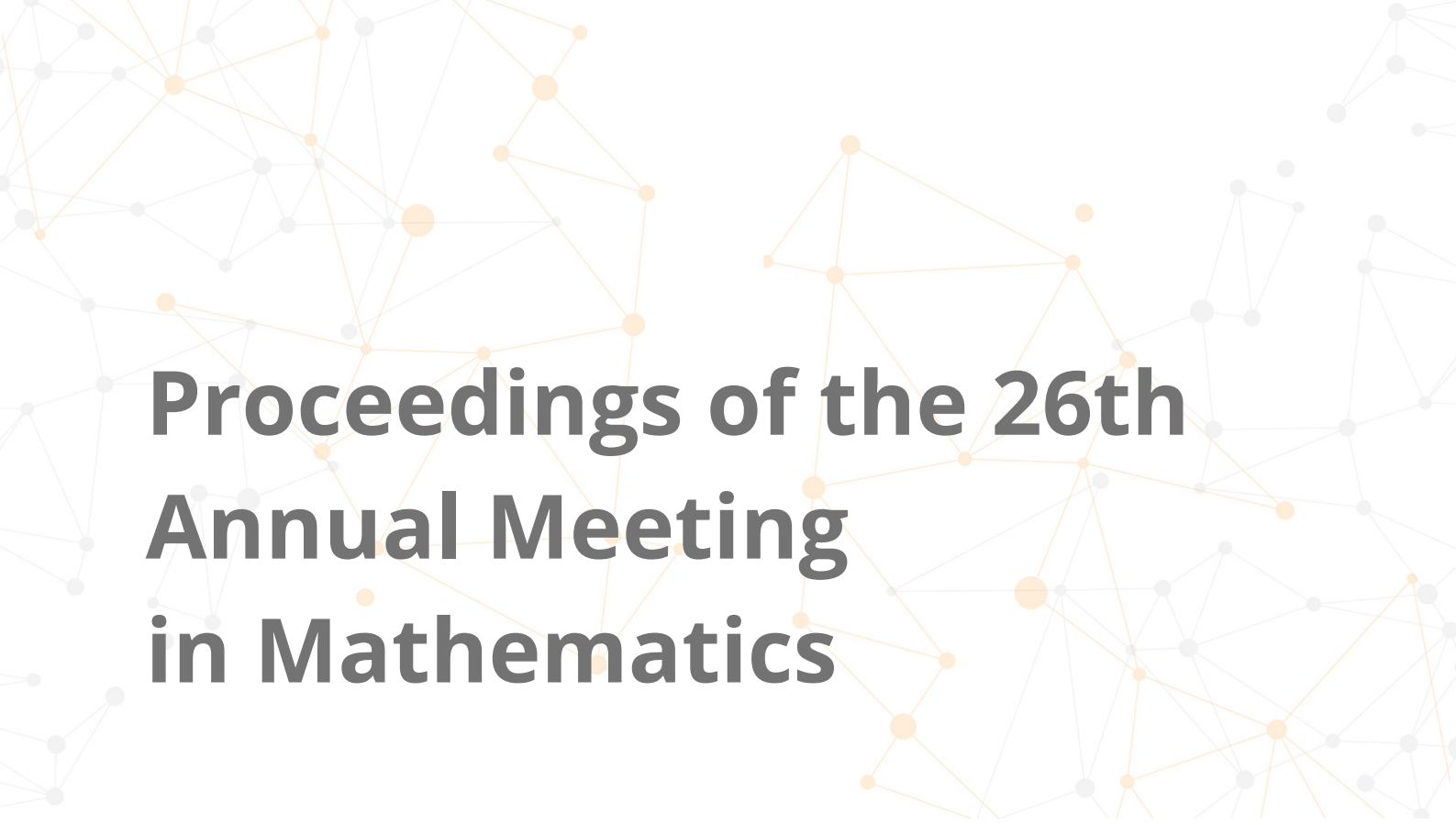
การประชุมวิชาการทางคณิตศาสตร์ ครั้งที่ 26 ประจำปี 2565



รายงาน
การประชุม

สารบัญ -- Table of Contents

รายนามผู้สนับสนุน -- List of Sponsors	1
สารจากอธิการบดี มหาวิทยาลัยเทคโนโลยีสุรนารี Message from the Rector, Suranaree University of Technology	2
สารจากนายกสมาคมคณิตศาสตร์แห่งประเทศไทย ในพระบรมราชูปถัมภ์ Message from the President of the Mathematical Association of Thailand Under Patronage of His Majesty the King	4
สารจากอธิการ มหาวิทยาลัยราชภัฏนครราชสีมา Message from the Rector of Nakhon Ratchasima Rajabhat University	5
สารจากผู้อำนวยการศูนย์ส่งเสริมการวิจัยคณิตศาสตร์แห่งประเทศไทย Message from the Director of the Center for Promotion of Mathematical Research of Thailand	6
สารนำจากคณะกรรมการจัดการประชุมฯ Message from the Conference Organizing Committee	7
สารบัญ -- Table of Contents	8
กำหนดการ -- Conference Schedule	9
Keynote Speakers	14
Invited Speakers	17
Part I: Proceedings of the 26th Annual Meeting in Mathematics	20
Part II: Proceedings of the 1st International Annual Meeting in Mathematics	417
คณะกรรมการจัดการประชุมฯ --- Conference Committees	A

A network diagram background consisting of interconnected nodes and lines. The nodes are represented by small circles in shades of orange and grey, connected by thin lines of the same colors. The overall structure is a complex, web-like pattern.

Proceedings of the 26th Annual Meeting in Mathematics



สารบัญ -- Table of Contents

01 Algebra and Number Theory (AN)

AN-N-01	Solutions to a Quadratic Equation over Finite Fields <i>Puchong Wongkumptra and Detchat Samart</i>	1
AN-N-08	The Weak First Exponential Law of Mixed Product of n -ary Hyperalgebras Carried by Good Homomorphisms <i>Nitima Phrommarat and Thanwarat Butsan</i>	12
AN-N-10	Iterative Algorithm for Polynomial Modular Inversion Modulo $x^{p^r} - 1$ Over Finite Field of order p <i>Samakorn Sripatthanakul and Wutichai Chongchitmate</i>	21

02 Geometry and Graph Theory (GG)

GG-N-01	Some Families of Self-Clique Graphs whose Clique Size Sequence is $(2, \dots, 2, 3, 3, 3)$ <i>Sajika Tubtim, Sirirat Singhun and Ratinan Boonklurb</i>	27
GG-N-02	Charges of Semistandard Young Tableaux of Certain Shapes and Contents <i>Nattanon Tualue and Ouamporn Phuksuwan</i>	36

04 Differential Equations and Dynamical Systems (DE)

DE-N-01	Algebraic Independence of Solutions of Certain Second Order Homogeneous Linear Differential Equations with Linear Coefficients <i>Phisitphong Ketrat and Vichian Laohakosol</i>	50
DE-N-03	Beyond Quenching Profile for Singular Semilinear Parabolic Problem With Mixed Boundary Conditions <i>Benjamin Thaitavorn and Ratinan Boonklurb</i>	62

05 Mathematical Modeling and Numerical Mathematics (MN)

MN-N-01	Numerical Simulation of Water Quality in a Couple of Ponds of Shrimp Farming <i>Nattinee Sittijinda and Nopparat Pochai</i>	70
---------	---	----

MN-N-02	A Non-Dimensional Mathematical Model of Shoreline Evolution with a Groin Structure <i>Surasak Manilam and Nopparat Pochai</i>	81
MN-N-03	การเลือกตำแหน่งที่ตั้งและจำนวนหัวขาร์จของรถยนต์ไฟฟ้าที่เหมาะสมในกรุงเทพมหานคร <i>ณิชากร ชัยบัญชากิจ, พศิกายุจน์ ทับประดง, ณัฐรุทธิ์ เหลืองสวัสดิ์พร, และ สายฝน จาตุรันตบุตร</i>	90
MN-N-05	แบบจำลองการแพร่กระจายผู้ติดเชื้อโควิด-19 ภายใต้ภาวะควบคุมการระบาดด้วยการฉีดวัคซีน <i>ธนาทร อินทรปัญญา, อภิชาติ ศุภธณี, ลีทิพย์ ภัทรดิลกรัตน และ กิติพร พลายมาศ</i>	115
MN-N-06	A Mathematical Model for Measuring Carbon Dioxide Concentration in a Bus Due to Passengers Breathing <i>Jenjira Sooknum and Nopparat Pochai</i>	132
MN-N-07	Numerical Algorithm Based on Finite Integration Method Using Shifted Chebyshev Expansion for Solving Moving Boundary Problems <i>Warunya Wong-u-ra and Ratinan Boonklurb</i>	141
MN-N-08	The Simplex Method with the Minimal Angle Jump for Solving Linear Programming Problems <i>Monsicha Tipawanna and Krung Sinapiromsaran</i>	153
MN-N-09	โจทย์ปัญหาทางคณิตศาสตร์ที่น่าสนใจและ การประยุกต์ใช้โปรแกรมภาษา C++ เพื่อช่วยใน การหาผลเฉลยของปัญหาทางคณิตศาสตร์ <i>อมรรัตน์ สุริยวิจิตรเศรษฐี และ เจษฎา ตัณฑนุช</i>	163
06 Probability Theory (PR)		
PR-N-01	การเปรียบเทียบการทดสอบค่ากลางของประชากรสองกลุ่มที่เป็นอิสระกันเมื่อตัวอย่างที่ขนาดเล็มาก <i>มนต์นภา พงษ์พรรณากุล และ วนิดา พงษ์ศักดิ์ชาติ</i>	176
PR-N-02	An Improvement of the Error Bound of Local Limit Theorems for Sums of Independent Lattice Random Variables <i>Punyapat Kammoo, Kritsana Neammanee and Kittipong Laipaporn</i>	187

