การปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวนเคตัวด้วยการให้น้ำหนักกับคุณลักษณะของข้อมูลจากค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันและการปรับละเอียดด้วยอัลกอริทึมพีเอสโอ

IMPROVING KNN ALGORITHM BASED ON WEIGHTED ATTRIBUTES BY PEARSON CORRELATION COEFFICIENT AND PSO FINE TUNING

นายวันนเรศวร์ สิงหัษฐิต

WANARASE SINHASHTHITA

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2564

KMITL-2021-EN-M-070-039

IMPROVING KNN ALGORITHM BASED ON WEIGHTED ATTRIBUTES BY PEARSON CORRELATION COEFFICIENT AND PSO FINE TUNING

WANARASE SINHASHTHITA

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN COMPUTER ENGINEERING
SCHOOL OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2021

KMITL-2021-EN-M-070-039

หัวข้อวิทยานิพนธ์	การปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวนเคตัวด้วยการให้น้ำหนักกับคุณลักษณะของข้อมูลจากค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันและการปรับละเอียดด้วยอัลกอริทึมพีเอสโอ
นักศึกษา	นายวันนเรศวร์ สิงหัษฐิต
รหัสนักศึกษา	59601094
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
พ.ศ.	2564
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.เกียรติกูล เจียรนัยธนะกิจ

บทคัดย่อ

การให้น้ำหนักกับคุณลักษณะของข้อมูลเพื่อระบุความสำคัญของคุณลักษณะเป็นวิธีการปรับปรุงประสิทธิภาพที่สำคัญในการจำแนกประเภท การให้น้ำหนักจะช่วยให้การจำแนกประเภทนั้นมีประสิทธิภาพดีขึ้นหากสามารถระบุความสำคัญของคุณลักษณะได้เหมาะสม ในงานวิจัยนี้ผู้ศึกษาได้ปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยการให้น้ำหนักกับคุณลักษณะจากค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน (Pearson Correlation Coefficient) จากนั้นทำการปรับละเอียดค่าน้ำหนักด้วยอัลกอริทึมการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization) เพื่อให้ค่าน้ำหนักที่ระบุถึงความสำคัญของคุณลักษณะมีค่าที่เหมาะสม ผลการทดลองแสดงให้เห็นว่าวิธีการที่นำเสนอช่วยให้การจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวมีความแม่นยำเพิ่มมากขึ้นเมื่อเปรียบเทียบกับการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม

Thesis Title	Improving KNN Algorithm based on Weighted Attributes by Pearson Correlation Coefficient and PSO Fine Tuning
Student	Mr. Wanarase Sinhashthita
Student ID.	59601094
Degree	Master of Engineering
Program	Computer Engineering
Year	2021
Thesis Advisor	Assoc.Prof.Dr. Kietikul Jearanaitanakij

ABSTRACT

Assigning proper weights to attributes in some datasets according to their importances can significantly improve the classification accuracy. Weighted attributes can support the classification methods effectively if their weights truly represent by their importances. In this research, we improve the K - Nearest Neighbors (KNN) algorithm by using Pearson correlation coefficient along with Particle Swarm Optimization (PSO) to find the optimal set of weights for attributes in the dataset. The experimental results show that the proposed method can significantly improve the classification accuracy when compared to the traditional KNN algorithm.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี เนื่องด้วยความช่วยเหลือจากอาจารย์ที่ปรึกษา รศ.ดร.เกียรติกูล เจียรนัยธนะกิจ ซึ่งท่านได้ให้ทั้งคำแนะนำและข้อสังเกตต่าง ๆ มาโดยตลอดในการทำงาน อีกทั้งยังคอยให้กำลังใจข้าพเจ้าในการดำเนินงาน

ขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ทุกท่านที่ได้กรุณาให้คำแนะนำตลอดจนข้อชี้แนะเพื่อให้เนื้อหาของงานวิจัยครอบคลุมและครบถ้วนจนทำให้วิทยานิพนธ์ฉบับนี้สมบูรณ์และสำเร็จลงได้

ขอขอบคุณเพื่อนและรุ่นพี่ทุกคนที่คอยให้ความอนุเคราะห์ช่วยเหลือในเรื่องต่าง ๆ จนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วง

สุดท้ายนี้ ขออุทิศความดีที่เกิดจากวิทยานิพนธ์ฉบับนี้ให้แด่บิดา มารดา ครอบครัวของข้าพเจ้า ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

วันนเรศวร์ สิงหัษฐิต

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

การจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว (K Nearest Neighbors หรือ KNN) เป็นวิธีที่กำหนดคลาสคำตอบให้กับข้อมูลใหม่หรือข้อมูลทดสอบ (test data) ที่จะจำแนกประเภท อัลกอริทึมจะทำการคำนวณระยะทางแบบยุคลิด (Euclidean distances) ระหว่างข้อมูลทดสอบกับข้อมูลการเรียนรู้ (training data) เพื่อค้นหากลุ่มเพื่อนบ้านที่ใกล้ที่สุดตามค่า K และกำหนดประเภทคำตอบให้กับข้อมูลทดสอบด้วยคลาสคำตอบส่วนใหญ่ของเพื่อนบ้านเหล่านั้น [1-3] แม้ว่าอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวจะเป็นวิธีการจำแนกประเภทที่มีประสิทธิภาพแต่ก็ยังพบกับปัญหาเมื่อนำมาใช้งาน โดยหนึ่งในปัญหาสำคัญของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมคือสมมติฐานที่ว่าแต่ละคุณลักษณะ (attribute of data) มีความสำคัญเท่ากัน ส่งผลให้ความแม่นยำในการจำแนกประเภทมีประสิทธิภาพไม่ดีเท่าที่ควร

การกำหนดน้ำหนักให้กับคุณลักษณะสามารถปรับปรุงความถูกต้องในการจำแนกประเภทได้อย่างมีประสิทธิภาพหากสามารถกำหนดค่าน้ำหนักของคุณลักษณะได้อย่างเหมาะสมตามความสำคัญ ในงานวิจัยเพื่อปรับปรุงประสิทธิภาพของวิธีการจำแนกประเภทแบบต่าง ๆ ได้นำเอาวิธีการให้น้ำหนักกับคุณลักษณะมาใช้ร่วมกัน เช่น การนำเอามาใช้ร่วมกับการจำแนกประเภทแบบต้นไม้ตัดสินใจ (Decision tree) ซึ่งได้นำวิธีการให้น้ำหนักกับคุณลักษณะจากค่าอัตราส่วนขยาย (Gain ratio) [4] การให้น้ำหนักกับคุณลักษณะด้วยการปรับค่าให้เหมาะสมที่สุดในเฉพาะจุดโดยใช้วิธีการซีแคนท์เสมือน (Quasisecant Method) มาใช้กับการจำแนกประเภทแบบเบย์อย่างง่าย (Naïve Bayes) [5] เป็นต้น นอกจากนี้การกำหนดน้ำหนักให้กับคุณลักษณะยังถูกนำมาใช้กับวิธีการค้นหาเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว โดยในงานวิจัยของ Li และคณะ [6] ได้นำเสนออัลกอริทึมการให้น้ำหนักกับคุณลักษณะด้วยอัตราความผิดพลาดของการจำแนกประเภทเพื่อวัดความสำคัญของคุณลักษณะ หากนำคุณลักษณะที่สำคัญออกจากข้อมูลการเรียนรู้อัตราความผิดพลาดในการจำแนกประเภทย่อมมีความน่าจะเป็นที่จะเพิ่มขึ้น ในทางตรงกันข้ามหากนำคุณลักษณะที่ไม่เกี่ยวข้องออกไปย่อมจะลดอัตราความผิดพลาดลง หรือในงานวิจัยของ Huang และคณะ [7] ซึ่งมีลักษณะของอัลกอริทึมที่ใกล้เคียงกัน โดยค้นหาความสำคัญของแต่ละคุณลักษณะด้วยการพิจารณาความแม่นยำในการจำแนกประเภทจากการนำเอาคุณลักษณะแต่ละประเภทออกในแต่ละรอบการจำแนก

การค้นหาค่าน้ำหนักที่เหมาะสมสำหรับคุณลักษณะยังเป็นสิ่งที่ท้าทายสำหรับงานวิจัยในปัจจุบัน งานวิจัยหลายงานได้กล่าวถึงการนำวิธีการหาค่าความสัมพันธ์ของข้อมูลที่มีอยู่มาใช้ในการ

ค้นหาค่าน้ำหนักของคุณลักษณะ เช่น ในงานวิจัยของ Diego และคณะ [8] ได้กำหนดน้ำหนักของคุณลักษณะด้วยค่าทดสอบทางสถิติไคสแควร์ (Chi-Square Test) ของแต่ละคุณลักษณะกับคลาสคำตอบของชุดข้อมูล งานวิจัยของ Xiao และคณะ [9] ได้กำหนดค่าน้ำหนักของแต่ละคุณลักษณะด้วยการคำนวณค่าเกนความรู้ (Information gain) เพื่อใช้ในการปรับปรุงประสิทธิภาพ เป็นต้น การกำหนดค่าน้ำหนักเพื่อปรับปรุงประสิทธิภาพไม่ได้กำหนดค่าน้ำหนักเพียงแค่กับคุณลักษณะเท่านั้น ในงานวิจัยของ Gou และคณะ [10] ได้นำเสนอการให้ค่าน้ำหนักกับระยะทาง ซึ่งคำนวณจากระยะทางของเพื่อนบ้านเหล่านั้นเพื่อจัดลำดับความสำคัญ โดยงานวิจัยทั้งสองมีพื้นฐานแนวคิดจากอัลกอริทึม WKNN ซึ่งได้ถูกนำเสนอโดย Dudani [11] ในบางงานวิจัยพยายามนำเอาเอาวิธีการปรับปรุงประสิทธิภาพแบบต่าง ๆ มาใช้ร่วมกันเพื่อแก้ปัญหาข้อจำกัดของวิธีการแบบดั้งเดิมโดยในงานวิจัยของ Wu และคณะ [12] และงานวิจัยของ Taneja และคณะ [13] ได้กล่าวถึงวิธีการแก้ปัญหา 3 วิธีคือ กำหนดวิธีการคำนวณระยะทางที่ดีกว่าวิธีการยุคลิดแบบมาตรฐาน การค้นหาค่า K ที่ดีที่สุดด้วยการค้นหาแบบพลวัต และกำหนดวิธีการจำแนกประเภทที่มีประสิทธิภาพมากกว่าการกำหนดคลาสคำตอบด้วยคำตอบส่วนใหญ่ของเพื่อนบ้าน K ตัวนั้น

ยิ่งไปกว่านั้นเพื่อปรับปรุงความแม่นยำให้มีประสิทธิภาพที่สูงยิ่งขึ้น ในบางงานวิจัยได้หาค่าน้ำหนักของคุณลักษณะที่เหมาะสมที่สุดโดยใช้อัลกอริทึมการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization) [14] เพื่อการปรับละเอียดค่าน้ำหนัก โดยกำหนดให้ค่าน้ำหนักเป็นตำแหน่งเริ่มต้นของอนุภาคและใช้ความแม่นยำในการจำแนกประเภทเป็นค่าของฟังก์ชันความเหมาะสม (fitness function) งานวิจัยของ Guo และคณะ [15] ได้นำเสนออัลกอริทึมเพื่อค้นหาค่าน้ำหนักที่เหมาะสมซึ่งใช้การสุ่มค่าน้ำหนักเป็นตำแหน่งเริ่มต้นจากนั้นทำการปรับค่าน้ำหนักด้วยวิธีการปรับมาตรฐานเพื่อหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค [16] นอกจากนี้ในงานวิจัยดังกล่าวยังอ้างถึงงานวิจัยของ Cao และ Liu [17] ซึ่งใช้การหาค่าน้ำหนักที่เหมาะสมที่สุดแบบกลุ่มอนุภาคด้วยเช่นกัน แต่ในงานของ Cao และ Liu ได้นำเอาอัลกอริทึมแบบคลาวด์มาใช้กำหนดค่าน้ำหนักเริ่มต้นแทนการสุ่ม อย่างไรก็ตามอัลกอริทึมที่ใช้การปรับละเอียดค่าน้ำหนักที่กล่าวถึงข้างต้นยังไม่สามารถหาน้ำหนักที่เหมาะสมได้ภายในเวลาที่สมเหตุผล ดังนั้นเป้าหมายของการศึกษาในงานวิจัยชิ้นนี้คือค้นหาค่าน้ำหนักที่เหมาะสมสำหรับแต่ละคุณลักษณะ โดยที่สามารถหาค่าน้ำหนักดังกล่าวนั้นได้ภายในเวลาที่เหมาะสม

ในงานวิจัยชิ้นนี้ผู้วิจัยใช้ค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน (Pearson correlation coefficient) เพื่อกำหนดค่าน้ำหนักเริ่มต้นให้กับแต่ละคุณลักษณะโดยหาความสัมพันธ์ระหว่างแต่ละคุณลักษณะและคลาสเป้าหมาย (target class) ของชุดข้อมูลนั้น ๆ ซึ่งจะถูกเรียกว่าเป็นการค้นหาค่าน้ำหนักแบบหยาบ จากนั้นทำการปรับละเอียดค่าน้ำหนักของแต่ละคุณลักษณะด้วยอัลกอริทึมการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization) เพื่อให้ค่าน้ำหนักที่ระบุถึง

ความสำคัญของคุณลักษณะมีค่าที่เหมาะสมและส่งผลต่อการจำแนกประเภทที่ถูกต้อง จากผลการทดลองแสดงให้เห็นว่าวิธีการที่นำเสนอช่วยให้การจำแนกประเภทสามารถจำแนกได้ถูกต้องมากยิ่งขึ้นเมื่อเปรียบเทียบกับการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม และสามารถใช้เวลาอย่างเหมาะสมเมื่อเปรียบเทียบกับงานวิจัยที่มีลักษณะใกล้เคียงกัน

1.2 จุดมุ่งหมายและวัตถุประสงค์

1.2.1 เพื่อศึกษาและวิจัยอัลกอริทึมที่ใช้ในการปรับปรุงประสิทธิภาพของการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว

1.2.2 เพื่อสามารถแก้ปัญหาด้านความแม่นยำในการจำแนกประเภทเมื่อนำมาเปรียบเทียบกับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม

1.2.3 เพื่อแก้ปัญหาในการให้ความสำคัญกับข้อมูลของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมเมื่อคุณลักษณะของชุดข้อมูลทดสอบมีความสำคัญไม่เท่ากัน

1.2.4 เพื่อสามารถแก้ปัญหาด้านการใช้เวลาที่เหมาะสมของวิธีการหาค่าเหมาะที่สุดเมื่อนำมาใช้กับการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว

1.3 สมมติฐานของการศึกษา

จากงานวิจัยที่เกี่ยวข้องได้มีการพัฒนาอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว พบว่ามีปัญหาซึ่งส่งผลต่อความแม่นยำในการจำแนกประเภท เนื่องจากวิธีการในการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมนั้นมองว่าคุณลักษณะแต่ละประเภทของข้อมูลมีความสำคัญเท่ากัน ซึ่งในความเป็นจริงแล้วคุณลักษณะแต่ละประเภทย่อมมีความสำคัญที่แตกต่างกัน ดังนั้นหากนำวิธีการที่สามารถระบุถึงความสำคัญ (น้ำหนัก) ของแต่ละคุณลักษณะมาใช้ในการคำนวณระยะทางเพื่อจำแนกประเภทแล้ว ย่อมมีความเป็นได้ว่าจะสามารถเพิ่มประสิทธิภาพความถูกต้องของวิธีการจำแนกประเภทแบบดั้งเดิมได้ และเมื่อนำการปรับละเอียดค่าความสำคัญของแต่ละคุณลักษณะให้เหมาะสมได้ย่อมส่งผลที่ดียิ่งขึ้นกว่าก่อนปรับละเอียด เพียงแต่ว่าในการปรับละเอียดนั้นจะนำมาซึ่งการใช้เวลาที่เพิ่มมากขึ้น ดังนั้นการระบุค่าความสำคัญของคุณลักษณะเบื้องต้นหรือการค้นหาแบบหยาบที่มีประสิทธิภาพจะช่วยให้การปรับละเอียดสามารถใช้เวลาได้อย่างเหมาะสม

1.4 ขอบเขตการวิจัย

วิทยานิพนธ์เล่มนี้เป็นการวิจัยเพื่อปรับปรุงประสิทธิภาพของการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวโดยมีขอบเขตการวิจัยดังต่อไปนี้

1.4.1 เพื่อการปรับปรุงประสิทธิภาพด้านความถูกต้องของการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว

1.4.2 เพื่อแก้ปัญหาด้านการใช้เวลาที่เหมาะสมของวิธีการหาค่าเหมาะที่สุดเมื่อนำมาใช้กับการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว

1.4.3 เพื่อเปรียบเทียบและวัดประสิทธิภาพของวิธีการที่นำเสนอกับวิธีการแบบดั้งเดิม และงานวิจัยที่เกี่ยวข้องด้วยข้อมูลสาธารณะ UCI

1.5 ขั้นตอนการศึกษา

ขั้นตอนการศึกษาของงานวิจัยนี้สามารถอธิบายได้ดังต่อไปนี้

1.5.1 ศึกษาทฤษฎีพื้นฐานและปัญหาของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว

1.5.2 ศึกษาและเลือกงานวิจัยเกี่ยวกับการนำวิธีการหาค่าที่เหมาะสมที่สุดมาใช้งานในการจำแนกประเภทร่วมกับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้ค่าน้ำหนักกับคุณลักษณะ

1.5.3 ทดสอบผลลัพธ์ของการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมและงานวิจัยที่เกี่ยวข้อง จากนั้นทำการวิเคราะห์ผล

1.5.4 ศึกษาวิธีการในการให้น้ำหนักกับคุณลักษณะและออกแบบอัลกอริทึมเพื่อแก้ไขปัญหา

1.5.5 ศึกษาประเภทของข้อมูลที่ใช้ในการนำมาทดสอบ

1.5.6 ตรวจสอบอัลกอริทึมที่ออกแบบเพื่อปรับค่าของตัวแปรที่ใช้ในอัลกอริทึม

1.5.7 ทดสอบอัลกอริทึมที่ออกแบบ

1.5.8 วิเคราะห์ผลลัพธ์ของอัลกอริทึมที่ออกแบบเพื่อพัฒนาและปรับปรุง

1.5.9 ทำการทดสอบอัลกอริทึมที่ออกแบบ

1.5.10 สรุปและวิเคราะห์ผลการทดลอง

1.5.11 จัดทำวิทยานิพนธ์

1.6 เครื่องมือและอุปกรณ์ที่ใช้ในงานวิจัย

1.6.1 เครื่องคอมพิวเตอร์ส่วนบุคคล หน่วยประมวลผลกลาง AMD FX-8320 หน่วยความจำหลัก (RAM) 16 GB จำนวน 1 เครื่อง

1.6.2 ระบบปฏิบัติการ Microsoft Windows 10

1.6.3 Python 3.7.1

1.6.4 Google Colaboratory – Python executor browser

1.7 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์เล่มนี้ได้แบ่งเนื้อหาออกเป็น 7 บท โดยแบ่งเป็นภาคผนวกอยู่ 1 บท แต่ละบทจะมีรายละเอียดดังนี้

บทที่ 1 อธิบายถึงที่มาของงานวิจัย วัตถุประสงค์ สมมติฐานของการศึกษา ขอบเขตของการวิจัย ขั้นตอนของการศึกษา เครื่องมือและอุปกรณ์ที่ใช้ในการวิจัย และโครงสร้างของวิทยานิพนธ์

บทที่ 2 อธิบายถึงทฤษฎีและความรู้พื้นฐานที่เกี่ยวข้องในการพัฒนาอัลกอริทึมได้แก่ อัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว ระยะทางแบบยุคลิด การหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค และค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน

บทที่ 3 อธิบายถึงงานวิจัยที่เกี่ยวข้องซึ่งได้มีการปรับปรุงประสิทธิภาพของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว เช่น อัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนักและการประยุกต์ใช้กับข้อมูลสาธารณะ UCI (Weighted-KNN and its application on UCI) การปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยผลสนับสนุนจากคลาสคำตอบและการให้น้ำหนักกับคุณลักษณะ (An Improved kNN Based on Class Contribution and Feature Weighting) และวิธีการปรับมาตรฐานเพื่อหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคและการประยุกต์ใช้ในปัญหาการให้น้ำหนักกับคุณลักษณะ (The Normalized-PSO and Its Application in Attribute Weighted Optimal Problem) เป็นต้น

บทที่ 4 อธิบายถึงแนวคิดในการแก้ปัญหา วิธีการที่นำมาใช้ในการปรับปรุงอัลกอริทึมแบบดั้งเดิม และขั้นตอนการทำงานของอัลกอริทึมที่นำเสนอ

บทที่ 5 อธิบายวิธีการทดลองโดยทำการทดลองกับชุดข้อมูลสาธารณะ UCI จำนวน 10 ชุดข้อมูล และเปรียบเทียบผลการทดลองระหว่างอัลกอริทึมที่นำเสนอและอัลกอริทึมที่เกี่ยวข้อง

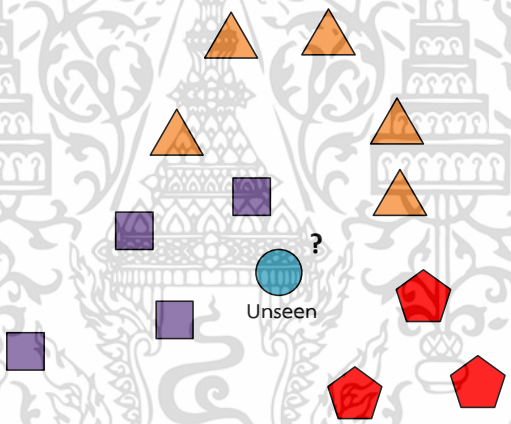
บทที่ 6 สรุปผลการทดลอง ข้อเสนอแนะ และแนวทางในการปรับปรุง

ภาคผนวก ก งานวิจัยที่ได้รับการตีพิมพ์

บทที่ 2
ทฤษฎีพื้นฐานที่เกี่ยวข้อง

2.1 การจำแนกประเภทแบบอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว

อัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว (K Nearest Neighbors หรือ KNN) คือวิธีการหนึ่งที่ใช้กันอย่างแพร่หลายในการเรียนรู้ของเครื่อง (Machine learning) สำหรับการวิเคราะห์ข้อมูลเพื่อนำความรู้มาใช้ โดยจัดเป็นการเรียนรู้แบบมีผู้สอน (Supervised Learning) วิธีการจำแนกประเภทมีวิธีการที่ไม่ซับซ้อนและสามารถเข้าใจได้ง่ายเนื่องจากลักษณะพิเศษของอัลกอริทึมคือการเรียนรู้แบบเกียจคร้าน (Lazy Learning) การทำงานของอัลกอริทึมจะไม่มีการสร้างโมเดลเตรียมไว้สำหรับการคำนวณเพื่อจำแนกประเภท แต่จะประมวลผลข้อมูลทดสอบใหม่และชุดข้อมูลที่มีอยู่ก่อนทุกครั้งเมื่ออัลกอริทึมทำการจำแนก ซึ่งสามารถแสดงตัวอย่างดังรูปที่ 2.1