PR-N-03	Poisson Approximation for Sums of Independent Non-Negative Integer-valued Random Variables <i>Supavit Kiatteerarat, Kritsana Neammanee and Suporn Jongpreechaharn</i>	197
PR-N-04	การเปรียบเทียบประสิทธิภาพของแผนภาพกล่องสำหรับการตรวจสอบค่า นอกเกณฑ์ <i>ทศวรรษ ฌ บางซาง และ บำรุงศักดิ์ เพื่อนอารีย์</i>	206
PR-N-05	A Local Limit Theorem for Negative Binomial Random Sums <i>Hattacha Kongjiw, Petcharat Rattanawong and Kritsana Neammanee</i>	218
07 Data Science and Statistics (DS)		
DS-N-01	โครงข่ายประสาทเทียมสำหรับการวิเคราะห์การถดถอยเชิงเส้นตามบริบท นัยทั่วไป <i>ชยานนท์ ชัตติยาภิรักษ์ และ เสกสรร เกียรติสุไพบูลย์</i>	228
DS-N-02	การทำนายราคาของหลักทรัพย์โดยวิศวกรรมคุณลักษณะและเทคนิคการ เรียนรู้ของเครื่อง <i>รัชพล ปรีโยทัย และ เบญจวรรณ โรจนดิษฐ์</i>	243
DS-N-04	โครงข่ายประสาทเทียมเชิงพยากรณ์แบบปรับปรุงโดยใช้การเลือกสับเซต ที่ดีที่สุด <i>พรชิตา ธนากร, สุปราณี ลิสวัสดิ์ และ พัทธ์ชนก ศรีสุรเดชชัย</i>	263
DS-N-05	การทำนายโรคเบาหวานโดยใช้วิศวกรรมคุณลักษณะสำหรับขั้นตอน วิธีการจำแนกในการเรียนรู้ของเครื่อง <i>คุณากรณ์ พันธุ์เพียร, จักรกฤษณ์ พลรบ และ เจษฎา ตัณทนุช</i>	278
DS-N-06	การเฉลี่ยตัวแบบบนต้นไม้การถดถอยสำหรับการพยากรณ์ <i>ชนินทร แก้ววิบูลย์พันธุ์, สุปราณี ลิสวัสดิ์ และ พัทธ์ชนก ศรีสุรเดชชัย</i>	290
DS-N-08	การวิเคราะห์พื้นที่ที่เหมาะสมสำหรับการปลูกยางพาราตามปัจจัยสภาพ ภูมิอากาศด้วยวิธีการตัดสินใจแบบหลายหลักเกณฑ์ <i>พรทิวชัย เดชพิชัย, กัญญาวิวี คำสนอนันตกุล, ธัญชนก ชัยกุล และ บิลกีส์ วงษ์พิริว</i>	316
DS-N-09	การวิเคราะห์ความรู้สึกที่มีต่อการทำประกันภัยด้วยเทคนิคการเรียนรู้เชิง ลึก กรณีศึกษากระทู้ออนไลน์พันทิป <i>คงภพ ไชยคร และ บุษยมาส พิมพ์พรรณชาติ</i>	321

DS-N-10	<p>การเปรียบเทียบตัวแบบ SARIMAX และตัวแบบแยกส่วนประกอบร่วมกับ SARIMAX ในการพยากรณ์ค่าสินไหมทดแทนของธุรกิจประกันภัยรถยนต์ในประเทศไทย</p> <p style="text-align: center;"><i>นิฉา แก้วหาวงษ์, ไอริน ลิมลีแก้ว, ณัฐกิตติ์ การเร็ว และ วรณกานต์ วงศ์เสนา</i></p>	332
DS-N-11	<p>การศึกษาและพัฒนาระบบการจัดการงานสินไหมในกรณีความเสียหายหนัก ของการประกันภัยรถยนต์</p> <p style="text-align: center;"><i>เบญจมาภรณ์ ศรีอัมพร, บุษยมาล พิมพ์พรรณชาติ และ เทิดขวัญ ช้างเผือก</i></p>	346
08 Mathematics Education (ED)		
ED-N-01	<p>การพัฒนาความเข้าใจทางคณิตศาสตร์ของนักเรียนโดยใช้การศึกษาชั้นเรียนและวิธีการแบบเปิด</p> <p style="text-align: center;"><i>กนกวรรณ รัตนจำนอง, วิภาพร สุทธิอัมพร และ สุริพร บุญเมือง</i></p>	358
ED-N-02	<p>การพัฒนาชุดการเรียนรู้ด้วยตนเองเรื่องเรขาคณิตสำหรับผู้เรียนในระดับมัธยมศึกษาตอนปลายโดยเปรียบเทียบมโนทัศน์เรขาคณิตแบบฉบับและเรขาคณิตวิเคราะห์</p> <p style="text-align: center;"><i>วุฒิชัย ไชยปัญญา และ ศุภณัฐ ชัยดี</i></p>	366
ED-N-03	<p>การคิดทางคณิตศาสตร์ของนักเรียนในชั้นเรียนที่จัดการเรียนรู้แบบผสมผสานด้วยวิธีการแบบเปิด</p> <p style="text-align: center;"><i>ธัญญารัตน์ ถ่องแท้, สุดาทิพย์ หาญเชิงชัย และศคลักษณ์ ขลิกก่ำ</i></p>	381



Binary Whale Optimization Algorithm for Improving Data Balance Based on Undersampling Techniques

Jakkrit Polrob[†], Benjawan Rodjanadid[‡], Jessada Tanthanuch and Eckart Schulz

School of Mathematics, Institute of Science
Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand

Abstract

The imbalance of data problem occurs frequently, causing poor prediction performance for minority classes. In addition, one of the most common problems with imbalanced data is the inaccuracy and bias of a predictive model using conventional machine learning algorithms. In this study, we propose a novel undersampling algorithm based on the idea of a binary whale optimization algorithm and K-nearest neighbor, called BWOA-KNN. The algorithm starts by splitting the dataset into two parts, the training set part and the testing set part. For the training set, the minority class stays the same, whereas the majority class is analyzed to extract the best representative subset by the proposed algorithm. Then, the decision tree technique is applied to the new training set for assessing performance. Ten datasets of varying imbalance ratio ranging from 1.82 to 42.01 from the KEEL and Imbalanced-learn repositories were used for evaluating the algorithm. The result found that overall BWOA-KNN performed better than three other undersampling algorithms, namely Random Undersampling, Cluster Centroid, and Near-Miss algorithms.

Keywords: Binary whale optimization algorithm, Imbalanced data, Undersampling technique, K-nearest neighbor, Decision Tree.

2020 MSC: Primary 68T09; Secondary 68U99.

1 Introduction

Resampling techniques represent methods that can rebalance the data. Once the data are balanced, one can use this new data set to create a machine learning model. It will result in better performance for correctly predicting minority class membership and reducing bias. Resampling techniques can be typically categorized into three groups: undersampling methods, oversampling methods, and hybrid methods [4]. Undersampling reduces the size of the majority classes to the corresponding size of the minority classes, even though one may lose some beneficial

[†]Speaker. [‡]Corresponding author.

Email: m6300289@g.sut.ac.th (J. Polrob), benjawan@sut.ac.th (B. Rodjanadid), jessada@g.sut.ac.th (J. Tanthanuch), eckart@sut.ac.th (E. Schulz).

information. However, in this way, besides a reduction in overfitting the processing time of the machine learning model is also reduced.

A number of nature-inspired algorithms have been applied to imbalanced data problems such as ant colony optimization algorithms [5], evolutionary algorithms [6], and genetic algorithms [7]. However, there is another algorithm that has become popular and has been applied to many fields of work, namely the binary whale optimization algorithm [8]. Examples of the efficient application of this algorithm are the feature selection problem [9–11] and the electrical engineer problem [8]. The binary whale optimization algorithm finds the solution with only binary vectors. This capability is admirably suited to applications that require a choice of doing or not. Therefore, it is an interesting algorithm if applied to solving the problem of imbalanced data because this algorithm can be used to select a subset from the majority class.

Therefore, in this work, we present a novel algorithm that uses the binary whale optimization algorithm combined with the K-nearest neighbors algorithm [12] for undersampling. We evaluate the performance of the proposed algorithm by comparing it with some of the most popular and widely used techniques, which are Random Undersampling [13], Cluster Centroid [14], and Near-Miss [15]. We use a Decision Tree [16] to assess the results and use several performance metrics to compare performance, such as Accuracy, Recall, G-mean, F-measure, Area Under the Receiver Operating Characteristic Curve (AUROC), and Area Under the Precision-Recall Curve (AUPRC). The research objective is to develop a novel technique for solving imbalanced data problems based on undersampling using the binary whale optimization algorithm.

2 Preliminaries

This section presents the knowledge of basic mathematics and machine learning related to imbalanced data problems. The contents consist of the main idea of the whale optimization algorithm, binary whale optimization algorithm, imbalance ratio, undersampling methods, the model for classification, performance metric, and standard competition ranking.

2.1 Whale Optimization Algorithm

The Whale Optimization Algorithm (WOA) [17] is a recently introduced metaheuristic algorithm proposed by Mirjalili and Lewis which mimics the foraging of humpback whales. The humpback whales hunt school of krill or small fishes close to the surface by swimming around them within a shrinking circle and creating distinctive bubbles along the circle or ‘9’-shaped path (see Figure 1).

Encircling prey and spiral bubble-net attacking method represent the first phase of the algorithm (exploitation phase). In the second phase they search randomly for new prey (exploration phase). The following subsections discuss the mathematical model of each phase in details.

2.1.1 Exploitation phase (encircling prey/bubble-net attacking method)

To hunt a prey, humpback whales first encircle it. Equations (2.1) and (2.2) can be used to mathematically model this behavior,

$$\vec{D} = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)|, \quad (2.1)$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D}, \quad (2.2)$$

where t indicates the current iteration, \vec{X}^* represents the best solution obtained so far, \vec{X} is the position vector, $|\cdot|$ is the elementwise absolute value, and \cdot is element-by-element multiplication. In addition, \vec{A} and \vec{C} are coefficient vectors that are calculated as in equations (2.3) and (2.4), respectively:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a}, \quad (2.3)$$

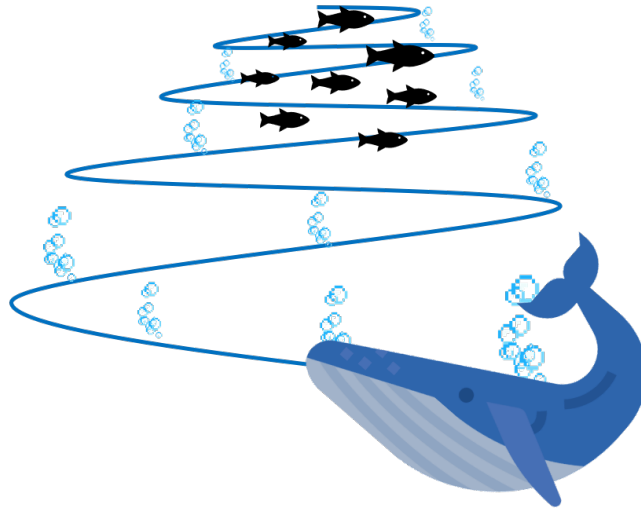


Figure 1: Bubble-net feeding of humpback whales.

$$\vec{C} = 2\vec{r}, \quad (2.4)$$

where $\vec{a} = a(1, 1, \dots, 1)$ and a decreases linearly from 2 to 0 over the course of iterations (in both exploration and exploitation phases) and \vec{r} is a random vector generated with uniform distribution in the interval of $[0,1]$.

According to equation (2.2) the search agents (whales) update their positions according to the position of the best known solution (prey). The adjustment of the values of \vec{A} and \vec{C} vectors control the areas where a whale can be located in the neighborhood of the prey.

The shrinking encircling behavior is achieved by decreasing the value of a in equation (2.3) according to equation (2.5) [18].

$$a = 2 - t \frac{2}{MaxIter}, \quad (2.5)$$

where t is the iteration number and $MaxIter$ is the maximum number of allowed iterations.

To simulate the spiral-shaped path, the distance between a search agent (\vec{X}) and the best known search agent so far (\vec{X}^*) is calculated, then a spiral equation is used to create the position of the neighbor search agent as in equation (2.7).

$$\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)|, \quad (2.6)$$

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), \quad (2.7)$$

where \vec{D}' indicates the distance of the i -th whale from the prey (best solution obtained so far), b is a constant for defining the shape of the logarithmic spiral, and l is a random number in $[-1,1]$.

To model the two mechanisms, shrinking encircling, and bubble-net attacking, a probability of 50% is assumed to choose between them during the optimization process as in equation (2.8).

$$\vec{X}(t+1) = \begin{cases} \text{Shrinking Encircling (equation (2.2))} & \text{if } p < 0.5 \\ \text{Bubble-net attacking (equation (2.7))} & \text{if } p \geq 0.5 \end{cases} \quad (2.8)$$

where p is a random number in $[0,1]$.

2.1.2 Exploration phase (search for prey)

In order to enhance the exploration in WOA, instead of updating the positions of the search agents according to the best position achieved so far, a random search agent is selected to guide

the search. So, a vector \vec{A} with the random coponents greater than 1 or less than -1 is used to force the search agent to move far away from the best known search agent. This mechanism can be mathematically modeled as in equations (2.9) and (2.10).

$$\vec{D}'' = |\vec{C} \cdot \vec{X}_{rand} - \vec{X}|, \tag{2.9}$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D}'', \tag{2.10}$$

where \vec{X}_{rand} is a random whale chosen from the current population.