รูปที่ 2.1 ตัวอย่างแสดงเหตุการณ์เมื่อมีข้อมูลใหม่ในการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว

จากรูปที่ 2.1 วงกลมในรูปคือข้อมูลใหม่ (Unseen data) ที่รับมาเพื่อจะทำการจำแนกซึ่งยังไม่ถูกจำแนกว่าเป็นข้อมูลประเภทใด เมื่ออัลกอริทึมต้องการจำแนกประเภทก็จะเปรียบเทียบความคล้ายคลึงกันของข้อมูลใหม่และชุดข้อมูลที่มีอยู่ก่อน แล้วจำแนกประเภทข้อมูลใหม่โดยกำหนดให้เป็นประเภทเดียวกับคำตอบ (คลาสคำตอบ) ของเพื่อนบ้านที่อยู่ใกล้เคียง [1-3] การทำงานของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวจะประกอบด้วย 2 ส่วนคือ ส่วนที่หนึ่งทำการคำนวณระยะทางระหว่างข้อมูลใหม่ที่รับเข้ามากับชุดข้อมูลที่มีอยู่ก่อนซึ่งเราจะเรียกส่วนนี้ว่าข้อมูลการเรียนรู้ (training data) และส่วนที่สองทำการเลือกกลุ่มเพื่อนบ้านที่ใกล้ที่สุดจากชุดข้อมูลการเรียนรู้แล้วด้วยค่าระยะทางน้อยที่สุดจำนวน K ตัว จากนั้นกำหนดคลาสคำตอบให้กับข้อมูลใหม่จากคลาสคำตอบของ

เพื่อนบ้านที่มีจำนวนมากที่สุด หรือ คำตอบส่วนใหญ่ (majority answer) ของกลุ่มเพื่อนบ้านที่ใกล้ที่สุด K ตัวนั้น โดยสามารถอธิบายขั้นตอนการจำแนกประเภทด้วยรหัสเทียมดังรูปที่ 2.2

K-Nearest Neighbor

Determine (K, distance)
Classify (X, Y, x) // X is training data, Y is labels of X, x is unknown sample

for i = 1 to n do // n is all the training data

Compute distance d $(Xi,)$

end for

Select training member to Set L which contain the K smallest distance d $(Xi,)$.
Return labels of x with majority label Y of smallest distance $\{Y$ where $Y \in L\}$

รูปที่ 2.2 รหัสเทียมของการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว

จากรูปที่ 2.2 เป็นรหัสเทียมของการจำแนกประเภทแบบอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวซึ่งสามารถอธิบายการทำงานเป็น 4 ขั้นตอนได้ดังนี้

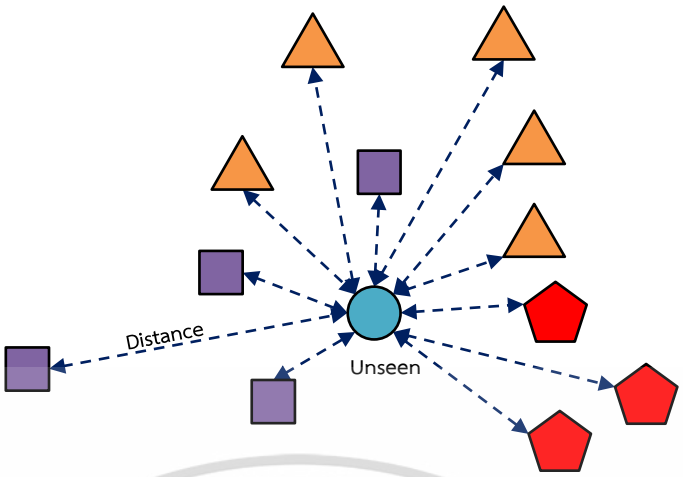
1. กำหนดค่า K และวิธีการคำนวณระยะทางของข้อมูลสำหรับเรียนรู้

2. คำนวณระยะทางระหว่างข้อมูลใหม่ที่ต้องการจำแนกกับข้อมูลสำหรับเรียนรู้

3. จัดลำดับระยะทางตามค่าน้อยไปยังค่ามากและกำหนดเพื่อนบ้านใกล้ที่สุดตามค่า K

4. นับจำนวนประเภทคลาสคำตอบของเพื่อนบ้านที่ใกล้ที่สุด K ตัวว่าเป็นประเภทใด แล้วทำการกำหนดคลาสคำตอบของข้อมูลใหม่จากประเภทคลาสคำตอบส่วนใหญ่ของกลุ่มเพื่อนบ้าน K ตัวนั้น

จากขั้นตอนของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวทั้ง 4 ขั้นตอน เราสามารถแสดงตัวอย่างของการคำนวณระยะทางระหว่างข้อมูลใหม่ หรือ ข้อมูลทดสอบ (test data) ที่ต้องการจำแนกกับข้อมูลการเรียนรู้ด้วยวิธีการค้นหาเพื่อนบ้านที่ใกล้ที่สุดจำนวน K ตัวมาใช้จำแนกประเภทให้กับข้อมูลใหม่ตามตัวอย่างปัญหาในการจำแนกประเภทดังรูปที่ 2.3



รูปที่ 2.3 การคำนวณระยะทางระหว่างข้อมูลทดสอบกับข้อมูลการเรียนรู้

จากขั้นตอนที่ 3 เมื่อนำระยะทางที่คำนวณได้ระหว่างข้อมูลทดสอบกับข้อมูลการเรียนรู้ในแต่ละประเภทมาจัดลำดับระยะทางจากน้อยไปมากจะได้ผลของการคำนวณดังตารางที่ 2.1 ดัง

ตารางที่ 2.1 ตารางแสดงผลการจัดลำดับระยะทางระหว่างข้อมูลทดสอบและข้อมูลการเรียนรู้

ลำดับที่	จุด	พิกัด (X , Y)	class
1	a	(a1 , a2)	
2	b	(b1 , b2)	
3	c	(c1 , c2)	
4	d	(d1 , d2)	
5	e	(e1 , e2)	
...	...	...	...
12	k	(k1 , k2)	

จากผลในตารางที่ 2.1 เมื่อทำการพิจารณาค่า K ที่กำหนด อัลกอริทึมจะกำหนดประเภทคลาสคำตอบของข้อมูลใหม่จากคำตอบส่วนใหญ่ของกลุ่มเพื่อนบ้านที่ใกล้ที่สุด K ตัวว่าเป็นประเภทใด หากกำหนดให้ค่า K เป็น 3 คำตอบของข้อมูลใหม่ก็จะถูกกำหนดเป็นคลาสสี่เหลี่ยม และหากกำหนดให้ค่า K เป็น 5 แม้จะมีคำตอบอีก 2 ประเภทที่แตกต่างเพิ่มขึ้นมาคำตอบของข้อมูลก็จะถูกกำหนดเป็นคลาสสี่เหลี่ยมเช่นเดิมจากคำตอบส่วนใหญ่ของเพื่อนบ้านที่ใกล้ที่สุดทั้ง 5 ตัวนั้น โดยส่วนใหญ่แล้วการคำนวณระยะทางระหว่างข้อมูลทดสอบและข้อมูลการเรียนรู้ของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวจะนิยมใช้ระยะทางแบบยุคลิด (Euclidean Distance) ในการคำนวณ แต่ในบางงานวิจัยก็ได้มีการนำวิธีคำนวณระยะทางตามการวัดระยะรูปแบบต่าง ๆ [18] มาใช้อีกด้วย เช่น ระยะทางแบบแมนแฮตตัน (Manhattan Distance) ระยะทางแบบมิงค์อฟสกี (Minkowski distance) เป็นต้น

ในงานวิจัยของ Rani [1] ได้กล่าวถึงปัญหาของการนำอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม (Traditional KNN) มาใช้งานในการจำแนกประเภท เช่น ปัญหาในการค้นหากลุ่มเพื่อนบ้านที่ใกล้ที่สุดมีความล่าช้าเมื่อชุดข้อมูลมีขนาดใหญ่ เนื่องจากอัลกอริทึมต้องคำนวณระยะทางระหว่างข้อมูลทดสอบและข้อมูลการเรียนรู้ทุกครั้งเมื่อต้องจำแนกข้อมูลทดสอบใหม่ ปัญหาการเลือกค่า K ที่เหมาะสม และปัญหาจากสมมติฐานที่ว่าทุกคุณลักษณะมีความสำคัญเท่ากัน เป็นต้น นักวิจัยได้ปรับปรุงประสิทธิภาพจากปัญหาดังกล่าวด้วยการนำเอาวิธีต่าง ๆ มาปรับใช้ ในงานวิจัยของ Taneja และคณะ [13] ได้แก้ปัญหาในด้านความซับซ้อนของเวลาในการทำงานของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม โดยแบ่งข้อมูลการเรียนรู้ออกเป็นกลุ่ม (Cluster) ตามลักษณะของข้อมูลที่คล้ายกัน ซึ่งจะช่วยลดเวลาในการคำนวณเนื่องจากข้อมูลทดสอบจะเลือกคำนวณระยะห่างระหว่างข้อมูลทดสอบและข้อมูลการเรียนรู้ของกลุ่มซึ่งมีระยะทางใกล้กับข้อมูลทดสอบมากที่สุด ปัญหาการเลือกค่า K ที่เหมาะสมเป็นปัญหาซึ่งได้มีงานวิจัยจำนวนมากได้กล่าวถึง เนื่องจากการเลือกค่า K ในแต่ละชุดข้อมูลจะส่งผลต่อการจำแนกที่มีประสิทธิภาพแตกต่างกัน ในงานวิจัยของ Wu และคณะ [12] และงานวิจัยของ Taneja และคณะ [13] ได้ค้นหาค่า K ที่ดีที่สุดจากการค้นหาแบบพลวัตเนื่องจากต้องการเลือกค่า K ที่เหมาะสมกับปัญหาเหล่านั้น งานวิจัยของ Liu และคณะ [19] ซึ่งค้นหาค่า K ที่เหมาะสมของข้อมูลทดสอบจากผลการจำแนกประเภทของข้อมูลการเรียนรู้ด้วยช่วงของค่า K ที่กำหนด หรือในงานวิจัยของ Dudani [11] ซึ่งพยายามที่จะลดอิทธิพลของเพื่อนบ้านทั้ง K ตัวลง เนื่องจากการกำหนดค่า K ที่ไม่เหมาะสมในปัญหาเหล่านั้น โดยจะพิจารณาเพียงเพื่อนบ้านซึ่งมีอิทธิพลจริงต่อข้อมูลทดสอบด้วยอัลกอริทึม WKNN นอกจากนี้อีกหนึ่งปัญหาสำคัญที่ถูกกล่าวถึงของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมคือ สมมติฐานว่าแต่ละคุณลักษณะมีความสำคัญเท่าเทียมกัน ซึ่งความเป็นจริงแล้วแต่ละคุณลักษณะย่อมมีความสำคัญที่แตกต่างกันซึ่งส่งผลต่อประสิทธิภาพ ในการแก้ปัญหานักวิจัยพยายามกำหนดน้ำหนักให้กับคุณลักษณะอย่างเหมาะสมตามความสำคัญของคุณลักษณะนั้น แต่ก็ยังไม่มีงานวิจัยที่สามารถค้นหาค่าน้ำหนักได้อย่างเหมาะสมที่สุดและสามารถนำมาใช้แทนการจำแนกประเภทแบบดั้งเดิมได้ทั้งหมด

2.2 ระยะทางแบบยุคลิด

ระยะทางแบบยุคลิด (Euclidean Distance) เป็นมาตรวัดระยะพื้นฐานเป็นที่นิยมใช้ในงานประเภทต่าง ๆ สำหรับการหาระยะทางระหว่างจุดสองจุด ซึ่งมีที่มาของลักษณะการคำนวณจากทฤษฎีบทพีทาโกรัส [18] โดยระยะทางแบบยุคลิดระหว่างจุดสองจุด p และ q คือความยาวของส่วนของเส้นตรง p และ q หรือ $d(p,q)$ ถ้า $p = (p_1, p_2, ..., p_n)$ และ $q = (q_1, q_2, ..., q_n)$ ในระบบพิกัดคาร์ทีเซียนเป็นจุดสองจุดบนปริภูมิยุคลิด n มิติ ระยะทางระหว่างจุด p กับ q คำนวณได้ตามสมการที่ 2.1 ดังนี้

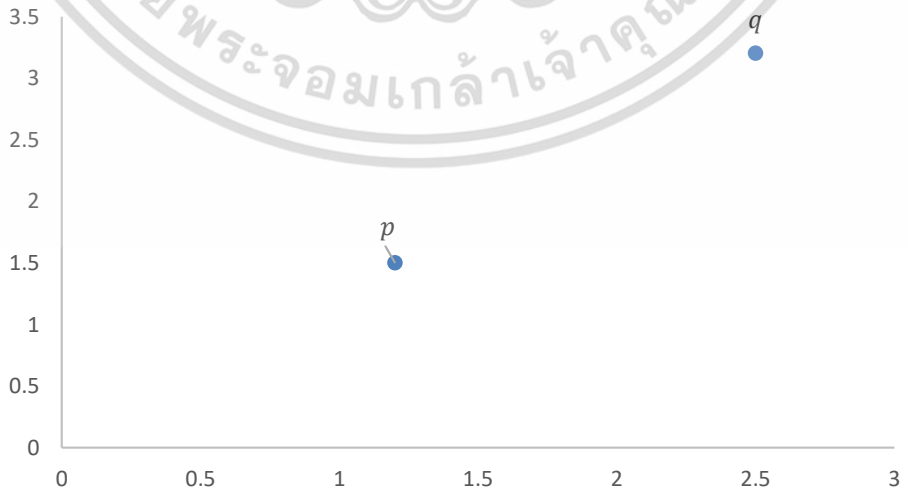
$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \qquad (2.1)$$

โดยที่ p_i คือพิกัดที่ i ของจุด p

q_i คือพิกัดที่ i ของจุด q

n คือจำนวนมิติของข้อมูล

โดยสามารถแสดงตัวอย่างข้อมูลที่ใช้ในการคำนวณระยะทางแบบยุคลิดระหว่างจุด p และ q ดังรูปที่ 2.4



รูปที่ 2.4 ตัวอย่างแสดงข้อมูลที่ใช้ในการคำนวณระยะทางแบบยุคลิดระหว่างจุด p และ q

จากรูปที่ 2.4 แสดงตัวอย่างข้อมูลที่ใช้ในการคำนวณระยะทางแบบยุคลิดโดยกำหนดให้พิกัดของจุด p และ q ในระบบพิกัดคาร์ทีเซียนคือ (1.2 , 1.5) และ (2.5 , 3.2) ตามลำดับ การคำนวณระยะทางระหว่างจุด p และ q ด้วยระยะทางแบบยุคลิดสามารถแสดงได้ดังนี้

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

$$= \sqrt{(1.2 - 2.5)^2 + (1.5 - 3.2)^2}$$

$$= \sqrt{1.69 + 2.89}$$

$$= \sqrt{4.58}$$

$$= 2.14009345$$

2.3 อัลกอริทึมการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค

การหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization หรือ PSO) เป็นวิธีการแก้ปัญหาการหาค่าเหมาะสมที่สุด (Optimization Problem) วิธีหนึ่งที่ค่อนข้างได้รับความนิยม ถูกนำเสนอในปี ค.ศ. 1995 โดย J. Kennedy และ R. Eberhart [14] ได้นำเสนอการหาค่าที่เหมาะสมที่สุดด้วยวิธีแบบกลุ่มอนุภาคด้วยการจำลองพฤติกรรมของการออกไปหาอาหารของฝูงนกมาเป็นขั้นตอนวิธีในการค้นหา โดยคล้ายคลึงกับนกในฝูงจะช่วยกันออกไปหาอาหาร ซึ่งนกแต่ละตัวไม่รู้ว่าสถานที่ที่มีอาหารอยู่ตำแหน่งใดแต่จะรู้ว่าว่าอยู่ห่างจากอาหารเท่าใด ดังนั้นเมื่อตัวใดตัวหนึ่งพบอาหารจะเกิดการเปรียบเทียบเพื่อไปยังแหล่งอาหารที่อยู่ใกล้ที่สุดที่ค้นพบ จากนั้นนกทั้งฝูงจะเลือกบินตามนกตัวที่อยู่ใกล้แหล่งอาหารมากที่สุดจนพวกมันบินมาถึงแหล่งอาหาร

ซึ่งผู้นำเสนอได้อธิบายว่านกแต่ละตัวคืออนุภาค (Particle) ทุกอนุภาคจะมีฟังก์ชันค่าความเหมาะสม (Fitness Values) ซึ่งจะถูกประเมินด้วยฟังก์ชันวัตถุประสงค์ (Objective Function) อนุภาคจะทำการหาตำแหน่งที่เหมาะสมที่สุด ซึ่งแต่ละอนุภาคจะมีหน่วยความจำเป็นของตนเองเอาไว้เก็บตำแหน่งที่เหมาะสมที่สุดของตนเอง (Particle Best) และจะทำการกระจายข้อมูลให้กับอนุภาคตัวอื่นๆ หากตัวไหนมีตำแหน่งที่เหมาะสมที่สุดจากทั้งฝูงตำแหน่งนั้นจะกลายเป็นตำแหน่งจ่าฝูง (Global Best) และอนุภาคจะทำการเคลื่อนที่เข้าหาจ่าฝูง [19] โดยตัวแปรสำคัญที่ทำให้อนุภาคเคลื่อนที่ได้คือ V (Velocity หรือ ความเร็ว) และ X (Position หรือ ตำแหน่ง) ตามสมการ 2.2 และ 2.3 ดังนี้

$$V_b[i+1] = (\omega * V_b) + c_1 r_1 (GB - X_b) + c_2 r_2 (PB_b - X_b) \qquad (2.2)$$

$$X_b[i+1] = V_b[i+1] + X_b[i] \qquad (2.3)$$

โดย $b = 1,2,3, \ldots, n$ คืออนุภาค

V_b คือค่าความเร็ว (Velocity) ของนกแต่ละตัวซึ่งจะเป็นตัวบอกทิศทางและระยะการเคลื่อนที่ของอนุภาค

X_b คือตำแหน่งอนุภาคแต่ละตัว

$PBest_b$ คือตำแหน่งที่ดีที่สุดของอนุภาค b นั้น

$GBest$ คือตำแหน่งที่ดีที่สุดของฝูงอนุภาค

ω คือค่าถ่วงทิศทาง

r_1, r_2 คือค่าสุ่มอยู่ในช่วง [0,1]

c_1 คือ ตัวแปรทางสังคม (Social Parameter) และ c_2 คือตัวแปรกระบวนการรับรู้ (Cognitive Parameter)

ขั้นตอนการทำงานของ PSO มีสามารถอธิบายได้ดังต่อไปนี้

ขั้นตอนที่ 1 เป็นการกำหนดค่าเริ่มต้นให้กับทุกอนุภาคของ PSO โดยจะสำหรับทุกอนุภาค b ในกลุ่มอนุภาค

ขั้นตอนที่ 1.1 สุ่มค่าเริ่มต้นให้ X_b

ขั้นตอนที่ 1.2 สุ่มค่าเริ่มต้นให้ V_b

ขั้นตอนที่ 2 เป็นขั้นตอนการประเมินค่า $f(x)$ ค่าตำแหน่งของอนุภาค (Fitness Value) ด้วยการนำค่าตำแหน่งของอนุภาค (X_b) มาคำนวณค่าฟังก์ชันวัตถุประสงค์แทนด้วย $f(X_b)$ ซึ่งในขั้นตอนนี้จะทำการปรับค่า $PBest_b$ ด้วยตำแหน่งของอนุภาคที่มีค่าฟังก์ชันดีที่สุดซึ่งหากเป็นรอบแรกก็จะถูกกำหนดจากค่าเริ่มต้น และทำการปรับค่าของ $GBest$ โดยสามารถอธิบายได้ดังนี้

$$f[GBest] < f[PBest_b], \forall b \leq N$$

การปรับ $PBest_b$ สำหรับทุกอนุภาค b

$$PBest_b = X_b \text{ ถ้า } f[X_b] < f[PBest_b] \; \forall b \leq N$$

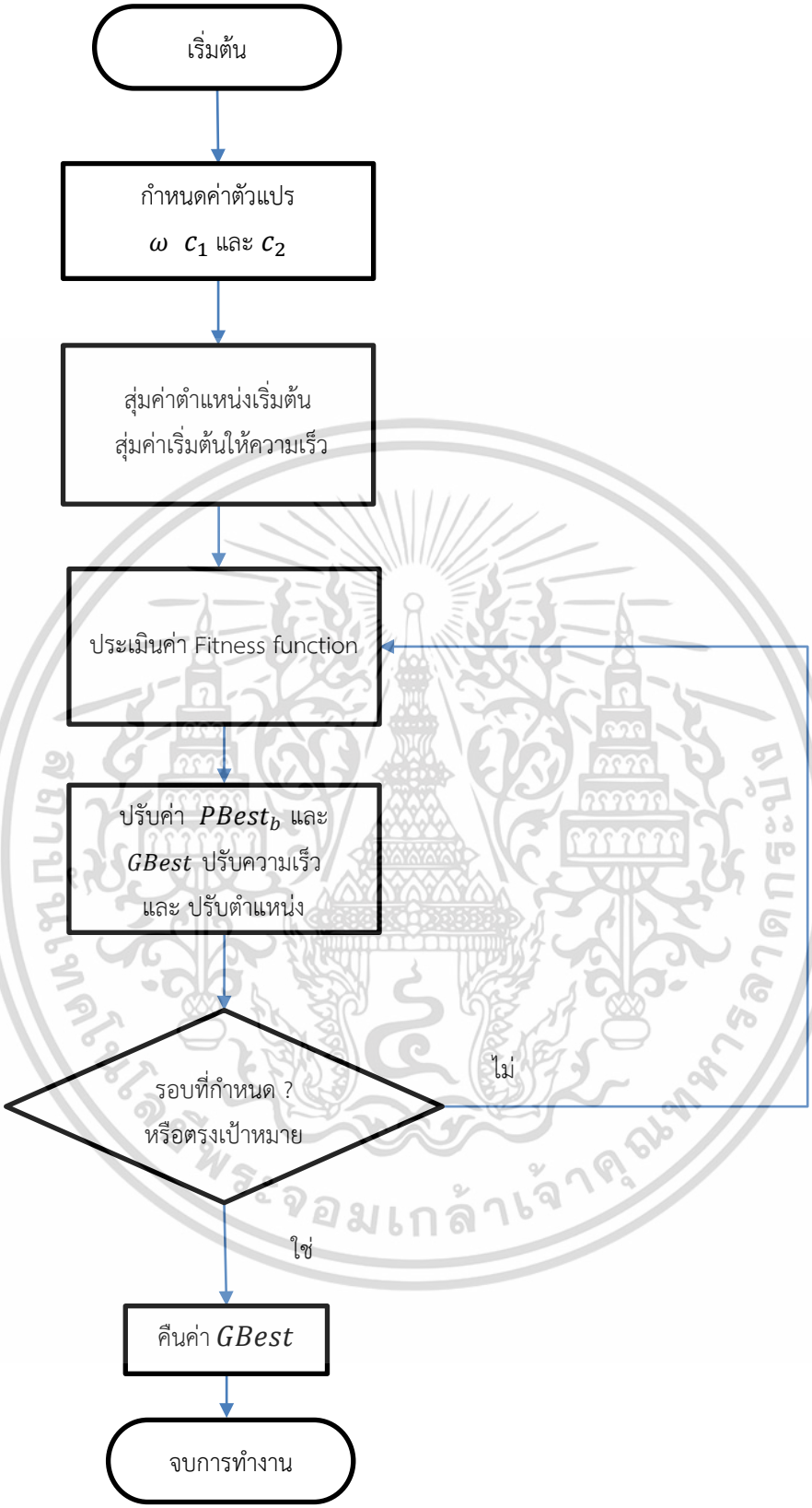
ขั้นตอนที่ 2.3 สำหรับอนุภาค b, ปรับปรุง V_b และ X_b โดยจะใช้สมการ (1), (2)