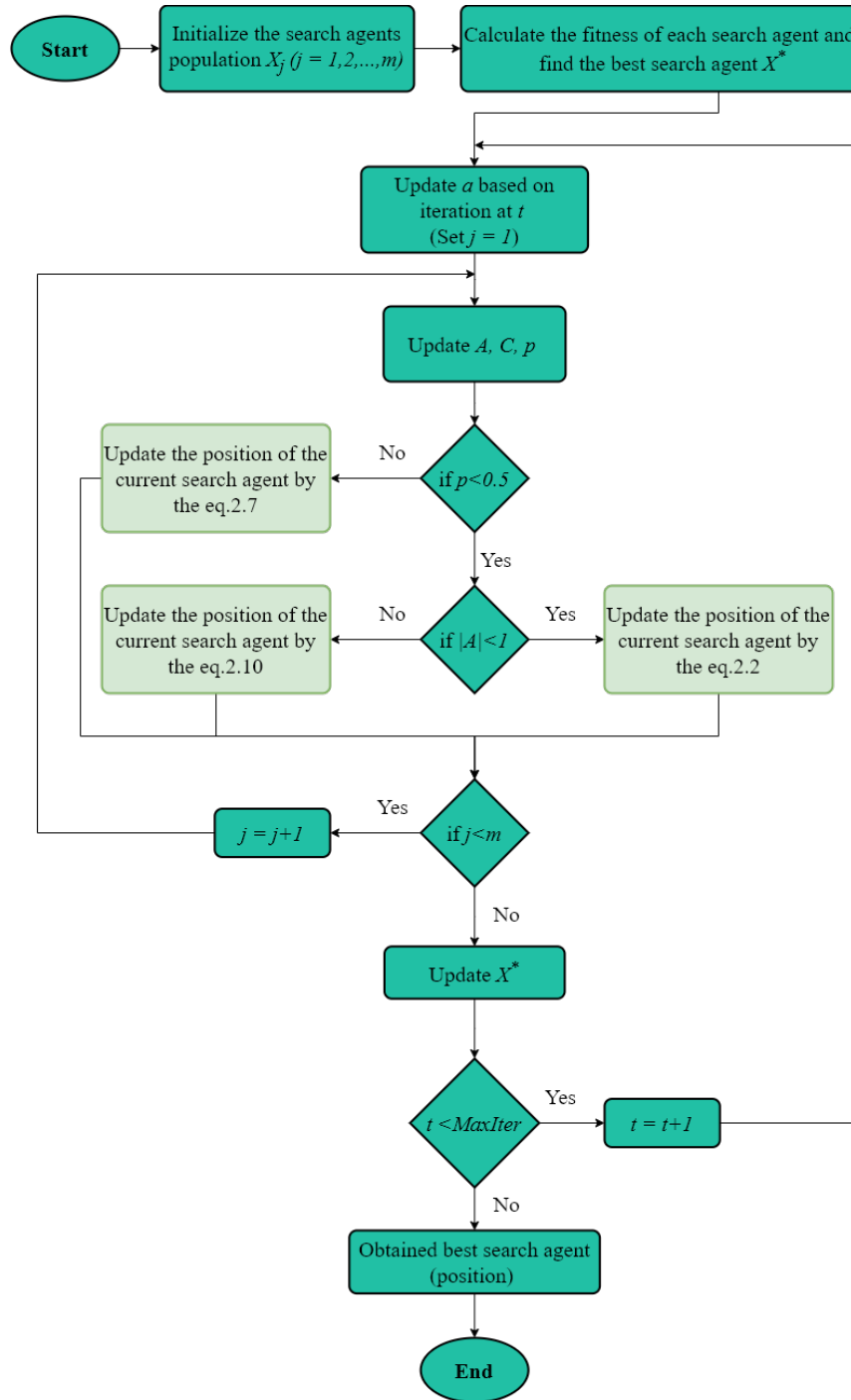


Figure 2: Flowchart of the whale optimization algorithm.

Figure 2 shows the workflow of the WOA algorithm. It may be seen that WOA creates

a random, initial population, and evaluates it using a fitness function once the optimization process starts. After finding the best solution, the algorithm repeatedly executes the following steps until is satisfied of an end criterion. Firstly, the main coefficients are updated. Secondly, a random value is generated. Based on this random value, the algorithm updates the position of a solution using either of equations (2.2),(2.10) or (2.7). Thirdly, the solutions are prevented from going outside the search landscape. Finally, the algorithm returns the best solution obtained as an approximation of the global optimum.

2.2 Binary Whale Optimization Algorithm

The Binary Whale Optimization Algorithm (BWOA) [8] was developed from the whale optimization algorithm (WOA) to be able to find solutions with only binary vectors, unlike WOA, which finds solutions as real vectors. The fact that the vector components are only 0 or 1 can be applied to many applications such as feature selection [9–11] or unit commitment [8]. The foremost difference between the original and binary versions of WOA is that of the position updating mechanism. In BWOA, the toggling between the values 0 and 1 indicates position updating. Furthermore, the value of the current bit is changed with a probability that is computed in accordance with the helix-shaped movement of the whale. To achieve this, an appropriate transfer function is required to map the helix-shaped movement values into probability values of position updating. The transfer function forces whales to travel in a binary space. Based on the above-mentioned concept, an appropriate probability of the function can be formulated as [8]:

$$\overrightarrow{Cstep} = \frac{1}{1 + e^{-10(\vec{A} \cdot \vec{D} - 0.5)}} , \quad (2.11)$$

where \overrightarrow{Cstep} is the step size that can be computed using sigmoidal function. \vec{D} is the distance between the position of prey and humpback whale calculated as in equation (2.1).

2.2.1 Exploitation phase (encircling prey/bubble-net attacking method)

First, the shrinking and encircling prey phase is modified. The position of whale is modified according to the equation mentioned below:

$$\vec{X}(t+1) = \begin{cases} \text{complement}(\vec{X}(t)), & \text{if } \overrightarrow{rand} < \overrightarrow{Cstep}; \\ \vec{X}(t), & \text{otherwise,} \end{cases} \quad (2.12)$$

where \overrightarrow{Cstep} is computed as is given by equation (2.11) and $\overrightarrow{rand} < \overrightarrow{Cstep}$ means an elementwise comparison.

The second modification is done in the bubble-net attacking behavior of whales that makes use of \overrightarrow{Cstep} computed as follows:

$$\overrightarrow{Cstep} = \frac{1}{1 + e^{-10(\vec{A} \cdot \vec{D}' - 0.5)}} , \quad (2.13)$$

where \vec{A} and \vec{D}' are computed using equation (2.3) and (2.6).

The position of the helix-shaped movement of humpback whales is updated according to bubble-attacking behavior. Then, the modification in the position updating process is done as follows:

$$\vec{X}(t+1) = \begin{cases} \text{complement}(\vec{X}(t)), & \text{if } \overrightarrow{rand} < \overrightarrow{Cstep}; \\ \vec{X}(t), & \text{otherwise,} \end{cases} \quad (2.14)$$

To model the two mechanisms, shrinking encircling, and the spiral-shaped path, a probability of 50% is assumed to choose between them during the optimization process as in equation (2.15).

$$\vec{X}(t+1) = \begin{cases} \text{Shrinking encircling (equation (2.12))} & \text{if } p < 0.5 \\ \text{Bubble-net attacking (equation (2.14))} & \text{if } p \geq 0.5 \end{cases} \quad (2.15)$$

where p is a random number in $[0,1]$.

2.2.2 Exploration phase (search for prey)

The third modification is done in the searching of prey, a random search agent is selected to guide the search. So, a vector \vec{A} with random entries greater than 1 or less than -1 is used to force the search agent to move far away from the best known search agent. The mathematical formulation of \overrightarrow{Cstep}'' is given below:

$$\overrightarrow{Cstep}'' = \frac{1}{1 + e^{-10(\vec{A} \cdot \vec{D}'' - 0.5)}}, \quad (2.16)$$

where \vec{D}'' is computed using equation (2.9). Hence, the position of whale is updated according to equation (2.17).

$$\vec{X}(t+1) = \begin{cases} \text{complement}(\vec{X}(t)), & \text{if } \overrightarrow{rand} < \overrightarrow{Cstep}''; \\ \vec{X}(t), & \text{otherwise,} \end{cases} \quad (2.17)$$

The workflow of the BWOA is similar to WOA but equations (2.10), (2.7), and (2.2) are replaced by equations (2.17), (2.14), and (2.12), respectively.

2.3 Imbalance Ratio

In two-class problems, the minority class is usually referred to as the positive class, whereas the majority class is considered to be the negative one. The conventional way of referring to the degree of imbalance of two-class problems is the Imbalance Ratio (IR) [19].

$$\text{Imbalance Ratio (IR)} = \frac{n^-}{n^+} \quad (2.18)$$

where n^- is the number of negative class samples and n^+ is the number of positive class samples. IR can be used to sort different datasets depending on their IR. One must take into account that the IR does not always give a good estimation of the difficulty of the dataset.

2.4 Undersampling Methods

Undersampling is an efficient method for balancing data. This method uses a subset of the majority class to train the classifier [20]. We will compare our new proposed undersampling method with three commonly used techniques: Random Undersampling, Cluster Centroid, and Near-Miss. Many of these methods use the concept of distance between two vectors in Euclidean space \mathbb{R}^d . Common choices are the Minkowski distances

$$\text{dist}(\vec{x}, \vec{y}) = \left[\sum_{i=1}^d |x_i - y_i|^p \right]^{1/p} \quad (1 \leq p < \infty)$$

for $\vec{x} = (x_1, x_2, x_3, \dots, x_d)$. We will use the Euclidean distance ($p = 2$) which can be expressed in the Euclidean norm $\|\vec{x} - \vec{y}\| = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$

2.4.1 Random Undersampling

Random undersampling (RUS) randomly selects samples from the majority class and deletes them from the training dataset. This method keeps the information of the minority class but reduces the size of the majority class, until class balance. However, if vast quantities of data are discarded, this can be highly problematic, as the loss of such data can make the decision boundary between minority and majority instances harder to learn, resulting in a loss in classification performance [21].

2.4.2 Cluster Centroid

The cluster centroid algorithm separates the majority class into K clusters, and replaces the majority class with the centroids of these clusters [14].

It starts with a first group of K randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative calculations to optimize the positions of the centroids. Given a set of observations $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathfrak{R}^d$, the K-means algorithm aims to cluster the n observations into K ($\leq n$) sets $\mathbf{S} = \{S_1, S_2, S_3, \dots, S_K\}$ in which each cluster has a centroid. An objective function for this clustering can be created by finding the minimum value of the sum of the squared distances of the samples from the centroids \mathbf{c}_k of their cluster follows:

$$J(\mathbf{r}, \mathbf{c}) = \sum_{i=1}^n \sum_{k=1}^K r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2,$$

where $r_{ik} \in \{0, 1\}$ is a variable that indicates the membership of the i -th sample in the k -th cluster. That is,

$$r_{ik} = \begin{cases} 1, & \text{if } k = \arg \min_j \|\mathbf{x}_i - \mathbf{c}_j\|, \\ 0, & \text{if } k \neq \arg \min_j \|\mathbf{x}_i - \mathbf{c}_j\|. \end{cases}$$

This means that $\sum_{k=1}^K r_{ik} = 1$ for each sample \mathbf{x}_i .

The centroid of each cluster during an iteration can be computed by

$$\mathbf{c}_k = \frac{\sum_{i=1}^n r_{ik} \mathbf{x}_i}{\sum_{i=1}^n r_{ik}}.$$

It can be seen that the divisor or $\sum_{i=1}^n r_{ik}$ is the total number of samples assigned to the k -th cluster, and \mathbf{c}_k is the mean of all samples assigned to this cluster.

2.4.3 Near-Miss

Near-miss is another efficient way to balance the data by undersampling [22]. It has three different versions [15]. In this study, we will use the first version. Its working principle is to find, for each sample from the majority class, the three closest samples from the minority class and compute the average distance from these three. Then sort the elements of the majority class by this average distance, and choose the N samples of smallest average distance as the new majority class. Here, N is the size of the minority class. In this way, majority samples that are located far from the minority class are removed.

2.5 Model for Classification

There are many techniques for classification available, including support vector machines, neural networks, K-nearest neighbors, and decision tree. We will be using the latter two.

2.5.1 K-nearest neighbors

The K-nearest neighbors (KNN) is a supervised learning algorithm. It is known for its simplicity and effectiveness [23].

To classify a data sample, predictions are made by searching the entire training set for the closest K neighbors and tabulating the output variable for those K cases. The factors that affect the performance of KNN are the value of K , the distance, and the normalization of the data. To understand the detailed working of the algorithm, the steps are as follows:

Given the training dataset: $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}, \dots, \mathbf{x}^{(m)}$ where $\mathbf{x}^{(i)} = (x_1, x_2, x_3, \dots, x_n)$, m is the number of training data, n is the number of features of each training data.

Step1: Store the training set

Step2: For each new yet unlabeled data sample $\mathbf{y} = (y_1, y_2, y_3, \dots, y_n)$:

A. Calculate distance from all training data points.

$$\text{dist}(\mathbf{x}^{(j)}, \mathbf{y}) = \|\mathbf{x}^{(j)} - \mathbf{y}\|$$

B. Find the K nearest neighbors of \mathbf{y} and sort them by class.

C. Assign the class with the largest number of nearest neighbors to \mathbf{y} .

2.5.2 Decision Tree

Decision Tree is supervised learning suitable for solving regression and classification problems. It is known to give good accuracy in classification. In 1984, a group of statisticians published the book Classification and Regression Trees (CART) [16], which described how a binary decision trees work. It can produce either classification or regression trees, depending on whether the dependent variable is a number or category, respectively. Since a decision tree can handle noisy data and many independent variables using simple If-Else rules, a decision tree is easy to interpret.

The components of a decision tree are nodes and branches. A branch represents the outcome of the node or the values of the attributes. The node on the top is called the root node, there is only one such root node, and there is a unique path from the root node to any other node. The remaining nodes are called the internal nodes, except for the leaf nodes, which represent the classes or the output of the model [24] shown in Figure 3.

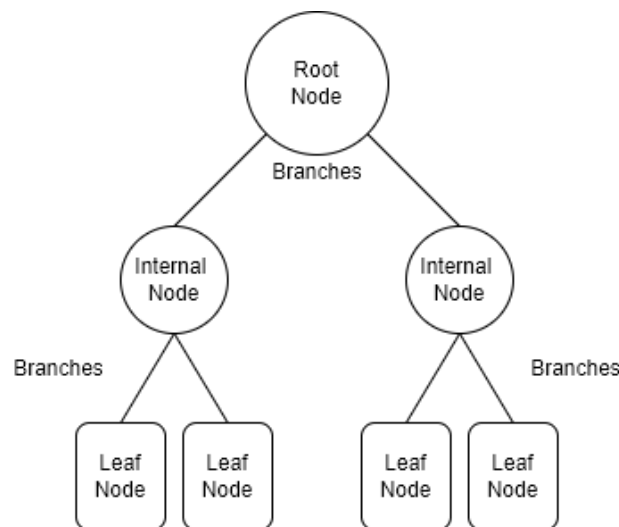


Figure 3: The components of a decision tree.

From all of the above, it can be seen that there are different versions of decision trees and each form will also use various splitting criteria. There are many measures of splitting that can be used to decide the best way to split the node. Common splitting criteria for the decision tree are as follows: gini index, twoing criteria, entropy, information gain, and gain ratio [25]. However, in this study, we are interested in the gini index as the only splitting criterion. It is calculated using the following formula:

$$\text{Gini Index} = 1 - \sum_j p_j^2, \quad (2.19)$$

where p_j is the probability of class j . The gini index measures the frequency at which any element of the dataset will be mislabeled when it is randomly labeled.

2.6 Performance Metrics

In this section, the performance measurement of binary classification models is discussed which will be used later on.

2.6.1 Confusion Matrix

A confusion matrix is a table that visualizes the performance of a classification algorithm. A confusion matrix is shown in Figure 4.

		Prediction	
		Negative (0)	Positive (1)
Actual	Negative (0)	True Negative (TN)	False Positive (FP)
	Positive (1)	False Negative (FN)	True Positive (TP)

Figure 4: Confusion matrix.

The entries in the confusion matrix are defined as the following:

1. True positive (TP) is the number of elements in the positive class that are correctly predicted as positive.
2. True negative (TN) is the number of elements in the negative class that are correctly predicted as negative.
3. False positive (FP) is the number of elements in the negative class that are wrongly predicted as positive.
4. False negative (FN) is the number of elements in the positive class that are wrongly predicted as negative.

Performance metrics used here are accuracy, precision, sensitivity, specificity, F1 score [4], and G-mean [26], which are calculated on the basis of the above-stated TP , TN , FP , and FN .

$$\begin{aligned} \text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN} \\ \text{Precision} &= \frac{TP}{TP + FP} \\ \text{Sensitivity or Recall} &= \frac{TP}{TP + FN} \\ \text{Specificity} &= \frac{TN}{TN + FP} \\ \text{F1 score} &= \frac{2 \times (\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \\ \text{G-mean} &= \sqrt{\text{Sensitivity} \times \text{Specificity}} \end{aligned}$$

2.6.2 Receiver operating characteristic curve (ROC Curve)

A ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. It is a plot of False Positive Rate (FPR) on the x -axis, and True Positive Rate (TPR) on the y -axis shown in Figure 5 [27]. The True Positive Rate is identical to sensitivity/recall while the False Positive Rate (FPR) is

$$\text{FPR} = 1 - \text{Specificity} = \frac{FP}{TN + FP}$$

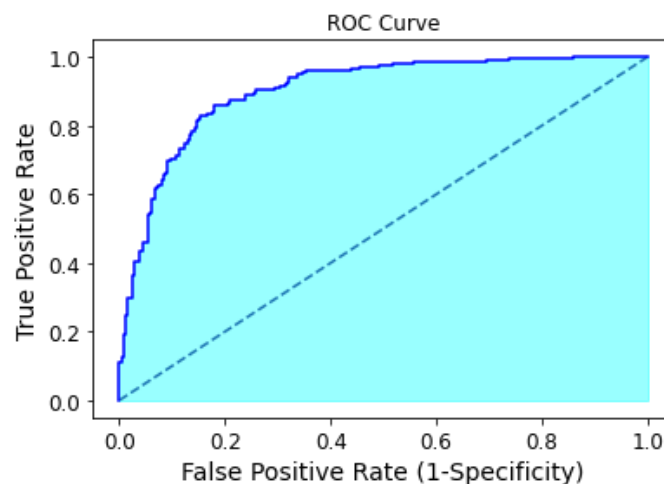


Figure 5: ROC Curve.

A good classifier should reach as close to the top left corner as possible. The value for Area Under the ROC Curve (AUROC) ranges from 0 to 1. A model that has an AUROC of 1 is able to perfectly classify observations into classes while a model that has an AUROC of 0.5 does no better than a model that performs random guessing.

2.6.3 Precision-Recall curve (PR Curve)

The precision-recall curve shows the tradeoff between precision and recall for different thresholds. The process of drawing the PR Curve is similar to ROC Curve but uses the x -axis for recall, and the y -axis for precision as shown in Figure 6. PR curves are often used in information retrieval, and focus only on the positive class [28].