ขั้นตอนที่ 2.4 ประเมิน $f[X_b]$ ทุกอนุภาค

ขั้นตอนที่ 3 เป็นขั้นตอนในการตรวจสอบ $GBest$ ว่าได้ค่าคำตอบที่พึงพอใจแล้วหรือไม่ หากใช่ก็จะจบการทำงานของ PSO และหากไม่จะดำเนินการค้นหาต่อในขั้นตอนที่ 4 ต่อ ซึ่งในขั้นตอนนี้ก็จะมีการนับจำนวนรอบหากครบจำนวนรอบการค้นหาสูงสุดที่กำหนดก็จะหยุดการค้นหา หรือในบางกรณีสามารถกำหนดหากค่า $GBest$ ไม่มีการเปลี่ยนแปลงค่าในรอบการค้นหาที่กำหนดก็สามารถหยุดการค้นหาได้เช่นกัน

ขั้นตอนที่ 4 เป็นการปรับปรุงค่าความเร็วในการเคลื่อนที่ให้กับอนุภาค และการปรับค่าตำแหน่งเมื่อเคลื่อนที่ให้กับอนุภาค โดยสามารถคำนวณได้จากสมการที่ 2.2 และ 2.3 โดยค่าตำแหน่งของอนุภาคก็จะเป็นค่าที่อยู่ในตำแหน่งที่โดเมนของปัญหาที่เราต้องการจะค้นหาค่าในช่วง $[-X_{b_{max}}, X_{b_{max}}]$ และสามารถกำหนดค่า Velocity ให้อยู่ในช่วง $[-V_{max}, V_{max}]$ เพื่อที่จะลดพื้นที่ในการค้นหาลง

โดยสามารถแสดงผังงานอธิบายขั้นตอนการทำงานของ อัลกอริทึมการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคดังรูปที่ 2.5



รูปที่ 2.5 ผังงานอธิบายขั้นตอนการทำงานของอัลกอริทึมการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค

2.4 ค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน

ค่าสหสัมพันธ์ (Correlation) เป็นการศึกษาความสัมพันธ์ระหว่างตัวแปรหรือชุดข้อมูล 2 ชุดขึ้นไป ตัวอย่างการศึกษาความสัมพันธ์ เช่น ความสัมพันธ์ระหว่างส่วนสูงและน้ำหนัก ความสัมพันธ์ระหว่างระดับความแรงลมและอุณหภูมิ เป็นต้น ลักษณะที่แตกต่างกันของค่าสัมประสิทธิ์สหสัมพันธ์จะสื่อถึงทิศทางความสัมพันธ์ของข้อมูล [20] หากค่าสัมประสิทธิ์สหสัมพันธ์เป็น + จะสื่อถึงความสัมพันธ์ของตัวแปรทั้งสองมีทิศทางไปในทิศทางเดียวกัน หรือหากค่าเป็น - จะสื่อถึงความสัมพันธ์ของตัวแปรทั้งสองมีทิศทางตรงข้ามกัน การพิจารณาความสัมพันธ์ระหว่างตัวแปรว่ามีมากน้อยเพียงใดนั้นจะใช้วิธีคำนวณค่าสัมประสิทธิ์สหสัมพันธ์แบบต่าง ๆ เช่น ค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน (Pearson correlation) ค่าสัมประสิทธิ์สหสัมพันธ์แบบสเปียร์แมน (Spearman rank correlation) ค่าสัมประสิทธิ์สหสัมพันธ์แบบเตตราคอห์ริค (Tetrachoric correlation) เป็นต้น

ค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน (Pearson correlation coefficient) ถูกนำเสนอโดย คาร์ล เพียร์สัน (Karl Pearson) เป็นค่าที่วัดความสัมพันธ์ระหว่างตัวแปรสองตัวโดยใช้สัญลักษณ์เป็น R_{xy} ค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันเป็นวิธีที่ใช้วัดความสัมพันธ์ระหว่างตัวแปร หรือข้อมูล 2 ชุด โดยที่ตัวแปรหรือข้อมูล 2 ชุดนั้นควรจะต้องอยู่ในรูปของมาตราข้อมูลแบบอันตรภาคหรืออัตราส่วน (Interval หรือ Ratio scale) ซึ่งค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันสามารถคำนวณจากสมการที่ 2.4 ดังนี้

$$R_{xy} = \frac{N\sum XY - (\sum X)(\sum Y)}{\sqrt{[N\sum X^2 - (\sum X)^2][N\sum Y^2 - (\sum Y)^2]}} \qquad (2.4)$$

เมื่อ R_{xy} เป็น ค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน

$\sum X$ เป็น ผลรวมของข้อมูลที่วัดได้จากตัวแปรตัวที่ 1

$\sum Y$ เป็น ผลรวมของข้อมูลที่วัดได้จากตัวแปรตัวที่ 2

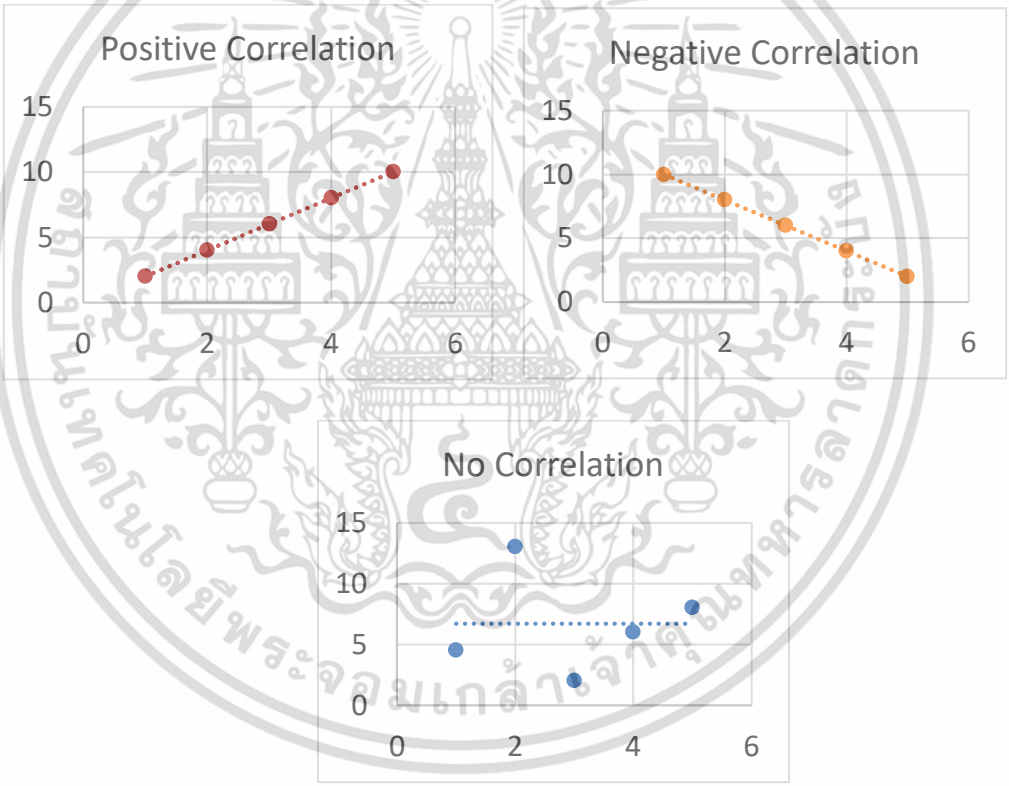
$\sum XY$ เป็น ผลรวมของผลคูณระหว่างข้อมูลตัวแปรที่ 1 และ 2

$\sum X^2$ เป็น ผลรวมของกำลังสองของข้อมูลที่วัดได้จากตัวแปรตัวที่ 1

$\sum Y^2$ เป็น ผลรวมของกำลังสองของข้อมูลที่วัดได้จากตัวแปรตัวที่ 2

N เป็น ขนาดของกลุ่มตัวอย่าง

ค่าที่ออกมาจะสรุปได้ว่าตัวแปรคู่ใดมีความสัมพันธ์กันหรือไม่ และมีค่าความสัมพันธ์กันมากน้อยเพียงใด ซึ่งการพิจารณาจะมองในแง่ของความเกี่ยวพัน ความสอดคล้องของข้อมูลว่าไปในทิศทางเดียวกันหรือไปในทิศทางตรงกันข้าม แต่ค่าที่คำนวณออกมาไม่ได้หมายความว่าตัวแปรหนึ่งเป็นเหตุและอีกตัวแปรเป็นผลของกันและกัน โดยค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันจะมีค่าอยู่ระหว่าง -1 ถึง 1 ช่วงของค่าจะสื่อถึงลักษณะของความสัมพันธ์แบ่งเป็น 3 ลักษณะคือ หากค่าเป็น + จะสื่อถึงความสัมพันธ์ของข้อมูลทั้งสองมีทิศทางไปในทิศทางเดียวกัน หากค่าเป็น - จะสื่อถึงความสัมพันธ์ของข้อมูลทั้งสองมีทิศทางตรงกันข้าม และหากค่าเป็น 0 จะสื่อถึงข้อมูลทั้งสองไม่มีความสัมพันธ์กัน ซึ่งสามารถแสดงกราฟความสัมพันธ์ในรูปแบบต่าง ๆ ของค่าสัมประสิทธิ์สหสัมพันธ์ได้ดังรูปที่ 2.6



รูปที่ 2.6 แสดงกราฟความสัมพันธ์ในรูปแบบต่าง ๆ ของค่าสัมประสิทธิ์สหสัมพันธ์

ในงานวิจัยของ Syed [21] ได้กล่าวถึงการนำเอาอัลกอริทึมต่าง ๆ มาใช้ในการหาค่าน้ำหนักของแต่ละคุณลักษณะเพื่อใช้ร่วมกับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว หนึ่งในวิธีที่ถูกกล่าวถึงคือค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน (Pearson Correlation Coefficient) ซึ่งถูกมาใช้ในการหาความสัมพันธ์ของคุณลักษณะของข้อมูลกับคลาสของคำตอบของข้อมูลเพื่อกำหนดค่าน้ำหนักให้กับแต่

ละคุณลักษณะ การคำนวณค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันเพื่อหาความสัมพันธ์ของข้อมูล 2 ชุดข้อมูลสามารถแสดงตัวอย่างได้จากค่าในตารางที่ 2.2

ตารางที่ 2.2 แสดงตัวอย่างข้อมูลที่ใช้ในการคำนวณค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน

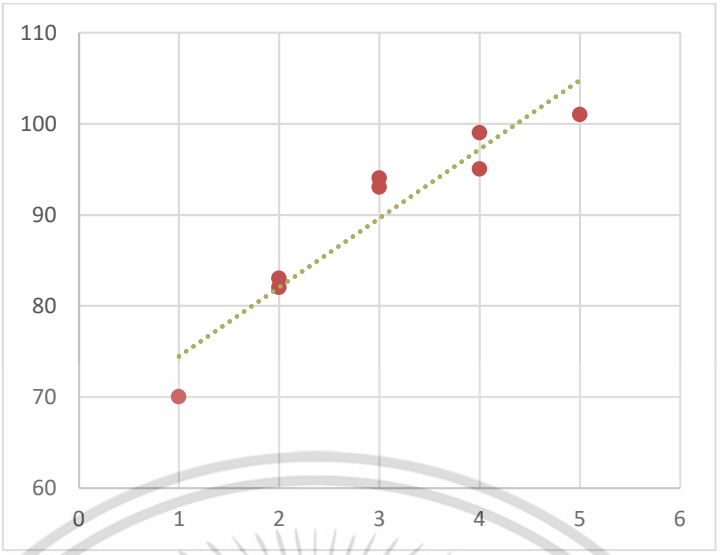
X	Y
3	94
2	82
4	99
2	83
1	70
4	95
5	101
3	93

จากค่าในตารางที่ 2.2 เราสามารถคำนวณค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันได้โดยนำเอาข้อมูลมาแทนค่าในสมการที่ 2.4 และเมื่อนำมาหาค่าความสัมพันธ์จะได้ลักษณะของกราฟดังรูปที่ 2.7 ดังนี้

$$R_{xy} = \frac{N\sum XY - (\sum X)(\sum Y)}{\sqrt{[N\sum X^2 - (\sum X)^2][N\sum Y^2 - (\sum Y)^2]}}$$

$$R_{xy} = \frac{(17936 - 17208)}{\sqrt{[84 - (24)^2][65025 - (717)^2]}}$$

$$R_{xy} = 0.95$$



รูปที่ 2.7 แสดงกราฟความสัมพันธ์ของชุดข้อมูล X และชุดข้อมูล Y จากตัวอย่าง

บทที่ 3
งานวิจัยที่เกี่ยวข้อง

3.1 อัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนักและการประยุกต์ใช้กับข้อมูลสาธารณะ UCI (Weighted-KNN and its application on UCI)

ในงานวิจัยชิ้นนี้ผู้วิจัยได้นำเสนออัลกอริทึมการให้น้ำหนักกับคุณลักษณะโดยใช้อัตราความผิดพลาดของการจำแนกเพื่อวัดความสำคัญของคุณลักษณะ หากนำคุณลักษณะที่สำคัญออกจากข้อมูลการเรียนรู้อัตราความผิดพลาดย่อมมีความน่าจะเป็นที่จะเพิ่มขึ้น ในทางตรงกันข้ามหากนำคุณลักษณะที่ไม่เกี่ยวข้องออกไปย่อมจะลดอัตราความผิดพลาดลง ซึ่งเมื่อทราบถึงความสำคัญของคุณลักษณะทำให้อัลกอริทึมสามารถลดการนำคุณลักษณะที่ไม่เกี่ยวข้องมาใช้ในการคำนวณ [6] ผู้วิจัยได้นำเสนอสมการที่ใช้ในการคำนวณระยะทางในอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนักดังสมการที่ 3.1

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2 \ . \ W_i} \qquad (3.1)$$

โดยที่ W_i คือน้ำหนักของคุณลักษณะที่ i

p_i คือค่าในคุณลักษณะที่ i ของจุด p

q_i คือค่าในคุณลักษณะที่ i ของจุด q

n คือจำนวนมิติของข้อมูล

โดยสามารถอธิบายขั้นตอนการทำงานของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนัก (W-KNN) ได้ดังนี้

ขั้นตอนที่ 1 การกำหนดค่าน้ำหนักของแต่ละคุณลักษณะซึ่งจะสามารถแบ่งเป็นขั้นตอนได้ทั้งหมด 4 ขั้นตอนดังนี้

1) จำแนกประเภทชุดข้อมูลทดสอบด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวและเก็บค่าผลลัพธ์ของการจำแนกประเภท ซึ่งจะเป็นค่าผิดพลาดของการจำแนกทางสถิติ (statistical classification errors) กำหนดให้เป็นค่า V

2) ทำการตัดเอาคอลัมน์ของแต่ละคุณลักษณะทั้งในชุดข้อมูลเรียนรู้และชุดข้อมูลทดสอบออกตามลำดับ ซึ่งในการตัดแต่ละคอลัมน์ของแต่ละคุณลักษณะออก อัลกอริทึมจะทำจำแนกประเภทด้วยชุดข้อมูลทดสอบด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว และเก็บค่าผลลัพธ์การจำแนกประเภทซึ่งจะเป็นค่าผิดพลาดของการจำแนกทางสถิติเมื่อตัดคุณลักษณะแต่ละตัวออก กำหนดให้เป็นค่า $V1(k)$ โดยแต่ละ k = 1, 2, 3, 4, ... , n ซึ่ง n คือจำนวนคุณลักษณะ

3) คำนวณค่าน้ำหนักของแต่ละ $W(k)$ โดยสมการที่ 3.2 ซึ่งจะกำหนดให้

$$W(k) = \frac{V1(k)}{V} \qquad (3.2)$$

โดย k = 1, 2, 3, ... , n ซึ่งหากค่า $V1(k)$ = 0 กำหนดให้ค่า $W(k)$ = 1 และถ้า V = 0 แล้วกำหนดให้ค่า $W(k)$ = 1 โดยหมายความว่าน้ำหนักของทุกคุณลักษณะมีค่าเท่ากัน

4) ได้ค่าน้ำหนักของแต่ละคุณลักษณะซึ่งมีผลรวมของค่าน้ำหนักเป็น 1

ขั้นตอนที่ 2

แทนที่ค่าน้ำหนัก W ของแต่ละคุณลักษณะในสมการที่ (3.1) จากนั้นทำการคำนวณค่าระยะทาง

ขั้นตอนที่ 3

ใช้ระยะทางใหม่ที่ได้จากการคำนวณร่วมกับค่าน้ำหนักเพื่อค้นหาเพื่อนบ้านที่ใกล้ที่สุดจำนวน K ตัวแล้วจำแนกประเภทของข้อมูลตามคำตอบส่วนใหญ่ของเพื่อนบ้านที่ใกล้ที่สุดเหล่านั้น

3.2 การปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยผลสนับสนุนจากคลาสคำตอบและการให้น้ำหนักกับคุณลักษณะ (An Improved kNN Based on Class Contribution and Feature Weighting)

ในงานวิจัยชิ้นนี้ของ Huang และคณะ [7] ได้ค้นหาความสำคัญของแต่ละคุณลักษณะโดยการวัดความถูกต้องจากการนำเอาคุณลักษณะแต่ละประเภทออกในแต่ละรอบการจำแนก หากนำคุณลักษณะที่ไม่สำคัญออกจากข้อมูลการเรียนรู้ความแม่นยำย่อมมีความน่าจะเป็นที่จะเพิ่มขึ้น ในทางตรงกันข้ามหากนำคุณลักษณะที่เกี่ยวข้องออกไปย่อมจะลดความแม่นยำของอัลกอริทึมลง ซึ่งเมื่อทราบถึงความสำคัญของคุณลักษณะทำให้อัลกอริทึมสามารถลดอิทธิพลของการนำคุณลักษณะที่ไม่เกี่ยวข้องมาใช้ในการคำนวณ ผู้วิจัยได้นำเสนอสมการที่ใช้ในการคำนวณระยะทางในอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนักดังสมการที่ 3.1 แต่เนื่องจากค่าที่ใช้วัดความสำคัญของคุณลักษณะนั้นเป็นค่าความถูกต้อง หากนำเอาคุณลักษณะที่ไม่เกี่ยวข้องออกค่าความสำคัญของคุณลักษณะนั้นควรจะมีค่าลดลงอย่างสอดคล้องกัน โดยค่าความถูกต้องนั้นคำนวณได้ดังสมการที่ 3.3

$$Disc_i = 1 - (pre_i - pre_t) \tag{3.3}$$

โดย $Disc_i$ คือค่าความสำคัญของคุณลักษณะที่ i

pre_t คือค่าความแม่นยำเฉลี่ยของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมด้วยค่า K เป็น 3 5 และ 7

pre_i คือค่าความแม่นยำเฉลี่ยของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมเมื่อตัดคุณลักษณะที่ i ออก ด้วยค่า K เป็น 3 5 และ 7

จากสมการที่ 3.3 สามารถอธิบายวิธีการพิจารณาค่าของความสำคัญของคุณลักษณะที่ i ได้ดังนี้ โดยค่าของความสำคัญของคุณลักษณะที่ i จะมีค่าความสำคัญน้อยกว่า 1 หรือ $Disc_i < 1$ ก็ต่อเมื่อผลลัพธ์ความแม่นยำของการจำแนกประเภทเมื่อเมื่อตัดคุณลักษณะที่ i ออก หรือค่า pre_i นั้นมีค่ามากกว่า pre_t ซึ่งหมายความว่าหากตัดคุณลักษณะที่ i ออกไปจะส่งผลให้ค่าความถูกต้องมากกว่าการมีอยู่ของคุณลักษณะที่ i นั่นเอง ในทางตรงกันข้ามหากค่าของความสำคัญของคุณลักษณะที่ i มีค่าความสำคัญมากกว่า 1 หรือ $Disc_i > 1$ ก็ต่อเมื่อผลลัพธ์ความแม่นยำของการจำแนกประเภทเมื่อเมื่อตัดคุณลักษณะที่ i ออก หรือค่า pre_i นั้นมีค่าน้อยกว่า pre_t ซึ่งหมายความว่าหากตัด

คุณลักษณะที่ i ออกไปจะส่งผลให้ค่าความถูกต้องน้อยกว่าการมีอยู่ของคุณลักษณะที่ i ในงานวิจัยชิ้นนี้ Huang และคณะได้ระบุขั้นตอนการทำงานของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนัก (W-KNN) ออกเป็น 4 ขั้นตอน ซึ่งสามารถอธิบายได้ดังนี้

ขั้นตอนที่ 1. ทำการตัดเอาคอลัมน์ของแต่ละคุณลักษณะทั้งในชุดข้อมูลเรียนรู้และชุดข้อมูลทดสอบออกตามลำดับ ซึ่งในแต่ละการตัดคอลัมน์ของแต่ละคุณลักษณะจะทำจำแนกประเภทด้วยชุดข้อมูลทดสอบด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว จากนั้นทำการแทนค่าลงในสมการที่ 3.3 เพื่อคำนวณค่าความสำคัญของแต่ละคุณลักษณะ

ขั้นตอนที่ 2.คำนวณค่าน้ำหนักจากค่าที่คำนวณได้ในขั้นตอนที่ 1 โดยนำค่าค่าความสำคัญของแต่ละคุณลักษณะมาปรับค่าเป็นมาตรฐานเพื่อใช้เป็นค่าน้ำหนักตามสมการที่ 3.4

$$Disc_i = \frac{Disc_i}{\sum_{i=1}^{n} Disc_i} \tag{3.4}$$

ขั้นตอนที่ 3. คำนวณค่าระยะทางจากอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนัก (W-KNN) และเก็บผลลัพธ์เพื่อนบ้านใกล้ที่สุดจำนวน K ตัวนั้น

ขั้นตอนที่ 4. นับจำนวนของคำตอบเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวนั้นและคำนวณระยะทางเฉลี่ยของคำตอบแต่ละประเภทและแทนค่าโดยใช้สมการที่ 3.5 ในการหาค่าการสนับสนุนของแต่ละคลาส

$$CT_j = \frac{K}{N_j} + \frac{1}{N_j} \sum d(X, Y_j) \tag{3.5}$$

โดย CT_j คือการสนับสนุนของคลาส j

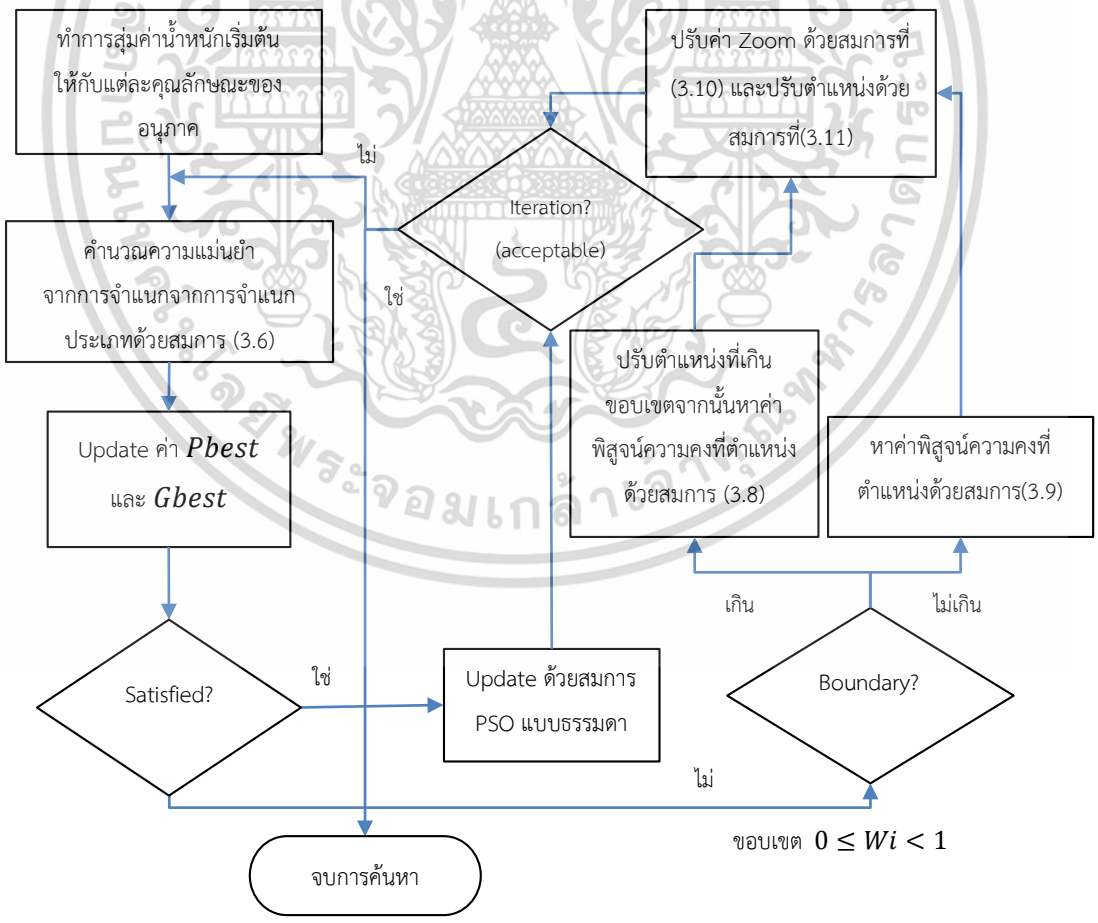
K คือจำนวนของเพื่อนบ้านใกล้ที่สุด

N_j คือจำนวนของเพื่อนบ้านใกล้ที่สุดที่มีคำตอบเป็นคลาส j

จะเห็นได้ว่าหากจำนวนของเพื่อนบ้านใกล้ที่สุดที่มีคำตอบแต่ละคลาสมีจำนวนมากก็จะส่งผลให้ค่าการสนับสนุนของคลาสนั้นน้อยลงตามไปด้วย ซึ่งหลังจากได้ผลรวมค่าการสนับสนุนของแต่ละคลาสในเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวนั้น อัลกอริทึมก็จะทำการกำหนดคลาสคำตอบจากค่าการสนับสนุนของคลาสที่มีค่าน้อยที่สุด

3.3 วิธีการปรับมาตรฐานเพื่อหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคและการประยุกต์ใช้ในปัญหาการให้น้ำหนักกับคุณลักษณะ (The Normalized-PSO and Its Application in Attribute Weighted Optimal Problem)

ในงานวิจัยชิ้นนี้ของ Guo และคณะ [15] ได้นำวิธีการให้น้ำหนักของคุณลักษณะมาใช้ร่วมกับการจำแนกประเภทแบบอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว โดยทำการปรับค่าน้ำหนักของคุณลักษณะโดยละเอียดด้วยวิธีการปรับมาตรฐานเพื่อหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค [16] เพื่อนำไปประยุกต์ใช้ในการให้น้ำหนักกับแต่ละคุณลักษณะเพื่อให้ได้ค่าน้ำหนักที่เหมาะสม (The Normalized-PSO and Its Application in Attribute Weighted Optimal Problem หรือ Weight Norm-PSO) ในขั้นตอนของการวนซ้ำเพื่อค้นหาจะมีการตรวจสอบค่าตำแหน่งของอนุภาคในการปรับแต่ละครั้งว่าอยู่ในเงื่อนความพอใจ (satisfy) หรือไม่ ซึ่งจะส่งผลให้ค่าน้ำหนักที่ระบุถึงความสำคัญของคุณลักษณะมีค่าที่เหมาะสม โดยขั้นตอนการทำงานของอัลกอริทึม Weight Norm-PSO สามารถอธิบายได้ดังรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการหาค่าน้ำหนักที่เหมาะสมของอัลกอริทึม Weight Norm-PSO

ซึ่งจากรูปที่ 3.1 สามารถอธิบายขั้นตอนการทำงานของอัลกอริทึม Weight Norm-PSO ได้ดังนี้

ขั้นตอนที่ 1 กำหนดค่าเริ่มต้นให้กับแต่ละอนุภาค โดยแต่ละอนุภาคก็คือชุดข้อมูลในการเรียนรู้และค่าเริ่มต้นหรือตำแหน่งเริ่มต้นก็คือค่าน้ำหนักของแต่ละคุณสมบัติ ซึ่งจะถูกกำหนดด้วยค่าสุ่มระหว่าง 0 ถึง 1

ขั้นตอนที่ 2 ทำการประเมินด้วยฟังก์ชันความเหมาะสม (fitness function) สำหรับแต่ละตำแหน่งปัจจุบันของแต่ละอนุภาคด้วยค่าความแม่นยำจากการประเมินเมื่อนำเอาวิธีการมาใช้กับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนักซึ่งนำเอาการคำนวณระยะทางแบบยุคลิดมาตรฐานมาใช้งานด้วยสมการที่ 3.6 ดังนี้

$$d(p,q) = \sqrt{\sum_{i=1}^{n} \left(\frac{(p_i - q_i)}{Si} \right)^2 . W_i} \qquad (3.6)$$

จากนั้นใช้การประเมินในการคำนวณของแต่ละอนุภาค คำนวณตำแหน่งที่ดีที่สุดของแต่ละอนุภาค ($Pbest$) และคำนวณตำแหน่งที่ดีที่สุดของกลุ่มทั้งหมด ($Gbest$)

ขั้นตอนที่ 3 ตรวจสอบว่าผลของการประเมินนั้นได้ค่าเหมาะสมที่สุด (optimum) หรือไม่ จากนั้นตรวจสอบเกณฑ์ในการหยุดการค้นหา หากยังไม่ถึงรอบการหยุดอัลกอริทึมจะทำการคำนวณค่า S_i ของแต่ละอนุภาค โดยกำหนดให้ค่า S_i มีค่าเท่ากับผลรวมของค่าน้ำหนัก (ตำแหน่งของอนุภาค) ดังสมการที่ 3.7

$$S_i = \sum_{p=1}^{n} x_p \qquad (3.7)$$

ซึ่งในขั้นตอนต่อมาอัลกอริทึมจะทำการพิจารณาว่าอนุภาคแต่ละตัวนั้นมีค่ามีความพอใจตามที่กำหนดหรือไม่ โดยการประเมินค่านั้นว่ามีความพอใจหรือไม่จะพิจารณาจากเงื่อนไขต่อไปนี้

$$satisfy \begin{cases} s_i = 1 \\ x_p \geq 0 \\ x_p < 1 \end{cases}$$

ซึ่งหากเข้าเงื่อนไขอัลกอริทึมจะทำการปรับค่าของความเร็วและตำแหน่ง (น้ำหนัก) โดยใช้สมการของความเร็วและตำแหน่งและวนกลับไปพิจารณาในขั้นตอนที่ 2 แต่หากไม่อยู่ในเงื่อนความพอใจ ก็จะไปยังขั้นตอนที่ 4 ต่อ

ขั้นตอนที่ 4 ในขั้นตอนนี้จะเป็นการปรับแบบวิธีมาตรฐานของวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคโดยในขั้นตอนแรกจะทำการพิจารณาว่าเหตุผลที่ไม่เข้าเงื่อนไขความพอใจว่าอยู่ในกรณีใดซึ่งจะทำการตรวจสอบว่าค่าตำแหน่งนั้นเกินขอบเขตหรือไม่โดย $x_p \leq 0$ และ $x_p > 1$ ซึ่งอัลกอริทึมจะปรับค่าตำแหน่งที่ไม่เข้าเงื่อนไขโดย

หาก $x_p \leq 0$ กำหนดให้ค่า

$$x_p = \frac{1}{n} * r_p$$ โดย r_p เป็นค่าระหว่าง 0 ถึง 1

และหาก $x_p > 1$ กำหนดให้ค่า

$$x_p = \frac{x_p}{(x_p + \frac{1}{n})}$$

ขั้นต่อมาจะทำการหาค่าพิสูจน์ความคงที่ของตำแหน่ง (proved stable position) ในวิธีมาตรฐาน PSO (ของ Weight Norm-PSO) ด้วยสมการที่ 3.8 ดังนี้

$$Pa_i = Gbest * r_1 + r_2 \quad (3.8)$$

โดย r_1 เป็นค่าระหว่าง 0.5 ถึง 1.5 และ r_2 เป็นค่าระหว่าง -0.5 ถึง 1.5

หากตรวจสอบว่าค่าตำแหน่งนั้นไม่เกินขอบเขต อัลกอริทึมจะทำการหาค่าศูนย์กลางการกระจายพื้นที่ค้นหาด้วยสมการที่ 3.9 ดังนี้

$$Pa_i = (Pbest + (Gbest - Pbest) * r_3) * \frac{1}{2} \quad (3.9)$$

โดย r_3 เป็นค่าระหว่าง 0 ถึง 1

จากนั้นจะทำการหาค่าศูนย์กลางการกระจายพื้นที่ค้นหา (space zoom center) ของทั้งสองเงื่อนไขทั้งในขอบเขตและเกินขอบเขตด้วยสมการที่ 3.10

$$z = \frac{(\sum_{j=1}^{n} x_i^j(t+1) - \sum_{j=1}^{n} Pa_i^j)}{1 - \sum_{j=1}^{n} Pa_i^j} \quad (3.10)$$

โดย $x_i^j(t+1)$ หมายถึงค่าตำแหน่งของอนุภาค i นั้นส่วน Pa_i^j หมายถึงค่าพิสูจน์ความคงที่ของตำแหน่งของอนุภาคนั้นจากนั้นนำค่าศูนย์กลางการกระจายพื้นที่ค้นหาไปใช้ในการหาค่าตำแหน่งใหม่ด้วยสมการที่ 3.11 และเข้าไปปรับค่าตำแหน่งและความเร็วของอนุภาคในขั้นตอนที่ 2 ต่อไป

$$x_i(t+1) = \frac{(x_i(t+1) - P_a)}{z} + P_a \quad (3.11)$$

3.4 การจำแนกประเภทโดยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยการหาค่าน้ำหนักเหมาะสมที่สุดแบบกลุ่มอนุภาคจากการเรียนรู้คุณลักษณะ (A KNN Classifier with PSO Feature Weight Learning Ensemble)

ในงานชิ้นนี้ [17] คณะผู้วิจัยได้นำเสนออัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยการหาค่าน้ำหนักเหมาะสมที่สุดแบบกลุ่มอนุภาคจากการเรียนรู้คุณลักษณะเพื่อเป็นการพัฒนาความสามารถของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม ซึ่ง Cao และ Liu ได้ใช้อัลกอริทึมแบบคลาวด์ย้อนกลับเพื่อจับคู่ข้อมูลการเรียนรู้ไปยังตัวเก็บข้อมูลแบบคลาวด์ อัลกอริทึมแบบคลาวด์ย้อนกลับมีความสามารถในการจัดการกับผลกระทบของข้อมูลรบกวน (Noisy data) เมื่อนำมาจำแนกประเภทได้อย่างมีประสิทธิภาพ การทำงานของอัลกอริทึมจะทำการจับคู่แต่ละคุณลักษณะกับคลาวด์เวกเตอร์ โดยอัลกอริทึมใช้การเปรียบเทียบความคล้ายคลึงกันของเวกเตอร์คลาวด์เป็นฟังก์ชันความเหมาะสม (fitness function) เพื่อค้นหาค่าน้ำหนักของคุณลักษณะที่เหมาะสม จากการศึกษาแนวทางในการพัฒนาอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวของงานวิจัยส่วนใหญ่มีเป้าหมายที่จะปรับปรุงความแม่นยำในการจำแนกประเภทของอัลกอริทึมแบบดั้งเดิมให้สูงขึ้น เช่นเดียวกันกับงานวิจัยชิ้นนี้ซึ่งผู้วิจัยได้กล่าวถึงแนวทางในการพัฒนาอัลกอริทึมโดยสามารถแบ่งออกเป็น 3 แนวทางดังนี้

1. ข้อมูลในปัจจุบันเมื่อนำมาใช้ในการเรียนรู้เลยโดยไม่ทำการเตรียมจะมีข้อมูลรบกวนซึ่งส่งผลต่อประสิทธิภาพในการนำมาใช้งาน วิธีในการจัดการกับข้อมูลรบกวนได้ถูกนำเสนอหลากหลายวิธีโดยวิธีการที่น่าสนใจวิธีการหนึ่งคืออัลกอริทึมแบบคลาวด์ การทำงานของอัลกอริทึมจะทำการจำลองการเปลี่ยนแปลงที่ไม่แน่นอนระหว่างแนวคิดเชิงคุณภาพและคำอธิบายเชิงปริมาณของข้อมูลที่จะนำมาประมวลผล อัลกอริทึมแบบคลาวด์เป็นเครื่องมือที่มีประสิทธิภาพในการศึกษาคุณลักษณะของข้อมูลที่มีความคลุมเครือ และสามารถจัดการกับผลกระทบของข้อมูลรบกวนซึ่งเป็นปัญหาในการศึกษาข้อมูล โดยอัลกอริทึมแบบคลาวด์เป็นวิธีการที่สามารถนำข้อมูลในลักษณะดังกล่าวมาใช้ในศึกษาได้โดยไม่จำเป็นต้องทำการเตรียมข้อมูลก่อนนำมาประมวลผล (Data Preprocessing)
2. ในการจำแนกประเภทของข้อมูล หากวิธีการที่ใช้พิจารณาค่าความสำคัญของคุณลักษณะสำหรับแต่ละคุณลักษณะเท่ากันอาจจะส่งผลให้เกิดความผิดพลาดในการจำแนกประเภทได้ ในงานวิจัยชิ้นนี้ได้มีแนวคิดในการคำนวณความคล้ายคลึงกันของอัลกอริทึมแบบคลาวด์จากการจับคู่กับคุณลักษณะเพื่อนำค่าที่คำนวณมาใช้เพื่อกำหนดค่าน้ำหนักของแต่ละคุณลักษณะ
3. แนวทางการปรับปรุงอัลกอริทึม นอกจากจะกำหนดค่าน้ำหนักให้แต่ละคุณลักษณะด้วยอัลกอริทึมแบบคลาวด์ ผู้วิจัยยังได้นำวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (PSO และกำหนดฟังก์ชันความเหมาะสม (fitness function) ซึ่งจะทำให้ได้ค่าน้ำหนักของคุณลักษณะ

ที่เหมาะสม โดยใช้ค่าคุณลักษณะที่ผ่านกระบวนการในเมทริกซ์ของอัลกอริทึมแบบคลาวด์เป็นค่าเริ่มต้นสำหรับค้นหาค่าน้ำหนักของคุณลักษณะในการจำแนกประเภท

จากแนวทางในการปรับปรุงอัลกอริทึมแบบดั้งเดิมที่กล่าวมา ผู้วิจัยจึงได้นำเสนออัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยการหาค่าน้ำหนักเหมาะสมที่สุดแบบกลุ่มอนุภาคจากการเรียนรู้คุณลักษณะ การทำงานของอัลกอริทึมสามารถแบ่งออกเป็น 3 ส่วนตามแนวทางในการปรับปรุงคือ 1. การหารูปแบบของชุดข้อมูลการเรียนรู้ ซึ่งในอัลกอริทึมที่นำเสนอใช้อัลกอริทึมแบบคลาวด์เพื่อแสดงข้อมูลคุณลักษณะจากการจับคู่ในโครงสร้างของคลาวด์ 2. อัลกอริทึมที่นำเสนอจะใช้การคำนวณความคล้ายคลึงกันของอัลกอริทึมแบบคลาวด์เพื่อวัดความสำคัญของคุณลักษณะ โดยหากค่าคุณลักษณะเมื่อเปรียบเทียบแล้วส่งผลเช่นเดียวกันในทุกคลาสคำตอบแสดงว่าคุณลักษณะนั้นไม่มีผลสำหรับการจำแนกประเภท 3. การกำหนดฟังก์ชันความเหมาะสมในวิธีการ PSO มีจุดมุ่งหมายคือการค้นหาค่าความคล้ายคลึงกันที่น้อยที่สุดของแต่ละคุณลักษณะในชุดข้อมูลการเรียนรู้ ซึ่งวิธีการ PSO จะกำหนดค่าน้ำหนักที่เหมาะสมให้แต่ละคุณลักษณะ อัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยการหาค่าน้ำหนักที่เหมาะสมที่สุดแบบกลุ่มอนุภาคจากการเรียนรู้คุณลักษณะได้นำเอาวิธีการต่าง ๆ มาใช้ร่วมกับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว ซึ่งสามารถอธิบายแต่ละวิธีการได้ดังนี้

1. **การจับคู่คุณลักษณะ (Feature Mapping)** ในชุดข้อมูลที่นำมาศึกษาแต่ละชุดข้อมูลจะประกอบด้วยคุณลักษณะของข้อมูลซึ่งประกอบด้วยข้อมูลจำนวนมาก เมื่อชุดข้อมูลเหล่านั้นมีข้อมูลรบกวนจะเป็นอุปสรรคสำหรับการวัดความคล้ายคลึงกันของข้อมูลเพื่อกำหนดค่าน้ำหนักให้กับแต่กับคุณลักษณะ และหากทำการค้นหาค่าน้ำหนักโดยตรงจากข้อมูลก็จะมีความผิดพลาดเกิดขึ้น ในอัลกอริทึมที่นำเสนอได้นำเอาอัลกอริทึมแบบคลาวด์เพื่อวิเคราะห์ชุดข้อมูลการเรียนรู้โดยใช้อัลกอริทึมระบบคลาวด์แบบย้อนกลับซึ่งสามารถจัดการกับผลกระทบของข้อมูลรบกวนและแปลงข้อมูลเป็นพารามิเตอร์เพื่อใช้ในอัลกอริทึมแบบคลาวด์

   อัลกอริทึมแบบคลาวด์ย้อนกลับจะทำการวิเคราะห์ชุดข้อมูลการเรียนรู้โดยกำหนดการกระจายของข้อมูลเป็นการแจกแจงแบบปกติ (Normal distribution) เพื่อให้สามารถคำนวณค่าไฮเปอร์เอนโทรปีหรือ He ได้ ลักษณะของอะตอมของอัลกอริทึมแบบคลาวด์จะอธิบายถึงข้อมูลจากรูปแบบการแจกแจงข้อมูลนั้นด้วย $Cloud(Ex, En, He)$ ซึ่งประกอบด้วยค่าหยดน้ำ (drop) จำนวน N ค่า โดยที่ Ex คือตำแหน่งที่ตรงกับศูนย์กลางจุดถ่วงของคลาวด์ ซึ่งองค์ประกอบของคลาวด์นั้นจะประกอบด้วยข้อมูลที่สามารถนำไปประมวลได้ En คือการวัดความครอบคลุมของแนวคิด เช่น การวัดความแปรปรวนของข้อมูลว่ามีข้อมูลที่สามารถนำไปประมวลผลได้จำนวนเท่าใดและ He เป็นหน่วยวัดการกระจายตัวของค่าหยดน้ำในคลาวด์หรือค่าเอนโทรปีของ En การจับคู่เวกเตอร์คุณลักษณะของตัวอย่างในการจำแนกประเภทคลาวด์ (CC) แบ่งออกเป็นสองขั้นตอนดังนี้

ขั้นตอนที่ 1 คำนวณ Ex และ En โดยใช้ค่าในแต่ละคุณลักษณะของแต่ละคลาสคำตอบเป็นพารามิเตอร์ของ $Cloud(Ex, En, He)$

ขั้นตอนที่ 2 กำหนดค่า He หาก $S^2 - En^2 \geq 0$ แล้ว $He = \sqrt{S^2 - En^2}$ ถ้าไม่กำหนดให้ $He = 0.98En$

จากขั้นตอนการจับคู่คุณลักษณะของข้อมูลไปยังคลาวด์สำหรับข้อมูล เมื่อข้อมูลประกอบด้วยเวกเตอร์ C ซึ่งแสดงคลาสคำตอบโดยที่ $C = \{ c_1, c_2, \ldots, c_m \}$ และเวกเตอร์ F โดยที่ $F = \{ F_1, F_2, \ldots, F_n \}$ ซึ่งแสดงถึงคุณลักษณะของข้อมูลเป็นเวกเตอร์ขนาด m เมื่อจับคู่ด้วยคลาวด์จะได้ $F = \{(Ex_{i1}, En_{i1}, He_{i1}), \ldots, (Ex_{ic}, En_{ic}, He_{im})\}$ หลังจากการจับคู่คุณลักษณะของชุดข้อมูลทั้งหมดจะได้คลาวด์เมทริกซ์ (Cloud Matrix) โดยที่แต่ละคอลัมน์ของเมทริกซ์คือการจับคู่ของแต่ละคุณลักษณะกับอัลกอริทึมแบบคลาวด์

$$CloudMatrix_{m\times n} = \begin{bmatrix} CC_{11} & CC_{12} & \cdots & CC_{1n} \\ CC_{21} & CC_{21} & \cdots & CC_{2n} \\ & \cdots & \cdots & \\ CC_{m1} & CC_{m1} & \cdots & CC_{mn} \end{bmatrix}$$