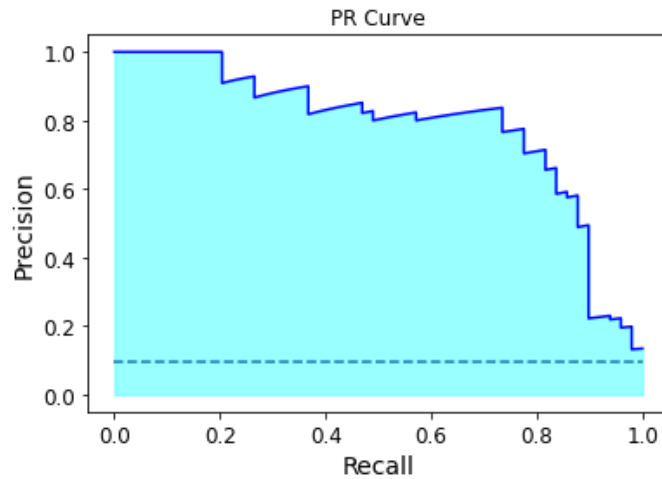


Figure 6: PR Curve.

A good classifier should be as close as possible to the top right, as this corner represents the best trade-off between precision and recall. The value for Area Under the PR Curve (AUPRC) ranges from 0 to 1 [29].

2.7 K-fold Cross-Validation

K-fold cross-validation is a very popular technique for machine learning models. The workflow divides the training data into k partitions (or folds), then uses $k - 1$ of the partitions for training and the k -th for testing. After that the procedure is repeated for another $k - 1$ times, rotating the test partition. The performance results are reported on the results across all k iterations.

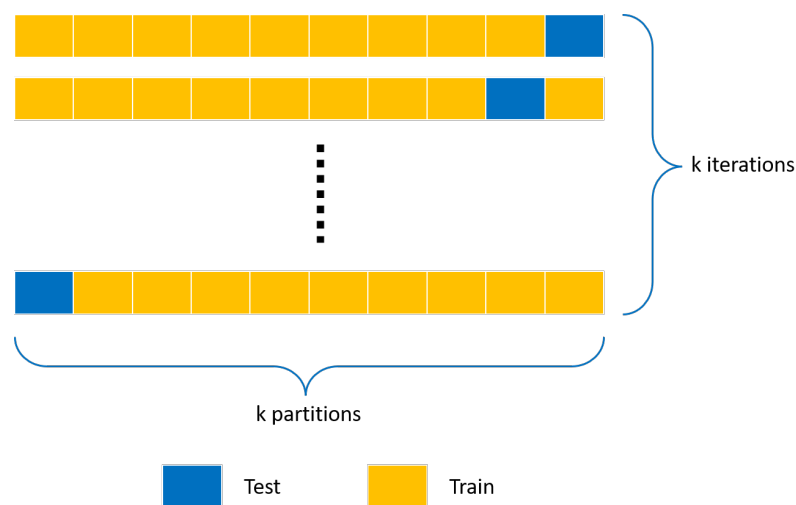


Figure 7: K-fold cross-validation.

2.8 Standard competition ranking (“1224” ranking)

In competition ranking, items that compare as equal receive the same ranking number, and then a gap is left in the ranking numbers [30]. This is called standard competition ranking and is used here to rank the performance of the various undersampling algorithms on individual datasets.

Then the average rank R_j of the j -th undersampling method is computed by [31]

$$R_j = \frac{1}{m} \sum_{i=1}^m r_{ij}, \quad (2.20)$$

where m is the total number of datasets and r_{ij} is the rank of the j -th undersampling method on the i -th dataset.

3 The Proposed Undersampling Algorithm

We now describe the proposed BWOA-KNN algorithm in mathematical terms.

Let D be a given dataset. Split D into the majority class D^- and the minority class D^+ , and let d and n^+ denote the number of samples in each class: $d = |D^-|$ and $n^+ = |D^+|$. When the data is highly unbalanced, then $d \gg n^+$.

The objective of our undersampling algorithm is to find a subset D_{red}^- of the majority class D^- with $|D_{red}^-| \approx |D^+|$ while at the same time giving best performance for a chosen classifier, when $D_{red}^- \cup D^+$ is the training data.

The performance metric which we choose is of the form

$$f = f(A) := (1 - \text{F1 score})^2 + (1 - \text{AUROC})^2 + (1 - \text{sensitivity})^2 + \beta(n^- - n^+)^2 \quad (3.1)$$

where $A \subseteq D^-$ is a given subset of the majority class, $n^- = n^-(A) = |A|$ is the number of samples in A , β is a parameter, and F1score, sensitivity and AUROC are obtained through 10-fold cross-validation of the chosen classifier using the dataset $A \cup D^+$. The parameter β influences how well the two datasets should be balanced. In this manner we obtain a function

$$f : 2^{D^-} \rightarrow [0, \infty)$$

defined on the power set of D^- which is to be minimized.

Observe that after fixing a labeling of the samples in the majority class, $D^- = \{\mathbf{x}_1, \dots, \mathbf{x}_d\}$, then there is a natural bijection

$$\Phi : \mathcal{X} := \{0, 1\}^d \rightarrow 2^{D^-}$$

given by

$$\Phi(\vec{X}) = \{\mathbf{x}_i \in D^- : \vec{X}(i) = 1\}.$$

That is, every binary vector \vec{X} of length d uniquely determines a subset of the majority class according to the vector components which are equal to one. Composition thus gives a function

$$f \circ \Phi : \mathcal{X} \rightarrow [0, \infty)$$

to be minimized. Since the domain of this function is a space of binary vectors, the binary whale optimization algorithm is a natural candidate for finding a minimizer of $f \circ \Phi$ as fitness function, in particular, since this algorithm has shown to be fairly efficient in applications. Furthermore, in order to keep computation time low, we choose the K -nearest-neighbors method with $K = 1$ as a simple classifier.

4 Results

We have selected 10 datasets from KEEL [32] and Imbalanced-learn [33] that represent a variety of imbalance ratios, as detailed in Table 1, in order to compare our proposed undersampling method with the random undersampling, cluster centroid and near-miss methods.

All tests proceeded as follows:

1. First split the given data set into the training and testing datasets at a ratio of 80 : 20.
2. Next split the training data set further into minority class D^+ and majority class D^- .
3. Obtain a reduced majority class D_{red}^- using any of the four undersampling methods, while the minority class remains D^+ .

In case of the binary whale optimization algorithm, we find a minimizing binary vector \vec{X}^* of $f \circ \Phi$ and the new reduced majority class is then $D_{red}^- = \{\mathbf{x}_i \in D^- : \vec{X}^*(i) = 1\}$. We have used 20 whales (search agents) and 1000 iterations. We also have chosen $\beta = 100$ to obtain balanced datasets.

4. Train a decision tree model using 10-fold validation for parameter optimization, and using the F1-score as performance metric.

Table 1: Detail of datasets.

Dataset name	Attributes	Size	Minority size	Majority size	IR
glass	9	214	76	138	1.82
iris0	4	150	50	100	2.00
glass-0-1-2-3_vs_4-5-6	9	214	51	163	3.20
ecoli2	7	336	52	284	5.46
ecoli	7	336	35	301	8.60
abalone	10	4177	391	3786	9.68
libras_move	90	360	24	336	14.00
solar_flare_m0	32	1389	68	1321	19.43
yeast_m2	8	1484	51	1433	28.10
mammography	6	11183	260	10923	42.01

An outline of the workflow is shown in Figure 8. The results are displayed in Table 2. There are 6 performance metrics listed: accuracy, F1-score, G-mean, Area Under the ROC curve (AUROC), Area Under the PR Curve (AUPRC) and sensitivity.

The results of the ranking score are shown in Table 3. This ranking score is based on standard competition ranking. The average rank of each resampling method is shown in Table 4.

5 Conclusion

The research result found from the average ranking score that the BWOA-KNN undersampling method, under the performance metrics F1-score, G-mean, AUPRC, and sensitivity, has the best (lowest) ranking, which means that the proposed algorithm has the best performance. Accuracy is usually high in imbalanced datasets, which causes the accuracy to be an inappropriate metric. It is therefore not surprising accuracy is decreased after undersampling. The AUROC metrics of random undersampling and the proposed algorithm are not much different. Therefore, the proposed algorithm which combines the binary whale optimization algorithm and K-nearest neighbor to an undersampling technique (BWOA-KNN) obtains excellent performance when compared with random undersampling, cluster centroid, and near-miss algorithms.

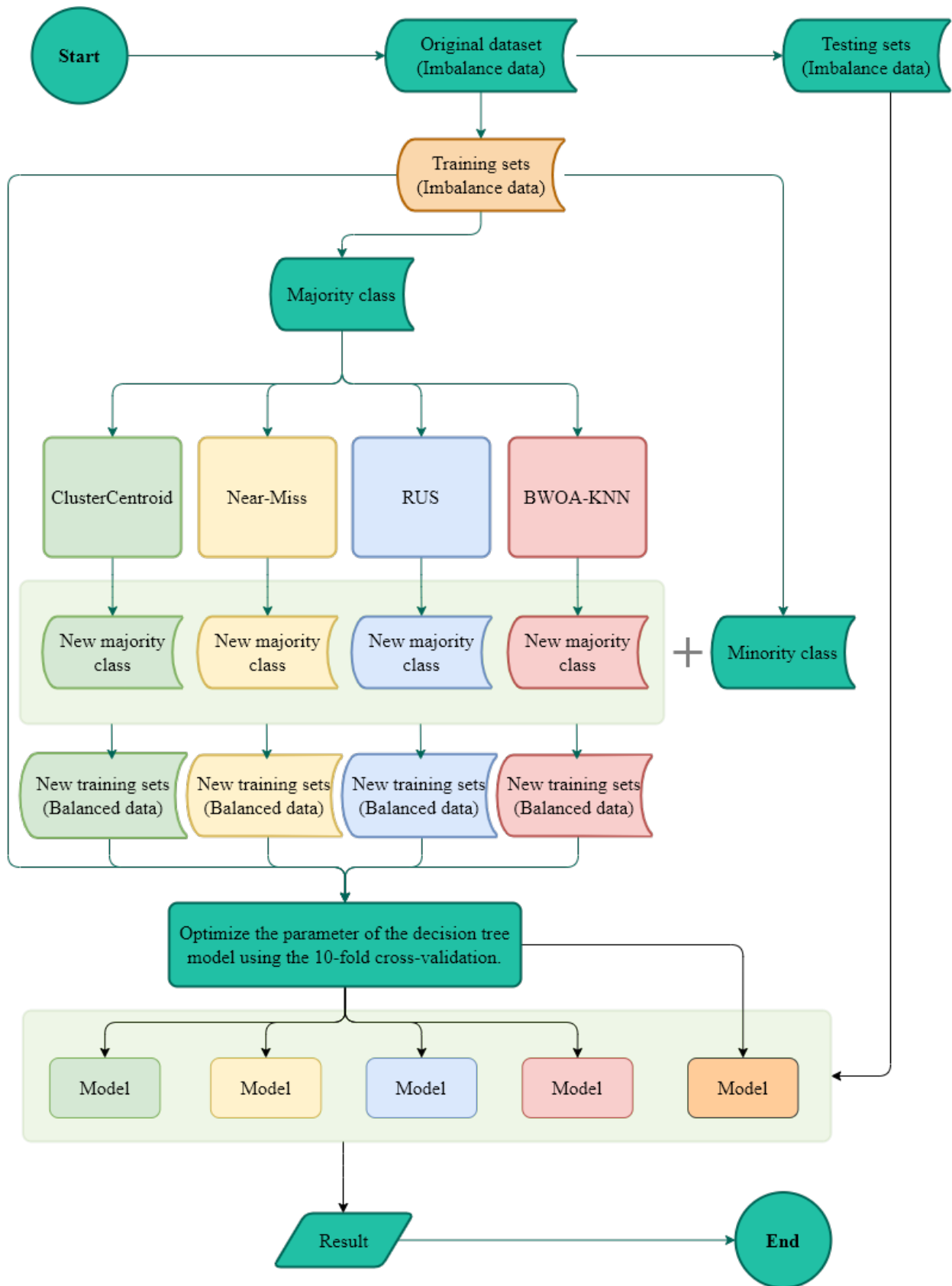


Figure 8: Outline of the workflow.

Table 2: The results of the various performance metrics by the decision tree model.

Dataset name	Measurement	Original	ClusterCentroid	Near-Miss	RUS	BWOA-KNN
glass	Accuracy	0.7674	0.7209	0.5349	0.7209	0.8372
	F1 score	0.6154	0.6000	0.5652	0.6842	0.7407
	G-mean	0.6901	0.6866	0.5563	0.7464	0.7868
	AUROC	0.7988	0.7452	0.6357	0.8024	0.7976
	AUPRC	0.7398	0.6820	0.6292	0.7671	0.8081
iris0	Sensitivity	0.5333	0.6000	0.8667	0.8667	0.6667
	Accuracy	1.0000	1.0000	1.0000	1.0000	1.0000
	F1 score	1.0000	1.0000	1.0000	1.0000	1.0000
	G-mean	1.0000	1.0000	1.0000	1.0000	1.0000
	AUROC	1.0000	1.0000	1.0000	1.0000	1.0000
glass-0-1-2-3_vs_4-5-6	AUPRC	1.0000	1.0000	1.0000	1.0000	1.0000
	Sensitivity	1.0000	1.0000	1.0000	1.0000	1.0000
	Accuracy	0.8372	0.8372	0.8372	0.8605	0.8372
	F1 score	0.6667	0.6667	0.6957	0.7273	0.6957
	G-mean	0.7843	0.7843	0.8239	0.8385	0.8239
ecoli2	AUROC	0.8288	0.8288	0.8242	0.8333	0.8242
	AUPRC	0.7201	0.7201	0.7309	0.7278	0.7309
	Sensitivity	0.7000	0.7000	0.8000	0.8000	0.8000
	Accuracy	0.9118	0.9412	0.4559	0.7647	0.9265
	F1 score	0.7273	0.8462	0.3509	0.5789	0.8148
ecoli	G-mean	0.8301	0.9643	0.5787	0.8481	0.9551
	AUROC	0.9450	0.9737	0.6372	0.8596	0.9561
	AUPRC	0.7843	0.8929	0.5702	0.7037	0.8438
	Sensitivity	0.7273	1.0000	0.9091	1.0000	1.0000
	Accuracy	0.9118	0.7941	0.9265	0.8824	0.8235
abalone	F1 score	0.5714	0.5000	0.6667	0.6000	0.5385
	G-mean	0.7371	0.8778	0.8241	0.8711	0.8963
	AUROC	0.9262	0.8852	0.8080	0.9520	0.9016
	AUPRC	0.3576	0.6667	0.7551	0.7690	0.6842
	Sensitivity	0.5714	1.0000	0.7143	0.8571	1.0000
libras_move	Accuracy	0.8565	0.6687	0.1555	0.7057	0.7022
	F1 score	0.2308	0.3420	0.1018	0.3594	0.3532
	G-mean	0.4610	0.7701	0.2468	0.7798	0.7726
	AUROC	0.5771	0.7828	0.2189	0.8573	0.8241
	AUPRC	0.2618	0.5701	0.0840	0.5757	0.5483
solar_flare_m0	Sensitivity	0.2308	0.9231	0.5128	0.8846	0.8718
	Accuracy	0.9583	0.5972	0.7500	0.8056	0.7778
	F1 score	0.7273	0.1714	0.3077	0.3000	0.3333
	G-mean	0.8810	0.5985	0.7727	0.7018	0.7880
	AUROC	0.8851	0.5985	0.7731	0.6463	0.7881
yeast_me2	AUPRC	0.7403	0.3639	0.5022	0.3993	0.5122
	Sensitivity	0.8000	0.6000	0.8000	0.6000	0.8000
	Accuracy	0.9353	0.4065	0.3381	0.7806	0.6115
	F1 score	0.1000	0.1270	0.1068	0.2078	0.1563
	G-mean	0.2647	0.5726	0.4970	0.6726	0.6580
mammography	AUROC	0.5764	0.6199	0.6640	0.7055	0.6692
	AUPRC	0.1129	0.4665	0.3483	0.3229	0.4084
	Sensitivity	0.0714	0.8571	0.7857	0.5714	0.7143
	Accuracy	0.9293	0.6734	0.6869	0.7037	0.6835
	F1 score	0.1600	0.1709	0.1622	0.1698	0.1754
mammography	G-mean	0.4370	0.8136	0.7820	0.7919	0.8200
	AUROC	0.5774	0.8310	0.7645	0.7984	0.9206
	AUPRC	0.1801	0.5467	0.2534	0.4986	0.3914
	Sensitivity	0.2000	1.0000	0.9000	0.9000	1.0000
	Accuracy	0.9844	0.4390	0.3317	0.9097	0.9137
mammography	F1 score	0.6237	0.0752	0.0639	0.3176	0.3322
	G-mean	0.7447	0.6464	0.5569	0.9068	0.9183
	AUROC	0.8999	0.9074	0.6485	0.9494	0.9525
	AUPRC	0.6441	0.3890	0.5071	0.5554	0.4796
	Sensitivity	0.5577	0.9808	0.9808	0.9038	0.9231

Table 3: The results of the ranking score by the decision tree model.