2. **ความคล้ายคลึงกันของคลาวด์ (Cloud Similarity)** ในการจำแนกประเภทหากแต่ละคลาสคำตอบของคุณลักษณะเหล่านั้นแยกออกจากกันอย่างสมบูรณ์อัลกอริทึมจะสามารถจำแนกประเภทได้ดีด้วยคุณลักษณะตัว ในทางตรงกันข้ามหากมีการทับซ้อนกันอย่างสมบูรณ์คุณลักษณะนั้นจะแทบไม่มีผลต่อการจำแนกประเภท วิธีการของอัลกอริทึมแบบคลาวด์สามารถนำเอาคุณลักษณะของข้อมูลจับคู่เป็นเวกเตอร์คลาวด์ เพื่อใช้คำนวณความคล้ายคลึงกันของเวกเตอร์และทำให้สามารถนำค่านั้นกำหนดเป็นค่าน้ำหนักของคุณลักษณะได้ จากการศึกษาความคล้ายคลึงกันของอัลกอริทึมแบบคลาวด์จะขึ้นอยู่กับความคล้ายคลึงกันของความหมายและความคล้ายคลึงกันของส่วนขยาย โดยความคล้ายคลึงกันของความหมายอธิบายถึงแกนกลางของระดับความคล้ายคลึงกันของสองข้อมูลซึ่งจะขึ้นอยู่กับค่า Ex และ En ส่วนความคล้ายคลึงกันของส่วนขยายอธิบายถึงระดับของการทับซ้อนของข้อมูลซึ่งขึ้นอยู่กับค่า En และ He ความคล้ายคลึงกันของความหมายจะถูกกำหนดเป็น α และความคล้ายคลึงกันของส่วนขยายจะถูกกำหนดเป็น β ในการคำนวณค่า α จะกำหนดค่าด้วยการคำนวณระยะทางระหว่าง 2 คลาวด์ ซึ่งสำหรับคลาวด์ $C_1(Ex_1, En_1, He_1)$ และ $C_2(Ex_2, En_2, He_2)$ ค่าระยะทาง $Dis(C_1, C_2)$ สามารถคำนวณได้ดังสมการที่ 3.12

$$Dis(C_1, C_2) = \begin{cases} 1 & |Ex_1 - Ex_2| > 3(En_1 + En_2) \\ \dfrac{|Ex_1 - Ex_2|}{3(En_1 + En_2)} & |Ex_1 - Ex_2| \leq 3(En_1 + En_2) \end{cases} \quad (3.12)$$

โดยเมื่อ $|Ex_1 - Ex_2| > 3(En_1 + En_2)$ จะหมายความว่าคลาวด์ทั้งสองไม่ทับซ้อนกัน ค่าระยะทางจะถูกกำหนดเป็น 1 ในทางตรงกันข้ามจะหมายความว่าแกนกลางของระดับความคล้ายคลึงกันของสองข้อมูลนั้นมีค่าใกล้เคียงกัน ซึ่งความคล้ายคลึงกันของข้อมูลคุณลักษณะต่อคลาสนี้ส่งผลกระทบอย่างมากต่อความแม่นยำของอัลกอริทึมในการจำแนกประเภท ค่าความคล้ายคลึงกันของความหมายสามารถคำนวณได้ดังสมการที่ 3.13 และค่าความคล้ายคลึงกันของส่วนขยายสามารถคำนวณได้ดังสมการที่ 3.14

$$\alpha = 1 - Dis(C_1, C_2) \quad (3.13)$$

$$\beta = e^{-\left|\left(\frac{He_1}{En_1}\right) - \left(\frac{He_2}{En_2}\right)\right|} \quad (3.14)$$

จากการคำนวณค่า He/En จะแสดงถึงระดับที่ไม่ต่อเนื่องของส่วนขยายแนวคิด เมื่อข้อมูลคลาวด์มีลักษณะคล้ายการแจกแจงแบบปกติ He/En จะมีแนวโน้มเป็น 0 หากค่า β เป็นค่าต่ำสุด $\beta = 0.375$ ซึ่งบ่งชี้ว่าคลาวด์ทั้งสองมีความแตกต่างกันมากที่สุดของค่าความคล้ายคลึงกันของส่วนขยาย ความคล้ายคลึงกันของคลาวด์จะขึ้นอยู่กับค่า α และ β ซึ่งค่า β จะทำการกำหนดระดับของผลกระทบต่อความคล้ายคลึงกันของคลาวด์ประมาณร้อยละ 10 โดยความคล้ายคลึงกันของคลาวด์สามารถคำนวณได้ดังสมการที่ 3.15

$$SIM(C_1, C_2) = \frac{\beta\alpha + b\alpha}{1 + b} \ , b = 5 \quad (3.15)$$

3. **ฟังก์ชันความเหมาะสมของวิธีการ PSO (Fitness Function For PSO)** อัลกอริทึมจะกำหนดเป้าหมายโดยต้องการให้คุณลักษณะทั้งหมดมีความคล้ายคลึงกันน้อยที่สุดสำหรับทุกคลาส คำตอบดังนั้นฟังก์ชันความเหมาะสมของวิธีการ PSO จึงสามารถกำหนดได้ดังสมการที่ 3.16

$$Fitness = \sum_{r=1}^{m} \frac{1}{m-1} \sum_{p=1, p \neq j}^{m} (1 - SIM(C_{ij}, C_{ip})) \quad (3.16)$$

การจำแนกประเภทโดยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยการหาค่าน้ำหนักเหมาะสมที่สุดแบบกลุ่มอนุภาคจากการเรียนรู้คุณลักษณะ จะคำนวณค่า w_{ij} ซึ่งจะอยู่ในรูปเมทริกซ์ของค่าน้ำหนักโดย w_{ij} หมายถึงน้ำหนักของคุณลักษณะ j สำหรับคลาสคำตอบ i ในการจับคู่คุณลักษณะของอัลกอริทึมแบบคลาวด์สำหรับแต่ละคุณลักษณะ K ผลลัพธ์ของการจับคู่จะเป็น $F = \{w_{1k}\ (Ex_{i1}, En_{i1}, He_{i1}), \ldots, w_{mk}(Ex_{ic}, En_{ic}, He_{im})\}$ โดยนำเอาค่าน้ำหนักไปคำนวณค่าความคล้ายคลึงกันของคลาวด์เพื่อหาค่าความคล้ายคลึงกันที่น้อยที่สุด การทำงานของอัลกอริทึม PFWKNN ในการจำแนกประเภทสามารถแบ่งเป็น 5 ขั้นตอนดังนี้

ขั้นตอนที่ 1 ทำการจับคู่แต่ละคุณลักษณะในชุดข้อมูลเรียนรู้ด้วยอัลกอริทึมแบบคลาวด์ย้อนกลับ

ขั้นตอนที่ 2 ให้ค่าเริ่มต้นกับเมทริกซ์ $W_{m \times n}$ ด้วยการสุ่มค่า [0,1] และนำเอาค่าน้ำหนักไปคำนวณค่าความคล้ายคลึงกันของคลาวด์

ขั้นตอนที่ 3 ใช้อัลกอริทึม PSO เพื่อหาค่าน้ำหนักที่เหมาะสมที่สุดโดยฟังก์ชันความเหมาะสมของวิธีการ PSO จากสมการที่ 3.16

ขั้นตอนที่ 4 ข้อมูลทดสอบ x จะคำนวณระยะทางระหว่างชุดข้อมูลเรียนรู้ y ด้วยสมการที่ 3.17 และพิจารณาเพื่อนบ้านที่ใกล้ที่สุดจำนวน K ตัวนั้น

$$d(x, y) = \sum_{r=1}^{m} w_i^2\ (a_i(x), a_i(y))^2 \quad (3.17)$$

ขั้นตอนที่ 5 กำหนดคลาสคำตอบของข้อมูลทดสอบด้วยคลาสคำตอบส่วนใหญ่ของเพื่อนบ้านที่ใกล้ที่สุดจำนวน K ตัว

3.5 การปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนักสำหรับการจำแนกประเภทข้อมูลที่ไม่สมดุล (An Improved Weighted KNN Algorithm for Imbalanced Data Classification)

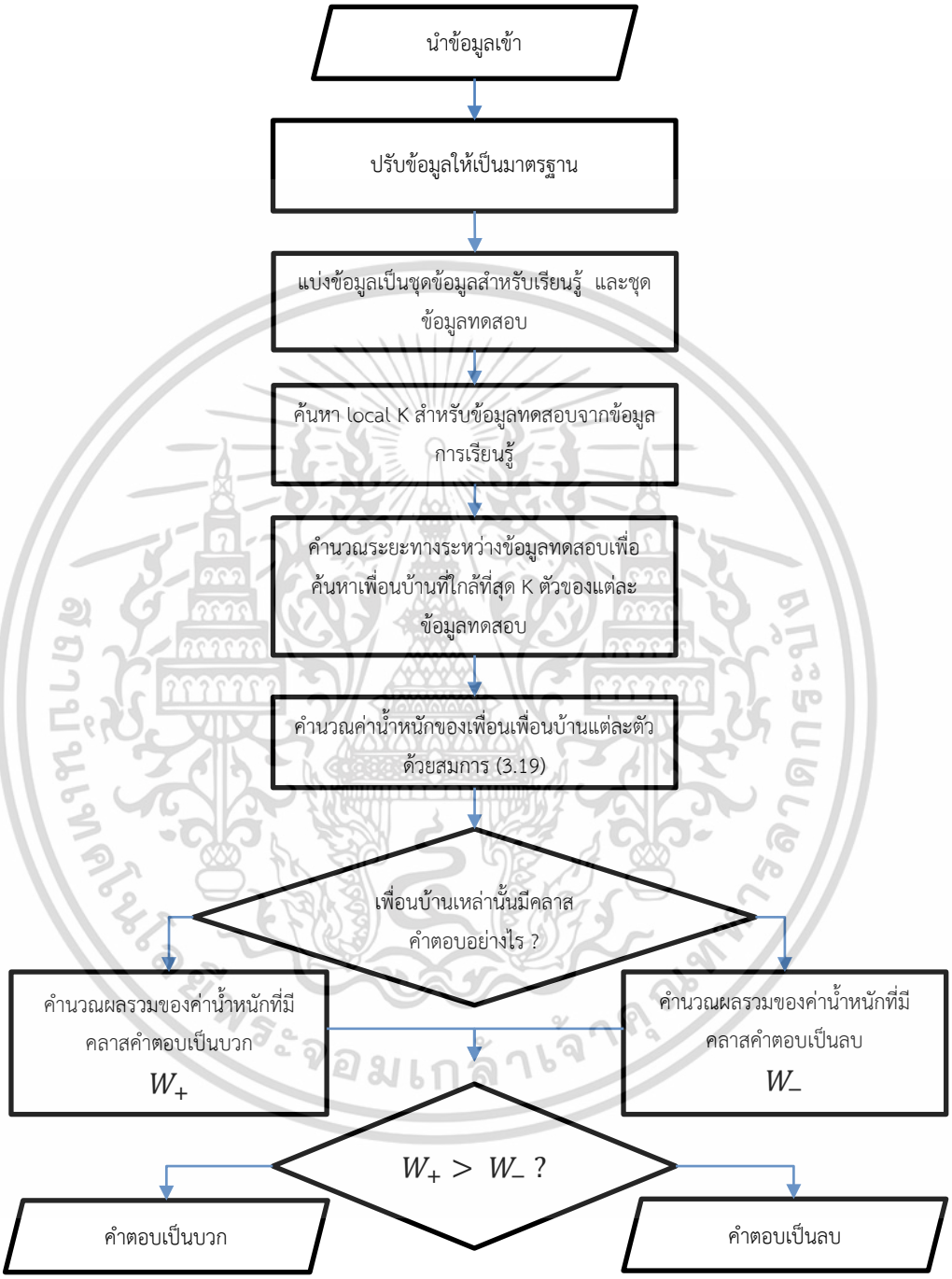
ในงานวิจัยนี้ผู้วิจัยได้นำเสนอวิธีการ PTM-WKNN [22] โดยได้ทำการทดลองนำเอาข้อมูลสำหรับการเรียนรู้ เพื่อที่จะเลือกค่าที่เหมาะสมที่สุดของค่า K สำหรับทุกชุดข้อมูลตามลักษณะเฉพาะของข้อมูลเหล่านั้นเพื่อนำไปใช้กับชุดข้อมูลที่คลาสไม่สมดุล (class-imbalanced data sets) โดยปัญหานี้เป็นปัญหาในการจำแนกประเภทแบบไบนารี ซึ่งส่วนใหญ่เกิดจากการที่ชุดข้อมูลจะแบ่งออกเป็นคลาสส่วนน้อยและคลาสส่วนใหญ่ ซึ่งในงานวิจัยนี้ผู้วิจัยได้กำหนดให้ใช้คลาสส่วนน้อยเป็นคลาสบวกและคลาสส่วนใหญ่เป็นคลาสลบ จากนั้นพิจารณาความแตกต่างของผลกระทบเนื่องจากระยะทางระหว่างข้อมูลที่พิจารณาเพื่อจำแนกประเภทกับข้อมูลเพื่อนบ้าน โดยผู้วิจัยได้กำหนดค่าน้ำหนักที่แตกต่างกันให้กับเพื่อนบ้านเหล่านั้นซึ่งเรียกว่าการให้น้ำหนัก (weighted) ในวิธีการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนัก (weighted K-nearest neighbor หรือ WKNN) ซึ่งวิธีการนี้จะจำแนกข้อมูลทดสอบด้วยการลงคะแนนของค่าน้ำหนัก (weighted voting) ผู้วิจัยได้กล่าวว่าวิธีการ PTM-WKNN ที่นำเสนอใหม่นี้จะรวมข้อดีของวิธีการที่ผ่านมาและมีจุดมุ่งหมายในการปรับปรุงประสิทธิภาพการจำแนกประเภทของข้อมูลที่ไม่สมดุล

แนวทางการทำงานของอัลกอริทึม PTM-WKNN จะยึดตามลักษณะเฉพาะของแต่ละข้อมูลเพื่อที่จะกำหนดค่า K ที่เหมาะสมที่สุดกับข้อมูลเหล่านั้น โดยขั้นแรกจะทำการปรับชุดข้อมูลที่จะทำการพิจารณาให้เป็นมาตรฐานและแบ่งชุดข้อมูลการเรียนรู้และชุดข้อมูลทดสอบตามสัดส่วน การปรับข้อมูลให้เป็นมาตรฐาน (normalization) นั้นสามารถปรับได้ดังสมการที่ 3.18

$$X_{new} = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{3.18}$$

โดยจากชุดข้อมูลการเรียนรู้แต่ละข้อมูลอัลกอริทึมจะทำการค้นหาค่า K ที่เหมาะสมที่สุดค่าและเก็บไว้เป็นค่า K เฉพาะข้อมูล (local K) โดยวิธีการนี้ได้ถูกนำเสนอในงานวิจัย [23] ซึ่งถูกพัฒนาโดย Pedrajas และคณะซึ่งได้นำเสนอวิธีการชุดข้อมูลเรียนรู้ที่นำเสนอ (The proposed training method หรือ PTM) โดยได้คำนึงถึงลักษณะเฉพาะของข้อมูลทดสอบ แต่ละข้อมูลจะได้ถูกกำหนดค่า K ที่เหมาะสมพร้อมประสิทธิภาพที่ดีที่สุด จากนั้นจำแนกข้อมูลการทดสอบโดยใช้ค่า K เฉพาะข้อมูลของข้อมูลนั้น ๆ อัลกอริทึมจะกำหนดคลาสคำตอบด้วยผลรวมค่าน้ำหนักในแต่ละคลาสคำตอบซึ่ง

สอดคล้องตามระยะทางที่แตกต่างของเพื่อนบ้านแต่ละตัวจากข้อมูลทดสอบ โดยกระบวนการของวิธีการ PTM-WKNN สามารถแสดงในรูปที่ 3.2



รูปที่ 3.2 ขั้นตอนการทำงานของวิธีการ PTM-WKNN

จากรูปที่ 3.2 ซึ่งแสดงกระบวนการของวิธีการ PTM-WKNN ผู้วิจัยได้อธิบายวิธีการในแต่ละขั้นตอนออกเป็นทั้งหมด 3 ขั้นตอนดังนี้

ขั้นตอนที่ 1 เพื่อให้ได้ค่า K ที่เหมาะสมที่สุดของข้อมูลการเรียนรู้ วิธีการจะทำการกำหนดช่วงค่า K คือ $[min\ ,\ max]$ เฉพาะจำนวนคี่ จากนั้นนำค่า K เฉพาะข้อมูลแต่ละตัวไปทำการหาค่าความแม่นยำของจำแนกประเภทด้วยข้อมูลการเรียนรู้ตามวิธีการ PTM และเลือกค่า K เฉพาะข้อมูลด้วยประสิทธิภาพที่ดีที่ ซึ่งในงานวิจัยนี้ได้ระบุขอบเขตการค้นหาด้วยค่า [1, 20] สำหรับตัวอย่างการฝึกอบรมค่า K เฉพาะข้อมูล โดยค่าเหมาะสมคือค่า K ขั้นต่ำที่สามารถใช้จำแนกประเภทได้อย่างถูกต้องเป็นครั้งแรก ถ้าค่า K ในช่วงที่พิจารณาไม่สามารถจำแนกประเภทได้อย่างได้อย่างถูกต้องจากช่วงที่กำหนดนั้นค่า K เฉพาะข้อมูลคือ 1

ขั้นตอนที่ 2 ทำการหาค่า K เฉพาะข้อมูลของข้อมูล ในการเรียนรู้ตามค่า K เฉพาะข้อมูลของการข้อมูลการเรียนรู้แต่ละค่าซึ่งต้องมีการเลือกจากเพื่อนบ้านที่ใกล้ที่สุด c ตัวของข้อมูลทดสอบในการเรียนรู้ โดยการค้นหาเพื่อนบ้านที่ใกล้เคียงที่สุดจะสามารถพิจารณาได้ตามระยะทางแบบยุคลิดและกำหนดค่า K เฉพาะข้อมูลทดสอบ ตามค่า K ที่มากที่สุดจากเพื่อนบ้านที่เป็นข้อมูลเรียนรู้ทั้ง c ตัวเพื่อใช้ในชุดข้อมูลทดสอบ ซึ่งในงานวิจัยนี้ผู้วิจัยกำหนดค่าของ c เท่ากับ 3 หรือกล่าวได้ว่าพิจารณาเพื่อนบ้านทั้งหมด 3 ตัว ซึ่งสามารถแสดงกระบวนการในขั้นตอนที่ 2 ดังรูปที่ 3.3

PTM-WKNN : Step2

1. for every $x_i \in R^D$, $i \in$ (m+1, … , n)
2. obtain the c nearest neighbors in training set
3. for i ∈ [1, c], do :
4. add $neighbors_i$'s local k_i to klist
5. end
6. local k_i=max(klist)
7. end
8. return local k

รูปที่ 3.3 กระบวนการของวิธีการ PTM-WKNN เพื่อกำหนดค่า K ที่เหมาะสมจากข้อมูลการเรียนรู้

ขั้นตอนที่ 3 จะเป็นการจำแนกประเภทของชุดข้อมูลทดสอบ ด้วยวิธีการลงคะแนนจากผลรวมค่าน้ำหนักของเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวเฉพาะข้อมูลในแต่ละคลาส โดยอัลกอริทึมเริ่มจากคำนวณระยะทางระหว่างข้อมูลเพื่อนบ้านตามจำนวน K ตัวเฉพาะแต่ละข้อมูลกับข้อมูลที่ใช้ทดสอบด้วยวิธีการแบบยุคลิด จากนั้นอัลกอริทึมจะทำการคำนวณค่าน้ำหนักตามสมการที่ 3.19

$$W_j = \frac{1}{\text{dist}(x_i, x_j)} \qquad (3.19)$$
$$j = 1, 2, \ldots, local\ k_i$$

สมการที่ 3.19 จะสังเกตได้ว่ายิ่งหากเพื่อนบ้านใกล้กับข้อมูลทดสอบมีค่าระยะห่างมีค่าน้อยเท่าใด จะส่งผลให้น้ำหนักมีค่ามาก ในทางกลับกันหากเพื่อนบ้านที่อยู่ห่างออกไปค่าน้ำหนักก็ยิ่งน้อยลงเท่านั้น หลังจากคำนวณน้ำหนักเพื่อใช้ในการจำแนกทั้งหมดของข้อมูลแล้ว อัลกอริทึมจะทำการคำนวณผลรวมของน้ำหนักเพื่อนบ้านที่เป็นบวก (W_+) หรือในงานวิจัยคือคำตอบซึ่งเป็นคลาสส่วนน้อย และผลรวมของน้ำหนักเพื่อนบ้านที่เป็นลบ (W_-) หากเลือกคำตอบเป็นคลาสส่วนมาก หลังจากนั้นคลาสคำตอบที่มีน้ำหนักรวมสูงสุดจะเป็นคำตอบของการจำแนกประเภทของข้อมูล

โดยสรุปแล้ววิธี PTM-WKNN จะรวมข้อดีของทั้งวิธีการ PTM-KNN และวิธีการ WKNN โดยคำนึงถึงลักษณะเฉพาะของแต่ละข้อมูลทดสอบที่แตกต่างกัน ซึ่งจะส่งผลให้มีความแตกต่างกันของค่า K เฉพาะแต่ละข้อมูล ในขณะที่ WKNN คำนึงถึงความแตกต่างในอิทธิพลของเพื่อนบ้านที่แตกต่างกันของเพื่อนบ้านด้วยค่าน้ำหนักซึ่งมีผลตามระยะห่างจากเพื่อนบ้านที่พิจารณา โดยการใช้ค่าน้ำหนักที่นำเสนอมีจุดมุ่งหมายเพื่อปรับปรุงประสิทธิภาพของการจำแนกข้อมูลที่ไม่สมดุลเพื่อให้อัลกอริทึมสามารถแก้ปัญหาความแม่นยำในการจำแนกประเภทสำหรับสถานการณ์เฉพาะที่กล่าวถึงได้ดียิ่งขึ้น

3.6 อัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบพลวัตด้วยระยะทางและการให้น้ำหนักกับคุณลักษณะสำหรับการจำแนกประเภท (Dynamic K-Nearest-Neighbor with Distance and Attribute Weighted for Classification)