Dataset name	Measurement	Original	ClusterCentroid	Near-Miss	RUS	BWOA-KNN
glass	Accuracy	2	3	5	3	1
	F1 score	3	4	5	2	1
	G-mean	3	4	5	2	1
	AUROC	2	4	5	1	3
	AUPRC	3	4	5	2	1
	Sensitivity	5	4	1	1	3
iris0	Accuracy	1	1	1	1	1
	F1 score	1	1	1	1	1
	G-mean	1	1	1	1	1
	AUROC	1	1	1	1	1
	AUPRC	1	1	1	1	1
	Sensitivity	1	1	1	1	1
glass-0-1-2-3_vs_4-5-6	Accuracy	2	2	2	1	2
	F1 score	4	4	2	1	2
	G-mean	4	4	2	1	2
	AUROC	2	2	4	1	4
	AUPRC	4	4	1	3	1
	Sensitivity	4	4	1	1	1
ecoli2	Accuracy	3	1	5	4	2
	F1 score	3	1	5	4	2
	G-mean	4	1	5	3	2
	AUROC	3	1	5	4	2
	AUPRC	3	1	5	4	2
	Sensitivity	5	1	4	1	1
ecoli	Accuracy	2	5	1	3	4
	F1 score	3	5	1	2	4
	G-mean	5	2	4	3	1
	AUROC	2	4	5	1	3
	AUPRC	5	4	2	1	3
	Sensitivity	5	1	4	3	1
abalone	Accuracy	1	4	5	2	3
	F1 score	4	3	5	1	2
	G-mean	4	3	5	1	2
	AUROC	4	3	5	1	2
	AUPRC	4	2	5	1	3
	Sensitivity	5	1	4	2	3
libras_move	Accuracy	1	5	4	2	3
	F1 score	1	5	3	4	2
	G-mean	1	5	3	4	2
	AUROC	1	5	3	4	2
	AUPRC	1	5	3	4	2
	Sensitivity	1	4	1	4	1
solar_flare_m0	Accuracy	1	4	5	2	3
	F1 score	5	3	4	1	2
	G-mean	5	3	4	1	2
	AUROC	5	4	3	1	2
	AUPRC	5	1	3	4	2
	Sensitivity	5	1	2	4	3
yeast_me2	Accuracy	1	5	3	2	4
	F1 score	5	2	4	3	1
	G-mean	5	2	4	3	1
	AUROC	5	2	4	3	1
	AUPRC	5	1	4	2	3
	Sensitivity	5	1	3	3	1
mammography	Accuracy	1	4	5	3	2
	F1 score	1	4	5	3	2
	G-mean	3	4	5	2	1
	AUROC	4	3	5	2	1
	AUPRC	1	5	3	2	4
	Sensitivity	5	1	1	4	3

Table 4: Average ranking score of each algorithm.

Measurement	Original	ClusterCentroid	Near-Miss	RUS	BWOA-KNN
Accuracy	1.5	3.4	3.6	2.3	2.5
F1 score	3.0	3.2	3.5	2.2	1.9
G-mean	3.5	2.9	3.8	2.1	1.5
AUROC	2.9	2.9	4.0	1.9	2.1
AUPRC	3.2	2.8	3.2	2.4	2.2
Sensitivity	4.1	1.9	2.2	2.4	1.8

Acknowledgment. The authors thank all experts who have provided their opinions and suggestions to improve this research. J. Polrob gratefully acknowledges receipt of the Kittibandit scholarship from Suranaree University of Technology.

References

- [1] Fotouhi, S., Asadi, S., and Kattan, M. W. (2019). A comprehensive data level analysis for cancer diagnosis on imbalanced data. *Journal of Biomedical Informatics*, 90, 103089.
- [2] Mqadi, N. M., Naicker, N., and Adeliyi, T. (2021). Solving Misclassification of the Credit Card Imbalance Problem Using Near Miss. *Mathematical Problems in Engineering*, 2021.
- [3] Kesornsit, W., Lorchirachonkul, V., and Jitthavech, J. (2018). Imbalanced data problem solving in classification of diabetes patients. *Khon Kaen University Research Journal*, 18(3), 11-21.
- [4] Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., and Herrera, F. (2018). *Learning from Imbalanced Data Sets* (Vol. 10): Springer.
- [5] Yu, H., Ni, J., and Zhao, J. (2013). ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data. *Neurocomputing*, 101, 309-318.
- [6] López, V., Triguero, I., Carmona, C. J., García, S., and Herrera, F. (2014). Addressing imbalanced classification with instance generation techniques: IPADE-ID. *Neurocomputing*, 126, 15-28.
- [7] Kim, H.-J., Jo, N.-O., and Shin, K.-S. (2016). Optimization of cluster-based evolutionary undersampling for the artificial neural networks in corporate bankruptcy prediction. *Expert Systems with Applications*, 59, 226-234.
- [8] Kumar, V., and Kumar, D. (2020). Binary whale optimization algorithm and its application to unit commitment problem. *Neural Computing and Applications*, 32(7), 2095-2123.
- [9] Hussien, A. G., Hassanien, A. E., Houssein, E. H., Bhattacharyya, S., and Amin, M. (2019). S-shaped binary whale optimization algorithm for feature selection. In *Recent Trends in Signal and Image Processing* (pp. 79-87): Springer.
- [10] Sayed, G. I., Darwish, A., and Hassanien, A. E. (2020). Binary whale optimization algorithm and binary moth flame optimization with clustering algorithms for clinical breast cancer diagnoses. *Journal of Classification*, 37(1), 66-96.
- [11] Hussien, A. G., Hassanien, A. E., Houssein, E. H., Amin, M., and Azar, A. T. (2020). New binary whale optimization algorithm for discrete optimization problems. *Engineering Optimization*, 52(6), 945-959.
- [12] Aha, D. W., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6(1), 37-66.

- [13] Mishra, S. (2017). Handling imbalanced data: SMOTE vs. random undersampling. *International Journal of Managing Information Technology*, 4(8), 317-320.
- [14] imbalancedlearn. (2022). ClusterCentroids. Retrieved from https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.ClusterCentroids.html
- [15] Mani, I., and Zhang, I. (2003). *kNN approach to unbalanced data distributions: a case study involving information extraction*. Paper presented at the Proceedings of workshop on learning from imbalanced datasets.
- [16] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (2017). *Classification and regression trees*. Routledge.
- [17] Mirjalili, S., and Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95, 51-67.
- [18] Mafarja, M. M., and Mirjalili, S. (2017). Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing*, 260, 302-312.
- [19] Orriols-Puig, A., and Bernadó-Mansilla, E. (2009). Evolutionary rule-based systems for imbalanced data sets. *Soft Computing*, 13(3), 213-225.
- [20] Sonak, A., and Patankar, R. (2015). A survey on methods to handle imbalance dataset. *International Journal of Computer Science and Mobile Computing*, 4(11), 338-343.
- [21] He, H., and Ma, Y. (2013). *Imbalanced learning: foundations, algorithms, and applications*.
- [22] MADHUKAR, B. (2020). Using Near-Miss Algorithm For Imbalanced Datasets. Retrieved from <https://analyticsindiamag.com/using-near-miss-algorithm-for-imbalanced-datasets/>
- [23] Taunk, K., De, S., Verma, S., and Swetapadma, A. (2019). *A brief review of nearest neighbor algorithm for learning and classification*. Paper presented at the 2019 International Conference on Intelligent Computing and Control Systems (ICCS).
- [24] Sá, A., Almeida, A., Rocha, B., Mota, M., Souza, J., and Dentel, L. (2011). *Lightning forecast using data mining techniques on hourly evolution of the convective available potential energy*. Paper presented at the Brazilian Congress on Computational Intelligence, Fortaleza, November.
- [25] Singh, S., and Gupta, P. (2014). Comparative study ID3, cart and C4. 5 decision tree algorithm: a survey. *International Journal of Advanced Information Science and Technology (IJAIST)*, 27(27), 97-103.
- [26] Akosa, J. (2017). *Predictive accuracy: A misleading performance measure for highly imbalanced data*. Paper presented at the Proceedings of the SAS Global Forum.
- [27] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861-874.
- [28] scikit-learn. (2022). Precision-Recall. Retrieved from https://scikit-learn.org/stable/auto-examples/model_selection/plot_precision_recall.html
- [29] Sofaer, H. R., Hoeting, J. A., and Jarnevich, C. S. (2019). The area under the precision-recall curve as a performance metric for rare binary events. *Methods in Ecology and Evolution*, 10(4), 565-577.
- [30] Wikipedia. (2021). Ranking. Retrieved from <https://en.wikipedia.org/wiki/Ranking>
- [31] Huang, L., Zhao, J., Zhu, B., Chen, H., and Broucke, S. V. (2020). An experimental investigation of calibration techniques for imbalanced data. *IEEE Access*, 8, 127343-127352.
- [32] KEEL. Imbalanced data sets. Retrieved from <http://www.keel.es/>
- [33] imbalancedlearn. (2020). fetch_datasets. Retrieved from https://imbalanced-learn.org/stable/references/generated/imblearn.datasets.fetch_datasets.html