ในงานวิจัยชิ้นนี้ได้นำเสนอการปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมด้วยวิธีการหาค่า K แบบพลวัต และอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยระยะทางจากการให้น้ำหนักกับคุณลักษณะสำหรับการจำแนกประเภท (DKNDAW) ซึ่งผู้วิจัยได้นำเอาวิธีการเลือกค่า K แบบพลวัต การให้น้ำหนักกับระยะทางหรือคุณลักษณะและ และการจำแนกประเภทด้วยการให้ค่าน้ำหนัก มาใช้ร่วมกันทั้ง 3 วิธี เพื่อแก้ปัญหาอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม โดยปัญหาหลัก 3 ประการที่เป็นปัญหาซึ่งส่งผลต่อความแม่นยำในการจำแนกประเภทคลาสคำตอบของข้อมูลทดสอบของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว คือ วิธีการคำนวณค่าระยะทางสำหรับวัดความแตกต่างหรือความคล้ายคลึงกันระหว่างข้อมูลซึ่งในวิธีการแบบดั้งเดิมคำนวณด้วยวิธีการแบบยุคลิด การกำหนดค่าพารามิเตอร์ที่ใช้ในการระบุจำนวนของเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว และวิธีการประมาณความน่าจะเป็นของคลาสคำตอบข้อมูลซึ่งในอัลกอริทึมแบบดั้งเดิมจะขึ้นอยู่กับคลาสคำตอบส่วนใหญ่ของเพื่อนบ้านที่ใกล้ที่สุด K ตัว จากการศึกษาผู้วิจัยพบว่าแนวมีทางในการแก้ปัญหาของอัลกอริทึมแบบดั้งเดิมทั้ง 3 ข้อดังนี้ 1. ใช้วิธีการคำนวณระยะทางที่ดีกว่าวิธีการยุคลิดแบบมาตรฐาน 2. ค้นหาค่า K ที่ดีที่สุดในแต่ละการค้นหาแบบพลวัตแทนการกำหนดค่าเองในแต่ละครั้ง 3. นำเอาวิธีการจำแนกประเภทที่มีประสิทธิภาพแม่นยำมากกว่ามาใช้งานแทนวิธีการกำหนดคลาสคำตอบของข้อมูลทดสอบด้วยคำตอบส่วนใหญ่

ด้วยแนวคิดในการแก้ปัญหาทั้ง 3 ข้อนี้ ผู้วิจัยจึงได้นำเอาการแก้ปัญหาดังกล่าวมารวมกันเป็นอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบพลวัตด้วยระยะทางและการให้น้ำหนักกับคุณลักษณะสำหรับการจำแนกประเภท โดยอัลกอริทึมที่นำเสนอได้แก้ปัญหาทั้ง 3 ข้อด้วยวิธีการดังนี้ 1. ในอัลกอริทึมได้ใช้วิธีการให้ค่าน้ำหนักกับระยะทางด้วยวิธีคำนวณค่าความเกี่ยวพันข้อมูล (Mutual Information) แทนการใช้ระยะทางแบบวิธีการยุคลิดแบบมาตรฐานเพื่อวัดความแตกต่างหรือความเหมือนกันระหว่างข้อมูลได้ถูกต้อง 2. เพื่อแก้ปัญหาของการกำหนดค่า K ในอัลกอริทึมแบบดั้งเดิม ในอัลกอริทึมที่นำเสนอผู้วิจัยได้ใช้วิธีในการค้นหาค่า K ที่ดีที่สุดสำหรับการจำแนกประเภท 3. ใช้วิธีคำนวณค่าความเกี่ยวพันข้อมูล (Mutual Information) ร่วมกับการจำแนกประเภทเพื่อแก้ปัญหาของวิธีการแบบดั้งเดิมซึ่งใช้คำตอบส่วนใหญ่ของเพื่อนบ้านที่ใกล้ที่สุด K ตัว ซึ่งการพัฒนาอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบพลวัตด้วยระยะทางและการให้น้ำหนักกับคุณลักษณะสำหรับการจำแนกประเภทที่นำเสนอผู้วิจัยได้นำเอาวิธีการต่าง ๆ มาใช้ร่วมกับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว ซึ่งสามารถอธิบายการทำงานของแต่ละวิธีการได้ดังนี้

1. **การค้นหาค่า K แบบพลวัตเพื่อหาค่าที่ดีที่สุด** เนื่องจากค่า K มีความสำคัญต่อความแม่นยำในการจำแนกประเภท ในอัลกอริทึมที่นำเสนอผู้วิจัยได้ใช้วิธีในการค้นหาค่า K ที่ดีที่สุดสำหรับการจำแนกประเภทโดย วิธีการจะค้นหาค่า K ที่ดีที่สุดในชุดข้อมูลการเรียนรู้คือการลองใช้ค่า K ต่าง ๆ แล้วเลือกค่าที่ดีที่สุด แนวทางหนึ่งที่ได้ผลในการเรียนรู้ค่า K ที่ดีที่สุดคือ DKNN ซึ่งใช้การตรวจสอบความถูกต้องแบบไขว้ (cross-validation) ด้วยวิธี Leave one out เพื่อเลือกค่าที่เหมาะสมที่สุดสำหรับ K ในชุดข้อมูลการเรียนรู้ ส่วนในวิธีการที่นำเสนอทำการเรียนรู้ค่า K ที่ดีที่สุดในจากข้อมูลการเรียนรู้เพื่อจำแนกประเภทของข้อมูลทดสอบโดยการทำงานของวิธีการ Leave one out เป็นตรวจสอบความถูกต้องแบบไขว้ซึ่งเกี่ยวข้องกับการใช้ข้อมูลเพียง 1 ตัวจากข้อมูลการเรียนรู้เป็นข้อมูลการทดสอบสอบความถูกต้องและข้อมูลในการเรียนรู้ที่เหลือเป็นข้อมูลที่ใช้ในการเรียนรู้ (training) โดยจะทำสิ่งนี้จนกระทั่งครบจำนวนของข้อมูลที่ใช้ในการเรียนรู้ ซึ่งในการทดสอบแต่ละครั้งวิธีการจะทำการกำหนดค่าของ K ตั้งแต่ค่าที่มากที่สุดไปจนถึงค่าที่น้อยที่สุดที่สามารถจำแนกได้ถูกต้อง จากนั้นทำการนับจำนวนการจำแนกประเภทที่ถูกต้องและเลือกค่า K ที่ทำให้การจำแนกประเภทถูกต้องมากที่สุดมาใช้กับอัลกอริทึม DKNDAW โดยสามารถแสดงรายละเอียดการค้นหาค่า K ที่ดีที่สุดของอัลกอริทึมได้ดังรูปที่ 3.4

$$Dynamic\ K - Nearest - Neighbor$$

```
1. For each instance t in T
2.      T = T-{t}
3.      Find the maxk nearest neighbors
4.      For k = maxk to mink
5.              clause the performance of each k nearest neighbors
6.      T = T+{t}
7. Select the best k value through the performance stats
```

รูปที่ 3.4 ขั้นตอนการทำงานของการค้นหาค่า K ที่ดีที่สุดในอัลกอริทึม DKNDAW

2. **การให้ค่าน้ำหนักกับคุณลักษณะด้วยค่าความเกี่ยวพันข้อมูล (Mutual Information)** ในอัลกอริทึมแบบดั้งเดิมความแม่นยำของการจำแนกประเภทจะขึ้นอยู่กับคลาสคำตอบส่วนใหญ่ของเพื่อนบ้านที่ใกล้ที่สุด K ตัว ซึ่งเพื่อนบ้านเหล่านั้นจะขึ้นอยู่กับการคำนวณระยะทางของแต่ละคุณลักษณะ โดยในอัลกอริทึมแบบดั้งเดิมใช้ระยะทางแบบยุคลิดมาตรฐานคำนวณจากคุณลักษณะทั้งหมดด้วยความสำคัญเท่ากัน แต่ในความเป็นจริงคุณลักษณะบางตัวอาจจะไม่มีความเกี่ยวข้องกับการจำแนกประเภทส่งผลให้เมื่อจำนวนคุณลักษณะที่ไม่เกี่ยวข้อง

เพิ่มขึ้นความแม่นยำของการจำแนกประเภทก็จะลดลง ในงานวิจัยชิ้นนี้ได้รับแรงบันดาลใจจากวิธีการ WAKNN ซึ่งนำเสนออัลกอริทึมที่ปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยการให้น้ำหนักกับแต่ละคุณลักษณะจากการคำนวณค่าความเกี่ยวพันข้อมูลระหว่างคุณลักษณะและคลาสคำตอบ $Ip(A_r; C)$ วิธีการให้ค่าน้ำหนักกับคุณลักษณะด้วยค่าความเกี่ยวพันข้อมูลสามารถคำนวณดังสมการที่ 3.20

$$d(x\ , y) = \sum_{r=1}^{m} w_r\ \delta(a_r(x)\ , a_r(y)) \tag{3.20}$$

โดยที่ w_r คือน้ำหนักของคุณลักษณะ A_r แต่เมื่อข้อมูลเป็นประเภทนามบัญญัติ (Nominal data) ค่าระยะทางสามารถคำนวณดังสมการที่ 3.21

$$d(x\ , y) = \sum_{r=1}^{m} Ip(A_r; C)\ \delta(a_r(x)\ , a_r(y)) \tag{3.21}$$

3. **การให้ค่าน้ำหนักกับระยะทาง** วิธีการนี้จะทำการกำหนดค่าน้ำหนักให้กับระยะทางของเพื่อนบ้าน K ตัวที่ใกล้ที่สุดเหล่านั้นด้วยค่าที่แตกต่างกันตามระยะทางระหว่างข้อมูลนั้นและข้อมูลทดสอบ เพื่อใช้ค่าน้ำหนักในการจำแนกประเภทข้อมูลทดสอบด้วยผลรวมของค่าน้ำหนักในแต่ละคลาสคำตอบที่มีค่ามากที่สุดดังในสมการที่ 3.22 แทนวิธีการจำแนกประเภทด้วยคลาสคำตอบส่วนใหญ่

$$c(x) = \ arg\ max \sum_{i=1}^{k} w\ \delta(c, c(y))) \tag{3.22}$$

การคำนวณค่าน้ำหนักของระยะทางในอัลกอริทึมที่นำเสนอได้นำเอาวิธีการ KNNDW มาคำนวณค่าน้ำหนักของเพื่อนบ้านแต่ละตัวตามค่าผกผันของระยะทางกำลังสองจากข้อมูลเรียนรู้ นอกจากนี้ในอัลกอริทึมที่นำเสนอผู้วิจัยได้เพิ่มตัวเลข 0.001 ไปยังตัวหารเพื่อหลีกเลี่ยงสถานการณ์ที่ระยะทางเป็น 0 ดังสมการที่ 3.23

$$w_r = \frac{1.0}{d(\ x_r\ , x_q\ )^2 + 0.001} \tag{3.23}$$

อัลกอริทึม DKNDAW ในงานวิจัยชิ้นนี้สามารถแบ่งการทำงานของอัลกอริทึมออกเป็น 2 ส่วนหลักซึ่งประกอบด้วย การคำนวณเพื่อค้นหาค่า K ที่ดีที่สุดเพื่อใช้ในการค้นหากลุ่มเพื่อนบ้านที่ใกล้ที่สุด และในการจำแนกประเภทของข้อมูลทดสอบจะกำหนดคลาสคำตอบจากเพื่อนบ้านที่ใกล้ที่สุด K ตัวด้วยการคำนวณร่วมกับการให้น้ำหนัก ซึ่งในงานวิจัยชิ้นนี้ผู้วิจัยได้ใช้ข้อมูลสาธารณะ UCI 36 ชุด ในการทดลอง โดยทำการเปรียบเทียบผลการจำแนกประเภทของอัลกอริทึมที่ได้นำเสนอกับอัลกอริทึม KNN แบบดั้งเดิม อัลกอริทึม WAKNN อัลกอริทึม KNNDW อัลกอริทึม KNNDAW และอัลกอริทึม DKNN ผลการทดลองแสดงให้เห็นว่าอัลกอริทึมที่นำเสนอเมื่อนำวิธีการให้ค่าน้ำหนักกับคุณลักษณะและการให้ค่าน้ำหนักกับระยะทาง รวมทั้งใช้แนวคิดแบบพลวัตเพื่อแก้ปัญหา สามารถเพิ่มประสิทธิภาพความแม่นยำในการจำแนกประเภทได้ดีกว่าวิธีการอื่น ๆ อย่างมีนัยสำคัญ โดยสามารถแสดงรายละเอียดการทำงานของอัลกอริทึมที่นำเสนอได้ดังรูปที่ 3.5

DKNDAW algorithm

Input: training instances T, a test instance y,
the maximum K value, the minimum K value
Output: the target class label
1. For each instance t in T
2. T = T-{t}
3. Find the maxK nearest neighbors
4. For k = maxK to minK
5. clause the performance of each K nearest neighbors
6. T = T+{t}
7. Select the best K value through the performance stats
8. Find the best K neatest neighbors to meet the training data sets T
9. Use Equation 3.20 and 3.21 to calculate the distance between y and each training instance x
10. Use Equation 3.23 to calculate distance weight w_r and 3.22 to calculate the class label of the test instance y on T
11. Return the class label $c(y)$

รูปที่ 3.5 ขั้นตอนการทำงานของอัลกอริทึม DKNDAW

3.7 การปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยวิธีค่าเกนความรู้และการแบ่งกลุ่มข้อมูล (An Enhanced K-Nearest Neighbor Algorithm Using Information Gain and Clustering)

ในงานวิจัยนี้ ผู้วิจัยได้ศึกษาวิธีการในการปรับปรุงประสิทธิภาพของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว ซึ่งจากการศึกษาปัญหาของอัลกอริทึมแบบดั้งเดิม ผู้วิจัยได้กล่าวว่าอัลกอริทึมแบบดั้งเดิมมีข้อบกพร่อง 3 ข้อหลักดังนี้ 1. วิธีการที่ใช้ในการคำนวณระยะทางแบบดั้งเดิมคือระยะทางแบบยุคลิดซึ่งจะทำการพิจารณาระยะทางของแต่ละคุณลักษณะอย่างเท่าเทียมกันทั้งหมด 2. การกำหนดจำนวนของเพื่อนบ้านที่ใกล้ที่สุดเพื่อพิจารณาคลาสคำตอบ ซึ่งจำนวนของเพื่อนบ้านที่ใกล้ที่สุด (ค่า K) เป็นพารามิเตอร์ซึ่งอาจส่งผลให้การจำแนกประเภทได้ผลลัพธ์ที่ไม่ถูกต้องเมื่อชุดข้อมูลไม่สมดุล (unbalanced data) 3. การจำแนกประเภทคลาสคำตอบของข้อมูลทดสอบในอัลกอริทึมแบบดั้งเดิมนั้นจะขึ้นอยู่กับจำนวนคลาสคำตอบที่มากที่สุด และด้วยวิธีการที่เรียบง่ายนี้อาจจะทำให้การจำแนกประเภทเกิดความผิดพลาด จากการศึกษาได้มีงานวิจัยที่นำเอาวิธีการต่าง ๆ ในการปรับปรุงความแม่นยำของการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว ตามปัญหาที่กล่าวมาข้างต้น เช่น ในงานวิจัยของ Wu และคณะ [12] ได้มีการปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมใกล้เคียงตามข้อสังเกตของผู้วิจัย โดยใช้ฟังก์ชันระยะทางที่แม่นยำยิ่งขึ้นด้วยการคำนึงถึงลำดับความสำคัญของคุณลักษณะ ค้นหาขนาดของกลุ่มเพื่อนบ้านที่ใกล้ที่สุดให้เหมาะสมกับข้อมูลทดสอบเพื่อผลลัพธ์ที่แม่นยำยิ่งขึ้น และวิธีการประมาณความน่าจะเป็นของคำตอบที่แม่นยำมากยิ่งขึ้นเพื่อใช้แทนการกำหนดคำตอบจากคลาสคำตอบส่วนใหญ่ในอัลกอริทึมแบบดั้งเดิม เป็นต้น

ผู้วิจัยจึงได้พัฒนางานวิจัยการปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยวิธีค่าเกนความรู้และการแบ่งกลุ่มข้อมูลขึ้นนี้ขึ้น ซึ่งเป็นการผสมผสานระหว่างเทคนิคของ 3 วิธีการ คือการเลือกค่า K แบบพลวัต วิธีการให้น้ำหนักกับคุณลักษณะโดยใช้ค่าเกนความรู้ และวิธีการให้น้ำหนักกับระยะทางในการกำหนดคำตอบให้กับข้อมูลที่จะทำการจำแนก โดยสามารถอธิบายลักษณะของแต่ละวิธีการที่นำมาใช้พอสังเขปได้ดังนี้

1. **การเลือกค่า K แบบพลวัต** การเลือกค่า K เป็นส่วนสำคัญของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว ในชุดข้อมูลซึ่งเรานำมาเรียนรู้ในการทดสอบอาจมีจำนวนของคลาสคำตอบไม่เท่ากันซึ่งคลาสคำตอบส่วนใหญ่จะมีจำนวนมากกว่าคลาสอื่น หากในการจำแนกประเภทกำหนดค่า K เป็นค่าคงที่ค่าหนึ่งแล้วผลลัพธ์ในการจำแนกจะมีความน่าจะเป็นที่จะเอนเอียงไปทางคลาสคำตอบส่วนใหญ่ เพื่อหลีกเลี่ยงปัญหานี้งานวิจัยหลายชิ้นได้เสนออัลกอริทึมที่แตกต่างกันเพื่อเพิ่มประสิทธิภาพในการเลือกค่า K วิธีการหนึ่งที่ได้ถูกกล่าวถึงคือแนวคิดในการทดลองใช้ค่า K ที่แตกต่างกันและเลือกค่า K ที่ให้ผลลัพธ์ดีที่สุด จากแนวคิดนี้ได้มีการเสนอวิธีการเลือกเพื่อนบ้านด้วยการเรียนรู้แบบเบย์อย่างง่าย (Selective Neighborhood

Naive Bayes) วิธีการนี้จะมีการทดสอบค่า K ต่าง ๆ จากการจำแนกข้อมูลทดสอบโดยใช้อัลกอริทึมการเรียนรู้แบบเบย์อย่างง่าย (Naïve Bayesian) สำหรับค่า K แต่ละค่า โดยจะทำการกำหนดค่า K เริ่มต้นที่น้อยที่สุด และค่า K ที่มากที่สุดเป็นช่วงในการค้นหา โดยในการทดสอบค้นหาค่า K ที่ดีที่สุดของอัลกอริทึมที่นำเสนอสามารถอธิบายได้ดังรูปที่ 3.6

Selective Neighborhood Naive Bayes algorithm

```
1. For K = minK to maxK ; coun[K]=0
2. For each intance e in T
3. T = T− {e}
4. Find the maxK nearest neighbors of e from T
5.        For K = maxK to mink
 6.                Train a local naive Bayes using the K nearest neighbors
7.                 Classify e using the local naive Bayes
 8.                If the classification of e is correct, count[K]++
9.                 Remove the instance with the greatest distance
10.      T = T+ {e}
11. bestK = minK
12. maxCount = count[minK]
13. For K = minK + 1 to maxK
14.       If count[K] > maxCount
15.                bestK = K
16.                maxCount = count[K]
17. Return the value of bestK
```

รูปที่ 3.6 ขั้นตอนการค้นหาค่า K แบบพลวัตด้วยวิธีการเรียนรู้แบบเบย์อย่างง่าย

จากนั้นนำค่า K ซึ่งสามารถจำแนกประเภทได้ถูกต้องและมีค่ามากที่สุดไปใช้สำหรับการจำแนกประเภทของข้อมูลทดสอบ วิธีนี้ใช้เวลานานมากเมื่อนำมาใช้งานจริง เนื่องจากจำเป็นต้องสร้างรูปแบบการคำนวณแบบเบย์อย่างง่ายเพื่อหาค่า K ที่เหมาะสม ผู้วิจัยยังได้กล่าวแนวทางอื่น ๆ ที่มีประสิทธิภาพการใช้เวลาในการค้นหาค่า K ที่ดีที่สุดสำหรับขั้นตอนการเรียนรู้คืออัลกอริทึม DKNN ของ Wu และคณะ [12] ซึ่งใช้การตรวจสอบความถูกต้องแบบไขว้ด้วยวิธี Leave one out เพื่อเลือกค่า K ที่เหมาะสมที่สุดสำหรับชุดข้อมูลการเรียนรู้

2. **วิธีการให้น้ำหนักกับคุณลักษณะ** อัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมใช้ระยะทางแบบยุคลิดเพื่อวัดความแตกต่างหรือความคล้ายคลึงกันระหว่างข้อมูลการเรียนรู้

และข้อมูลทดสอบ โดยจะพิจารณาความสำคัญของแต่ละคุณลักษณะในการจำแนกประเภทอย่างเท่าเทียมกันทั้งหมดไม่ว่าคุณลักษณะนั้นจะเกี่ยวข้องหรือไม่ก็ตาม ดังนั้นเมื่อมีคุณลักษณะที่ไม่เกี่ยวข้องจำนวนมากการจำแนกประเภทโดยใช้ค่าระยะทางแบบยุคลิดแบบมาตรฐานจึงไม่แม่นยำ ผู้วิจัยได้นำเอาวิธีการซึ่งมีแนวทางในการเอาชนะปัญหานี้อย่างมีประสิทธิภาพคือการกำหนดระดับความสำคัญให้กับคุณลักษณะ โดยกำหนดค่าน้ำหนักให้กับแต่ละคุณลักษณะตามความสำคัญเพื่อใช้ในการคำนวณระยะห่างระหว่างข้อมูลการเรียนรู้และข้อมูลทดสอบ ค่าน้ำหนักนั้นจะถูกกำหนดด้วยค่า w_i ซึ่งค่า i ระบุถึงคุณลักษณะลำดับที่ i ค่าน้ำหนักของคุณลักษณะแต่ละค่าจะคำนวณจากค่าเกนความรู้ (Information gain) ระหว่างคุณลักษณะแต่ละตัวกับคลาสคำตอบของชุดข้อมูลเรียนรู้ การคำนวณด้วยระยะทางแบบยุคลิดแบบให้ค่าน้ำหนักกับคุณลักษณะ สามารถคำนวณได้ดังสมการที่ 3.24

$$d(x_i, x_j) = \sum_{r=1}^{m} w_i \left(a_r(x_i) - a_r(x_j)\right) \qquad (3.24)$$

3. **วิธีการให้น้ำหนักกับระยะทาง** อัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมใช้การกำหนดคำตอบให้ข้อมูลทดสอบด้วยการกำหนดคลาสคำตอบจากคำตอบส่วนใหญ่ของเพื่อนบ้านใกล้ที่สุด K ตัว การกำหนดคลาสคำตอบด้วยวิธีดังกล่าวมีผลเสียอย่างมากหากข้อมูลไม่สมดุล ในงานวิจัยชิ้นนี้ได้นำวิธีการปรับปรุงข้อบกพร่องนี้มาใช้ทำการกำหนดคลาสคำตอบจากผลรวมค่าน้ำหนักของเพื่อนบ้านที่ใกล้ที่สุด K ตัวเหล่านั้น โดยค่าน้ำหนักจะแตกต่างกันไปตามระยะทางจากข้อมูลทดสอบ วิธีการนี้เรียกว่าอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้ค่าน้ำหนักกับระยะทางซึ่งได้รับอิทธิพลมาจากงานวิจัยของ Dudani [13] การกำหนดคลาสคำตอบด้วยการให้ค่าน้ำหนักกับระยะทางสามารถคำนวณได้ดังสมการที่ 3.25

$$c(x) = arg\,max \sum_{i=1}^{k} w\,\delta(c, c(y_i)) \qquad (3.25)$$

โดยการคำนวณค่าน้ำหนักของระยะทางของกลุ่มเพื่อนบ้านที่ใกล้ที่สุด K ตัวซึ่งถูกพิจารณาสามารถคำนวณได้ดังสมการที่ 3.26

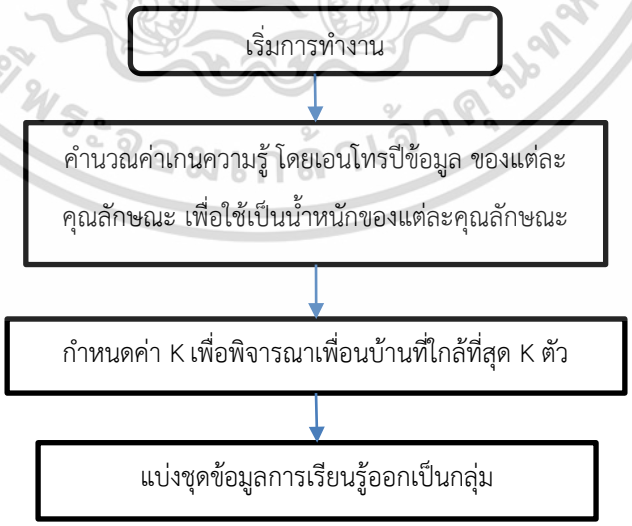
$$w = \frac{1}{(d)^2} \qquad (3.26)$$

นอกจากกระบวนในการแก้ปัญหาทั้ง 3 วิธีที่ได้กล่าวมาข้างต้น วิธีการที่นำเสนอยังได้แก้ปัญหาในด้านความซับซ้อนของเวลาในการทำงานสำหรับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม เนื่องจากการคำนวณระยะห่างระหว่างข้อมูลทดสอบและข้อมูลการเรียนรู้ซึ่งในกระบวนการนี้ใช้เวลามากเพราะต้องคำนวณกับทุกข้อมูลการเรียนรู้ ดังนั้นในวิธีการที่นำเสนอจึงแบ่งข้อมูลการเรียนรู้ออกเป็นกลุ่ม (Cluster) ตามลักษณะของข้อมูลที่คล้ายกัน การแบ่งข้อมูลการเรียนรู้ออกเป็นกลุ่มย่อยช่วยลดเวลาในการคำนวณได้อย่างมากเนื่องจากข้อมูลทดสอบจะเลือกคำนวณระยะห่างระหว่างข้อมูลทดสอบและข้อมูลการเรียนรู้ในกลุ่มนั้น การพิจารณาว่ากลุ่มใดจะเป็นกลุ่มที่ถูกข้อมูลทดสอบพิจารณาสามารถเลือกได้จากศูนย์กลางของกลุ่ม (centroid) ซึ่งคำนวณจากค่าเฉลี่ยของตำแหน่งข้อมูลการเรียนรู้ในแต่ละกลุ่ม จากนั้นศูนย์กลางของกลุ่มที่มีระยะทางใกล้กับข้อมูลทดสอบมากที่สุดจะถูกเลือกเพื่อพิจารณาหาเพื่อนบ้านที่ใกล้ที่สุดจำนวน K ตัวต่อไป

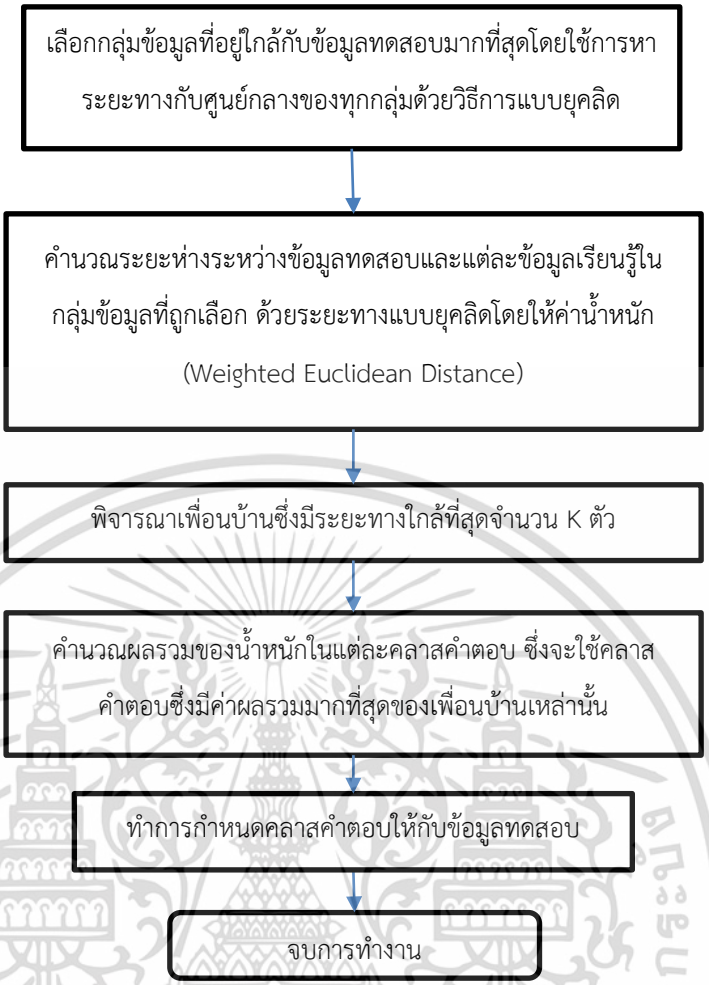
ขั้นตอนกระบวนการของการปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว ที่นำเสนอสามารถอธิบายการทำงานโดยแบ่งออกเป็น 2 ส่วนดังนี้

ส่วนที่ 1 การเตรียมข้อมูลก่อนการประมวลผล: ก่อนการประมวลผล อัลกอริทึมจะให้น้ำหนักกับคุณลักษณะของข้อมูล กำหนดค่า K สำหรับตัวอย่างทดสอบ และแบ่งชุดข้อมูลการเรียนรู้ออกเป็นกลุ่ม (Cluster)

ส่วนที่ 2 การจำแนกประเภท: โดยการจำแนกประเภทข้อมูลการทดสอบจะได้ผลลัพธ์ในส่วนนี้ ซึ่งการทำงานส่วนนี้ทำงานทุกครั้งเมื่อมีการทำการจำแนก โดยขั้นตอนกระบวนการทำงานของวิธีการที่นำเสนอสามารถอธิบายได้ดังรูปที่ 3.7 และรูปที่ 3.8



รูปที่ 3.7 ขั้นตอนกระบวนการของการเตรียมข้อมูลก่อนการประมวลผลในวิธีการที่นำเสนอ



รูปที่ 3.8 ขั้นตอนการทำงานของการจำแนกประเภทในวิธีการที่นำเสนอ

โดยจากรูปที่ 3.7 และรูปที่ 3.8 สามารถอธิบายขั้นตอนกระบวนการของการปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยวิธีค่าเกนความรู้และการแบ่งกลุ่มข้อมูลได้ดังต่อไปนี้

ขั้นตอนที่ 1: คำนวณค่าเอนโทรปีข้อมูล (Information entropy) ของแต่ละคุณลักษณะซึ่งใช้ในการคำนวณค่าเกนความรู้ ของข้อมูลในทุกคุณลักษณะซึ่งจะใช้ค่าเกนความรู้ทำหน้าที่เป็นน้ำหนักของคุณลักษณะที่จะเป็นค่าที่จะจัดลำดับความสำคัญให้กับคุณลักษณะ

ขั้นตอนที่ 2: กำหนดค่า K สำหรับชุดข้อมูลการเรียนรู้

ขั้นตอนที่ 3: แบ่งชุดข้อมูลการเรียนรู้ออกเป็นกลุ่ม

ขั้นตอนที่ 4: ค้นหาค่าเฉลี่ยของกลุ่มทั้งหมดเพื่อให้ได้ศูนย์กลางของทุกกลุ่ม

ขั้นตอนที่ 5: เลือกกลุ่มข้อมูลที่อยู่ใกล้กับข้อมูลทดสอบมากที่สุดโดยใช้การหาระยะทางกับศูนย์กลางของทุกกลุ่มด้วยวิธีการแบบยุคลิด

ขั้นตอนที่ 6: ใช้วิธีคำนวณระยะทางแบบยุคลิดโดยให้ค่าน้ำหนัก (Weighted Euclidean Distance) ตามสมการที่ 3.24 ซึ่งคำนวณระยะห่างระหว่างข้อมูลที่จะทำการจำแนกประเภทและแต่ละข้อมูลเรียนรู้ในกลุ่มข้อมูลที่ถูกเลือก จากนั้นทำการกำหนดเพื่อนบ้านที่ใกล้ที่สุด K ตัวสำหรับข้อมูลที่จะทำการจำแนก

ขั้นตอนที่ 7: ทำการกำหนดคลาสคำตอบให้กับข้อมูลทดสอบด้วยคำตอบส่วนใหญ่ของเพื่อนบ้านที่ใกล้ที่สุด K ตัวในกลุ่มข้อมูล (Cluster) ที่ใกล้ที่สุดตามสมการที่ 3.25 ซึ่งกำหนดจากผลรวมของน้ำหนักในแต่ละคลาสคำตอบซึ่งมีค่ามากที่สุดของเพื่อนบ้านที่ใกล้ที่สุด K ตัว

จากผลการทดลองอัลกอริทึมที่นำเสนอในงานวิจัยชิ้นนี้ จะเห็นได้ว่าสามารถแก้ไขปัญหาของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมได้ด้วยการปรับปรุงจากปัญหาที่สำคัญของอัลกอริทึมแบบดั้งเดิม การนำเอาวิธีคำนวณค่าเกนความรู้ วิธีการให้น้ำหนักกับระยะทางเพื่อใช้ในการจำแนกประเภทคลาสคำตอบของข้อมูลทดสอบ และการแบ่งกลุ่มข้อมูลมาใช้งาน สามารถเพิ่มความถูกต้องของการจำแนกด้วยการให้ค่าน้ำหนักกับคุณลักษณะและลดอิทธิพลของเพื่อนบ้านที่ไม่เกี่ยวข้องกับข้อมูลทดสอบ อีกทั้งยังสามารถลดเวลาในการจำแนกประเภทลงมากจากการนำเอาเทคนิคการจัดกลุ่มเข้ามาผสมผสาน

3.8 การปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยการให้น้ำหนักจากค่าเอนโทรปีของคุณลักษณะ (Enhancement of K-nearest Neighbor Algorithm Based on Weighted Entropy of Attribute Value)

ในงานวิจัยชิ้นนี้ได้นำเสนออัลกอริทึมเพื่อปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม โดยใช้อัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนักจากการวัดค่าความสำคัญของคุณลักษณะ (ค่าน้ำหนัก) ด้วยการคำนวณค่าเอนโทรปี ในส่วนแรกของงานวิจัยผู้วิได้กล่าวถึงข้อบกพร่องของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม โดยจากการศึกษาสามารถสรุปได้ 3 ข้อคือ 1. การระบุค่า K ไม่ได้มีหลักเกณฑ์ที่กำหนดอย่างชัดเจน ซึ่งค่า K มีผลต่อความแม่นยำในการจำแนกประเภทของอัลกอริทึมและยังส่งผลถึงเวลาในการประมวลผลจำแนกอีกด้วย 2. การคำนวณระยะทางจากข้อมูลทดสอบไปยังชุดข้อมูลการเรียนรู้นั้นคุณลักษณะทั้งหมดจะได้รับค่าน้ำหนักเท่ากัน เมื่ออัลกอริทึมประมวลผลข้อมูลที่ไม่เกี่ยวข้องหรือมีความสำคัญน้อยจะส่งผลให้อัตราความแม่นยำในการจำแนกประเภทของอัลกอริทึมได้รับผลกระทบ 3. เวลาที่ใช้ในการประมวลผลค่อนข้างมาก เนื่องจากอัลกอริทึมต้องคำนวณใหม่ทุกครั้งที่มีการค้นหาเพื่อนบ้านของข้อมูลทดสอบนั่นเอง ซึ่งในงานวิจัยนี้ผู้วิจัยมุ่งเน้นไปที่การปรับปรุงข้อบกพร่องในข้อที่ 2 โดยได้นำเสนอการปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุด K ตัวด้วยการให้น้ำหนักกับคุณลักษณะของข้อมูลจากค่าเอนโทรปีของคุณลักษณะเพื่อเพิ่มประสิทธิภาพของอัลกอริทึมแบบดั้งเดิม โดยค่าระยะทางจะมีการเปลี่ยนแปลงไปหลังจากทำการคูณระยะทางแบบยูคลิดด้วยน้ำหนักจากค่าเกนความรู้ (Information gain) ซึ่งได้จากการคำนวณค่าข้อมูลเอนโทรปี (information entropy) จากผลการทดลองนั้นพบว่าผลการจำแนกประเภทของอัลกอริทึมที่นำเสนอมีความแม่นยำเพิ่มมากขึ้นเมื่อเทียบกับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม ในการพัฒนาอัลกอริทึมที่นำเสนอผู้วิจัยได้กล่าวถึงแนวคิดและความรู้ที่เกี่ยวข้องในการใช้พัฒนาอัลกอริทึมที่นำเสนอโดยแบ่งเป็น 3 หัวข้อดังนี้

1. **ค่าข้อมูลเอนโทรปีของคุณลักษณะแต่ละค่า (information entropy of attribute value)** การหาค่าข้อมูลเอนโทรปีในอัลกอริทึมจะทำการพิจารณาชุดข้อมูลการเรียนรู้ด้วยค่าเอนโทรปีของข้อมูลนั้นซึ่งเป็นค่าที่วัดความไม่แน่นอนในการพิจารณาความน่าจะเป็นของการเกิดเหตุการณ์แต่ละเหตุการณ์ในการทดลอง โดยในอัลกอริทึมที่นำเสนอจะพิจารณาค่าโอกาสของการเกิดเหตุการณ์แต่ละค่าคุณลักษณะนั่นเอง การคำนวณจะพิจารณาชุดข้อมูลการเรียนรู้ซึ่งกำหนดให้เป็นชุดข้อมูล D ในชุดข้อมูล D จะประกอบด้วยคลาสคำตอบข้อมูล c จำนวน n คำคอบซึ่ง $D = \{ c_1, c_2, \dots, c_n \}$ ในแต่ละคุณลักษณะจะมีค่า v ซึ่งมีค่าที่แตกต่างกันจำนวน m ตัวซึ่ง $V = \{ v_1, v_2, \dots, v_m \}$ โดยจำนวนของแต่ละค่า

v แทนด้วย $|v_i|$ และจำนวนของแต่ละค่า v ที่มีคำตอบเป็นคลาสคำตอบ j จะแทนด้วย $|v_{ij}|$ ซึ่งการหาค่าข้อมูลเอนโทรปีของค่าในคุณลักษณะแต่ละค่าสามารถคำนวณได้จากสมการที่ 3.27 ดังนี้

$$D(v_i) = -\sum_{j=1}^{m} p_{ij} \ln(p_{ij}) \tag{3.27}$$

โดย $p_{ij} = \frac{|v_{ij}|}{|v_i|}$ คือความน่าจะเป็นของค่าการเกิดค่า v_i เมื่อมีคลาสคำตอบ c_j จากการคำนวณสามารถสรุปได้ว่าหากค่าของคุณลักษณะมีค่าที่สื่อถึงแต่ละคลาสคำตอบตามแต่ละค่าในการจำแนกประเภท ค่าข้อมูลเอนโทรปี $D(v_i)$ จะมีค่าเท่ากับ 0 หรือค่า $|v_i| = |v_{ij}|$ ซึ่งมีความหมายสื่อถึงว่าคุณลักษณะแต่ละค่านั้นสามารถนำไปจำแนกประเภทแต่ละคลาสคำตอบได้ ทั้งนี้ในงานวิจัยยังได้กล่าวว่าหากค่าข้อมูลเอนโทรปีของคุณลักษณะมีค่าน้อยเท่าใดจะสื่อถึงถึงความสำคัญของคุณลักษณะนั้นในการจำแนกประเภทมากขึ้นเท่านั้น

2. **ค่าเกนความรู้ของแต่ละคุณลักษณะ (Information gain)** ค่าเกนความรู้คือการวัดค่าข้อมูลเอนโทรปีของคุณลักษณะว่ามีประสิทธิภาพอย่างไร โดยค่าเกนความรู้ของแต่ละคุณลักษณะจะประกอบด้วยการพิจารณาชุดข้อมูล K ตัวในชุดข้อมูลเรียนรู้ D ซึ่งมีค่าคุณลักษณะเป็น v_i และจำนวนของคำตอบที่อยู่ในคลาสคำตอบข้อมูล c ซึ่งแทนด้วยค่า T_i ค่าเกนความรู้ของแต่ละคุณลักษณะสามารถคำนวณได้ดังสมการที่ 3.28

$$Info(v_i) = D(v_i) - \sum_{i=1}^{m} \frac{T_i}{K} \ln\left(\frac{T_i}{K}\right) \tag{3.28}$$

ซึ่งจากการศึกษาแนวคิดที่ใช้พัฒนาในข้อที่ 1 ผู้วิจัยได้สรุปความสำคัญของค่าเกนความรู้ของแต่ละคุณลักษณะซึ่งจะส่งผลต่อความแม่นยำที่สูงขึ้นเมื่อมีค่าเกนความรู้ที่สูงขึ้น และในทางตรงกันข้ามเมื่อมีค่าเกนความรู้ที่ลดลงค่าความแม่นยำก็จะลดลงตามไปด้วย

3. **การคำนวณระยะทางเมื่อการให้น้ำหนักกับคุณลักษณะด้วยค่าเอนโทรปี (distance based on weighted entropy of attribute value)** การหาค่าน้ำจะเป็นการคำนวณระยะห่างซึ่งมีที่มาจากระยะทางแบบยุคลิดแล้วนำค่าเกนความรู้ของแต่ละคุณลักษณะคูณกับระยะทางในแต่ละคุณลักษณะตามสมการที่ 3.29

$$d(X, Y) = \sqrt{\sum_{i=1}^{m} Info\,(v_i)\,(x_i - y_i)} \qquad (3.29)$$

จากแนวคิดและความรู้ที่เกี่ยวข้องกับการใช้พัฒนาอัลกอริทึมที่นำเสนอทั้ง 3 หัวข้อผู้วิจัยได้สรุปการพัฒนาอัลกอริทึมที่นำเสนอ โดยในการคำนวณระยะทางระหว่างข้อมูลทดสอบและชุดข้อมูลเรียนรู้ในอัลกอริทึมที่นำเสนอจะไม่ได้คำนวณเฉพาะวิธีการแบบยุคลิดตามวิธีการแบบดั้งเดิม แต่จะพิจารณาความสำคัญจากค่าเกนความรู้ของแต่ละคุณลักษณะ โดยมีที่มาจากการวัดค่าข้อมูลเอนโทรปีของแต่ละคุณลักษณะกับคำตอบของข้อมูล เมื่อได้ค่าที่ใช้ระบุถึงความสำคัญของแต่ละคุณลักษณะมาแล้วอัลกอริทึมจะนำค่าดังกล่าวคูณกับระยะทางในแต่ละคุณลักษณะตามสมการที่ 3.28 ซึ่งส่งผลให้การจำแนกประเภทมีความแม่นยำมากยิ่งขึ้นนั่นเอง

อัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยการให้น้ำหนักค่าเอนโทรปีของคุณลักษณะที่นำเสนอนี้ มีการทำงานหลักอยู่ 3 ส่วนโดยในส่วนที่ 1 ทำการคำนวณเอนโทรปีข้อมูลของแต่ละคุณลักษณะตามสมการที่ 3.27 จากนั้นระยะห่างจากตัวอย่างทดสอบไปยังตัวอย่างการฝึกอบรมแต่ละครั้งตามสูตรตามสมการที่ 3.28 ส่วนที่ 2 ค้นหาเพื่อนบ้านที่ใกล้ที่สุดจำนวน K ตัวจากชุดข้อมูลเรียนรู้ด้วยวิธีการคำนวณระยะทางเมื่อมีการให้น้ำหนักกับคุณลักษณะด้วยค่าเอนโทรปีตามสมการที่ 3.29 และส่วนที่ 3 จะเป็นส่วนที่ทำการจำแนกประเภทของข้อมูลทดสอบ ซึ่งจากการทำงานของอัลกอริทึมทั้ง 3 ส่วนนี้สามารถแบ่งขั้นตอนการทำงานของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยการให้น้ำหนักค่าเอนโทรปีของคุณลักษณะออกเป็น 7 ขั้นตอนย่อยได้ดังนี้

ขั้นตอนที่ 1: ปรับค่าในชุดข้อมูลให้อยู่ในค่ามาตรฐานเพื่อป้องกันการคำนวณค่าน้ำหนักในตอนเริ่มต้นที่มากเกินไปหรือน้อยเกินไป โดยใช้วิธีการปรับมาตรฐานสูงที่สุด – น้อยที่สุด (minimum - maximum standardized method)

ขั้นตอนที่ 2: คำนวณเอนโทรปีข้อมูลของแต่ละค่าในแต่ละคุณลักษณะ $D(v_i)$ ตามสมการที่ 3.27 และเก็บค่าเพื่อใช้ในสมการต่อไป

ขั้นตอนที่ 3: คำนวณค่าเกนความรู้ของแต่ละค่าในแต่ละคุณลักษณะ $Info(v_i)$ ตามสมการที่ 3.28 และเก็บค่าเพื่อใช้ในการหาระยะทาง

ขั้นตอนที่ 4: คำนวณระยะทางระหว่างข้อมูลทดสอบกับชุดข้อมูลเรียนรู้ด้วยวิธีการแบบยุคลิดตาม โดยนำค่าเกนความรู้ของแต่ละค่าในแต่ละคุณลักษณะมาคูณกับระยะทางในแต่ละคุณลักษณะตามสมการที่ 3.29

ขั้นตอนที่ 5: ค้นหาเพื่อนบ้านที่ใกล้ที่สุดจำนวน K ตัวของข้อมูลทดสอบ

ขั้นตอนที่ 6: นับจำนวนคำตอบที่เหมือนกันของเพื่อนบ้านที่ใกล้ที่สุด K ตัว

ขั้นตอนที่ 7: ระบุคลาสคำตอบให้กับข้อมูลทดสอบด้วยคำตอบส่วนใหญ่

ในงานวิจัยชิ้นนี้ ผู้วิจัยได้ทำการทดลองเปรียบเทียบระหว่างอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวที่นำเสนอกับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมด้วยชุดข้อมูล 3 ชุด ซึ่งจากการทดลองได้ให้ผลสรุปว่าอัลกอริทึมที่นำเสนอได้เพิ่มประสิทธิภาพของวิธีการแบบดั้งเดิม ส่งผลให้ความแม่นยำของการจำแนกประเภทมากขึ้น

3.9 วิธีการให้น้ำหนักกับระยะทางแบบใหม่สำหรับการจำแนกการจำแนกประเภทโดยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว (A New Distance-weighted k-nearest Neighbor Classifier)

ในงานวิจัยชิ้นนี้ผู้วิจัยได้รับแรงบันดาลใจจากปัญหาที่มักถูกกล่าวถึงของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว (KNN) คือการเลือกค่า K ซึ่งจะระบุถึงจำนวนเพื่อนบ้านที่ใกล้ที่สุด K ตัวในการจำแนกประเภทให้กับข้อมูลทดสอบ โดยอัลกอริทึมใหม่ที่นำเสนอใช้อัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนักกับระยะทางแบบใหม่ (Distance-weighted k-nearest Neighbor หรือ DWKNN) ซึ่งใช้วิธีการให้น้ำหนักกับระยะทางแบบคู่ (Dual Distance-Weighted) บนพื้นฐานของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนักกับระยะทาง (WKNN) วิธีการแบบใหม่นี้ผู้วิจัยได้ใช้การให้น้ำหนักระยะทางแบบคู่เพื่อกำหนดคลาสคำตอบให้กับข้อมูลทดสอบโดยใช้คลาสคำตอบของเพื่อนบ้านที่มีผลรวมค่าน้ำหนักมากที่สุดในการกำหนดคำตอบ การพัฒนาอัลกอริทึมที่นำเสนอนี้ได้รับอิทธิพลมาจากงานวิจัยของ Dudani [11] ซึ่งได้นำเสนอการกำหนดคลาสคำตอบให้กับข้อมูลทดสอบสำหรับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว โดยใช้กฎที่เรียกว่า การให้น้ำหนักกับระยะทางของเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว (WKNN) ในอัลกอริทึม WKNN เพื่อนบ้านที่อยู่ใกล้จะมีค่าน้ำหนักมากกว่าเพื่อนบ้านที่อยู่ไกลกว่า โดยใช้การคำนวณด้วยฟังก์ชันหาค่าน้ำหนักจากระยะทาง w_i สำหรับเพื่อนบ้านที่ใกล้ที่สุดตัวที่ i ของข้อมูลทดสอบ ตามสมการที่ 3.30 ดังนี้

$$\overline{w_i} = \begin{cases} \dfrac{d(\overline{x}, x_k^{NN}) - d(\overline{x}, x_i^{NN})}{d(\overline{x}, x_k^{NN}) - d(\overline{x}, x_1^{NN})}, & if\ d(\overline{x}, x_k^{NN}) \neq d(\overline{x}, x_1^{NN}) \\ 1, & if\ d(\overline{x}, x_k^{NN}) = d(\overline{x}, x_1^{NN}) \end{cases} \quad (3.30)$$

จากสมการที่ 3.30 จะเห็นได้ว่าเพื่อนบ้านที่มีระยะทางน้อยกว่าจะมีน้ำหนักมากกว่าเพื่อนบ้านที่มีระยะห่างมากกว่า เพื่อนบ้านที่ใกล้ที่สุดมีจะมีค่าน้ำหนักของระยะทางเท่ากับ 1 ส่วนเพื่อนบ้านที่อยู่ไกลที่สุดจะมีค่าน้ำหนักของระยะทางเท่ากับ 0 และค่าน้ำหนักของเพื่อนบ้านตัวอื่น ๆ จะถูกปรับให้สัมพันธ์กับระยะทางระหว่างข้อมูลที่ทำการทดสอบ อัลกอริทึม WKNN นี้ได้แก้ปัญหาของการเลือกค่า K ในอัลกอริทึมแบบดั้งเดิมซึ่งกำหนดคลาสคำตอบจากคลาสคำตอบส่วนใหญ่ของเพื่อนบ้านที่อยู่ใกล้ที่สุด เนื่องจากวิธีการแบบดั้งเดิมอาจจะมีปัญหาเมื่อกลุ่มเพื่อนบ้านที่ใกล้ที่สุดมีคลาสคำตอบแตกต่างกันอย่างมากและเพื่อนบ้านที่ใกล้ที่สุดเพียงตัวเดียวอาจจะบ่งบอกคลาสคำตอบของข้อมูลทดสอบได้ถูกต้อง อัลกอริทึม WKNN ได้ออกแบบมาเพื่อแก้ปัญหาดังกล่าวนี้แต่ถึงอย่างนั้นอัลกอริทึม WKNN ก็ยังมีปัญหาจากการกำหนดค่า K เนื่องจากชุดข้อมูลการเรียนรู้ในการจำแนกประเภทมีคลาสคำตอบที่

ไม่สมดุลบางคลาสคำตอบจะมีจำนวนมากกว่าคลาสคำตอบอื่น ด้วยเหตุนี้ทำให้การจำแนกประเภทคลาสคำตอบของแต่ละข้อมูลทดสอบไม่น่าเชื่อถือ ส่งผลให้คลาสคำตอบนั้นอาจขึ้นอยู่กับคลาสคำตอบที่ไม่ถูกต้องเนื่องจากผลรวมค่าน้ำหนักของคลาสคำตอบที่ไม่สมดุล จากปัญหาดังที่กล่าวไว้ข้างต้นผู้วิจัยจึงนำเสนออัลกอริทึม DWKNN ซึ่งเป็นการให้ค่าน้ำหนักกับระยะทางแบบใหม่โดยให้น้ำหนักระยะทางแบบคู่แทนการให้น้ำหนักตามระยะทางแบบที่ใช้ในอัลกอริทึม WKNN การให้น้ำหนักระยะทางแบบคู่สามารถคำนวณโดยการคูณน้ำหนักของระยะทางเดิมจากในสมการที่ 3.30 โดยผลรวมของระยะทางซึ่งผกผันกับค่าที่ใช้ในการคำนวณน้ำหนัก

อัลกอริทึม DWKNN มีวิธีการคำนวณซึ่งมีที่มาจากอัลกอริทึม WKNN เพื่อให้น้ำหนักกับระยะทางที่แตกต่างกันแก่เพื่อนบ้านที่ใกล้ที่สุดตามระยะทางของเพื่อนบ้านเหล่านั้นด้วยการให้น้ำหนักกับระยะทางแบบคู่ตามสมการที่ 3.31

$$\overline{w}_i = \begin{cases} \dfrac{d(\overline{x}, x_k^{NN}) - d(\overline{x}, x_i^{NN})}{d(\overline{x}, x_k^{NN}) - d(\overline{x}, x_1^{NN})} \times \dfrac{d(\overline{x}, x_k^{NN}) + d(\overline{x}, x_1^{NN})}{d(\overline{x}, x_k^{NN}) + d(\overline{x}, x_i^{NN})}, & if\ d(\overline{x}, x_k^{NN}) \neq d(\overline{x}, x_1^{NN}) \\ 1, & if\ d(\overline{x}, x_k^{NN}) = d(\overline{x}, x_1^{NN}) \end{cases} \quad (3.31)$$

จากสมการ กำหนดให้ $T = \{x_i^{NN}, y_i^{NN}\}$ ซึ่งจะประกอบด้วยเพื่อนบ้านที่ใกล้ที่สุด K ตัวของข้อมูลทดสอบ $\overline{x}$ ซึ่งจะจัดเรียงตามลำดับระยะทาง $d(\overline{x}, x_i^{NN})$ จากน้อยไปหามากระหว่างข้อมูลทดสอบและเพื่อนบ้าน กำหนด $\overline{W} = \{\overline{w}_1, \ldots, \overline{w}_k\}$ คือเซตของค่าน้ำหนักด้วยระยะทางแบบคู่ที่สอดคล้องกัน โดยสามารถแบ่งขั้นตอนการทำงานของอัลกอริทึม DWKNN ได้ทั้งหมด 5 ขั้นตอนดังนี้

ขั้นตอนที่ 1: คำนวณระยะทางเพื่อค้นหาเพื่อนบ้านที่ใกล้ที่สุดของข้อมูลทดสอบจากชุดข้อมูลการเรียนรู้ด้วยวิธีการแบบยุคลิด

ขั้นตอนที่ 2: จัดเรียงระยะทางตามลำดับจากน้อยไปมาก

ขั้นตอนที่ 3: ระบุเพื่อนบ้านที่ใกล้ที่สุด K ตัวเพื่อพิจารณาคลาสคำตอบของข้อมูลทดสอบ

ขั้นตอนที่ 4: คำนวณน้ำหนักของระยะทางด้วยระยะทางแบบคู่ของเพื่อนบ้านที่ใกล้ที่สุด K ตัวเหล่านั้นตามสมการที่ 3.31

ขั้นตอนที่ 5: กำหนดคลาสคำตอบให้กับข้อมูลทดสอบด้วยคลาสคำตอบซึ่งมีผลรวมของค่าน้ำหนักในแต่ละคลาสของเพื่อนบ้านที่มากที่สุดด้วยสมการที่ 3.32

$$\overline{y} = argmax_y \sum_{(x_i^{NN}, y_i^{NN}) \in \overline{T}} \overline{w}_i \times \delta(y = y_i^{NN}) \quad (3.32)$$

จะสังเกตได้ว่าในการพัฒนาวิธีการให้น้ำหนักกับระยะทางแบบคู่เป็นการพัฒนาต่อยอดมาจากแนวคิดพื้นฐานของการให้น้ำหนักระยะทางเหมือนในอัลกอริทึม WKNN แต่ในอัลกอริทึมที่นำเสนอผู้วิจัยได้นำเอาการคำนวณค่าน้ำหนักแบบใหม่หรือวิธีการให้น้ำหนักกับระยะทางแบบคู่เข้ามาแทนที่ โดยทั้ง 2 อัลกอริทึมสร้างขึ้นเพื่อแก้ปัญหาการเลือกค่า K ที่มักถูกกล่าวถึงของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม แต่ในอัลกอริทึมที่นำเสนอแบบใหม่จะลดน้ำหนักของเพื่อนบ้านที่ใกล้ที่สุดแต่ละตัวเพื่อป้องกันไม่ให้น้ำหนักมากเกินไป จากสมการที่ 3.31 เห็นได้ว่าค่าน้ำหนักที่เกิดจากการคำนวณระยะทางแบบคู่จะมีค่าน้อยกว่าค่าน้ำหนักที่คำนวณด้วยวิธีการ WKNN ในสมการที่ 3.30 ยกเว้นน้ำหนักของเพื่อนบ้านตัวที่ใกล้ที่สุดและตัวที่ K ซึ่งไกลที่สุดจะเป็นค่าเดียวกันกับในอัลกอริทึม WKNN ด้วยเหตุนี้เพื่อนบ้านที่มีลักษณะใกล้เคียงที่สุดจะมีอิทธิพลตามการปรับค่าของการให้น้ำหนักแบบคู่ โดยค่าน้ำหนักของระยะทางสำหรับเพื่อนบ้านที่ใกล้ที่สุดมีค่าเท่ากับ 1 และเพื่อนบ้านตัวที่ K จะมีค่าน้ำหนักของระยะทางเป็น 0 ค่าน้ำหนักของเพื่อนบ้านตัวอื่น ๆ ซึ่งถูกปรับให้สัมพันธ์กับระยะทางจะมีค่าน้ำหนักที่น้อยลงเพื่อให้อิทธิพลต่อข้อมูลทดสอบมีค่าลดลง นอกจากนี้อัลกอริทึม DWKNN ยังสามารถจัดการกับค่าผิดปกติเพื่อให้อิทธิพลในการเลือกค่า K ลดลงได้ จากผลการทดลองในงานวิจัยชิ้นนี้ชี้ให้เห็นว่าอัลกอริทึมที่นำเสนอใหม่เป็นอัลกอริทึมที่สามารถนำไปใช้กับข้อมูลในสถานการณ์จริงได้เมื่อเปรียบเทียบกับอัลกอริทึมแบบดั้งเดิมและอัลกอริทึมที่มีความเกี่ยวข้อง

3.10 วิธีการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนักกับคุณลักษณะ (Feature-weighted k-Nearest Neighbor Classifier)

ในงานวิจัยชิ้นนี้ได้นำเสนอวิธีการให้ค่าน้ำหนักกับคุณลักษณะตามการคำนวณค่าทดสอบทางสถิติไคสแควร์ (Chi-Square Test หรือ x^2 Test) เพื่อใช้ร่วมกับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว โดยในงานวิจัยได้กล่าวถึงปัญหาของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว เนื่องจากชุดข้อมูลที่มีคุณลักษณะซ้ำซ้อนหรือไม่เกี่ยวข้องกับการจำแนกประเภทจำนวนมาก ส่งผลให้เกิดข้อผิดพลาดในการจำแนกประเภทตามมา ในงานวิจัยได้กล่าวถึงความสำคัญของคุณลักษณะที่เกี่ยวข้องกับการทำการจำแนกประเภทโดยจากศึกษาทำให้ผู้วิจัยเข้าใจถึงกระบวนการเมื่อระยะทางในทั้งสองคุณลักษณะมีค่าเท่ากัน ค่าความเกี่ยวข้องของคุณลักษณะหรือค่าน้ำหนักจะมีบทบาทสำคัญในกระบวนการจำแนกประเภทนอกเหนือจากการวัดระยะทางเพียงอย่างเดียว ผู้วิจัยจึงได้นำเอาวิธีการให้น้ำหนักกับคุณลักษณะมาใช้เพื่อแก้ปัญหาดังกล่าว ซึ่งในงานวิจัยชิ้นนี้ผู้วิจัยได้มีเป้าหมายหลักคือการเพิ่มประสิทธิภาพของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม (Traditional KNN) โดยใช้การหาค่าน้ำหนักของคุณลักษณะด้วยวิธีการแบบไคสแควร์ในการจำแนกประเภทข้อมูล โดยใช้แนวคิดของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนัก (Weighted KNN) ตามงานวิจัยของ Mitchell ในปี 1997 [24] ซึ่งใช้การคำนวณระยะทางระหว่างข้อมูลทดสอบและข้อมูลการเรียนรู้ร่วมกับค่าน้ำหนักดังสมการที่ 3.33

$$d(x_i, x_j) = \sum_{r=1}^{m} w_i \left(a_r(x_i) - a_r(x_j)\right) \qquad (3.33)$$

วิธีการให้ค่าน้ำหนักกับคุณลักษณะตามการคำนวณค่าทดสอบทางสถิติไคสแควร์ (x^2FW) ได้ใช้พื้นฐานการคำนวณจากค่าทดสอบทางสถิติไคสแควร์ระหว่างข้อมูลของแต่ละคุณลักษณะและคลาสคำตอบของชุดข้อมูลในการเรียนรู้ เพื่อกำหนดค่าน้ำหนัก x^2FW การคำนวณค่าทดสอบทางสถิติไคสแควร์ ในขั้นตอนแรกต้องกำหนดคะแนนระหว่างแต่ละคุณลักษณะและคลาสคำตอบของชุดข้อมูลในการเรียนรู้ตามการคำนวณในสมการที่ 3.34

$$x^2 = \sum_{i=1}^{l} \sum_{j=1}^{c} \frac{(n_{ij} - e_{ij})^2}{e_{ij}} \qquad (3.34)$$

โดยจากสมการที่ 3.34 วิธีการคำนวณจะทำการพิจารณาตัวแปรจากข้อมูล 2 ข้อมูล ซึ่งจะประกอบด้วยตัวแปรไม่ต่อเนื่อง (discrete variable) i จำนวน l ข้อมูล , ข้อมูลตัวแปรไม่ต่อเนื่อง j จำนวน c ข้อมูล , ความถี่ที่ได้จากการสังเกต (Observed Frequency) n_{ij} และความถี่ที่คาดหวัง (Expected Frequency) e_{ij} จากนั้นทำการคำนวณค่า x^2 เพื่อวัดว่าตัวแปรสองตัวระหว่างตัวแปร i และ j นี้มีความเกี่ยวข้องหรือสัมพันธ์กันหรือไม่ ในอัลกอริทึมที่นำเสนอจะทำการคำนวณค่าน้ำหนักของแต่ละคุณลักษณะกับคลาสคำตอบเพื่อหาความสัมพันธ์ ซึ่งสามารถแสดงขั้นตอนการทำงานได้ดังรูปที่ 3.9

$x^2FW\ algorithm$

Input: dataset described by features
plus class, criterion;
Output: Weight Vector (V)
Chi: vector with the X score for eachfeature;
V: weight vector;
begin
for each feature f do
    Chi[f] v_ X (class,f,dataset);
    V <- create_vector(criterion,Chi);
end.

รูปที่ 3.9 ขั้นตอนการคำนวณค่าน้ำหนักของคุณลักษณะด้วยค่าทางสถิติไคสแควร์

หลังจากการคำนวณค่าในขั้นต้นคะแนน x^2 ระหว่างแต่ละคุณลักษณะและคลาสคำตอบแล้วอัลกอริทึมจะทำการจัดอันดับคุณลักษณะ โดยคุณลักษณะที่ค่า x^2 ต่ำสุดจะมีค่าน้ำหนักถูกกำหนดให้มีค่าเป็น 1 คุณลักษณะซึ่งมีคะแนนน้อยที่สุดเป็นอันดับสองมีค่าน้ำหนักถูกกำหนดให้มีค่าเป็น 2 และคุณลักษณะต่อมาค่านำหนักก็จะถูกกำหนดให้เป็นค่าถัดมาตามลำดับจากนั้นทำการปรับค่าน้ำหนักเป็นมาตรฐาน (Normalized Weighting) ให้เป็นค่าที่อยู่ในช่วง 0 ถึง 10 และสุดท้ายน้ำหนักของแต่ละคุณลักษณะจะถูกนำไปใช้ในการคำนวณระยะทางของด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนักแล้วกำหนดประเภทของคลาสคำตอบให้กับข้อมูลทดสอบด้วยคำตอบส่วนใหญ่ของกลุ่มเพื่อนบ้านที่ใกล้ที่สุดจำนวน K ตัว โดยในงานวิจัยได้พบว่าอัลกอริทึมการให้น้ำหนักกับคุณลักษณะจะทำการกำหนดค่าน้ำหนักที่มีค่าต่ำให้กับคุณลักษณะที่มีความสำคัญน้อยหรือให้ข้อมูลที่ไม่จำเป็นสำหรับการจำแนกประเภทและค่าน้ำหนักที่สูงขึ้นสำหรับคุณลักษณะที่มีความสำคัญหรือจำเป็นในการจำแนกประเภทนั่นเอง

บทที่ 4
งานวิจัยที่นำเสนอ

การจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมได้ใช้วิธีการคำนวณระยะทางในแต่ละคุณลักษณะเหมือนกันเนื่องจากอัลกอริทึมแบบดั้งเดิมมีสมมติฐานที่ว่าแต่ละคุณลักษณะมีความสำคัญเท่ากันทั้งหมด จากงานวิจัยที่เกี่ยวข้องซึ่งได้กล่าวถึงในบทที่ 3 จะเห็นได้ว่าเมื่อใช้การคำนวณระยะทางแบบยุคลิดเพียงอย่างเดียวกับคุณลักษณะทั้งหมดเพื่อการจำแนกประเภทจะเกิดปัญหาเมื่ออิทธิพลจากแต่ละคุณลักษณะมีความสำคัญแตกต่างกันและส่งผลต่อความแม่นยำในการจำแนกประเภท การแก้ปัญหาที่ได้ผลวิธีการหนึ่งคือนำเอาวิธีการให้น้ำหนักกับคุณลักษณะมาใช้เพื่อแก้ปัญหาของอัลกอริทึมแบบดั้งเดิม ในงานวิจัยของ Li และคณะ [6] และ งานวิจัยของ Huang และคณะ [7] ได้พิจารณาความสำคัญของแต่ละคุณลักษณะด้วยการระบุค่าน้ำหนักให้กับคุณลักษณะนั้น ซึ่งจะส่งผลให้อัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวสามารถจำแนกประเภทอย่างมีประสิทธิภาพ ในอัลกอริทึมที่มีการปรับปรุงจะกำหนดค่า w_i ซึ่งแสดงถึงน้ำหนักของคุณลักษณะที่ i ตามค่าความสำคัญ การคำนวณระยะทางในอัลกอริทึมเพื่อนบ้านใกล้ที่สุด K ตัวเมื่อมีการให้ค่าน้ำหนักกับคุณลักษณะจะนำเอาค่าน้ำหนักของแต่ละคุณลักษณะมาคูณเข้ากับระยะทางในแต่ละคุณลักษณะ โดยสามารถแสดงได้ดังสมการที่ 4.1

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2 \cdot w_i} \qquad (4.1)$$

การนำเอารูปแบบของสมการที่ 4.1 มาใช้ในงานวิจัยของ Li และคณะ [6] และงานของ Huang และคณะ [7] จะนำค่าน้ำหนักที่คำนวณจากการวัดผลประสิทธิภาพของการจำแนกประเภทเมื่อมีการนำเอาแต่ละคุณลักษณะออกในแต่ละครั้งมาใช้ร่วมกับการคำนวณระยะทางเพื่อค้นหาเพื่อนบ้านที่ใกล้ที่สุด K ตัวของข้อมูลทดสอบ ยิ่งไปกว่านั้นการค้นหาค่าน้ำหนักที่เหมาะสมสำหรับคุณลักษณะยังเป็นสิ่งที่นักวิจัยหลายคนยังคงตั้งคำถาม โดยจะเห็นได้จากงานวิจัยจำนวนมากได้กล่าวถึงการใช้วิธีการหาค่าความสัมพันธ์ของข้อมูลที่มีอยู่ในการค้นหาค่าน้ำหนักของคุณลักษณะ ในงานวิจัยของ Diego และคณะ [8] ได้กำหนดน้ำหนักของคุณลักษณะด้วยค่าทดสอบทางสถิติไคสแควร์ (Chi-Square Test) หรือในงานวิจัยของ Xiao และคณะ [9] และงานวิจัยของ Taneja และคณะ [13] ได้นำเอากำหนดค่าน้ำหนักของคุณลักษณะด้วยการคำนวณค่าเกนความรู้ระหว่างแต่ละคุณลักษณะกับชุดคลาสคำตอบของข้อมูล ตัวอย่างข้อมูลในรูปที่ 4.1 จะแสดงถึงปัญหาในการจำแนกประเภทเมื่อคุณลักษณะมี

ความสำคัญที่ต่างกันส่งผลถึงความแม่นยำของการจำแนก และเมื่อมีการนำค่าน้ำหนักมาใช้ระบุถึงความสำคัญของคุณลักษณะจะสามารถทำให้การจำแนกประเภทมีความถูกต้องมากยิ่งขึ้นดังตัวอย่าง

Id	X	Y	class
1	11	2	A
2	7	3	A
3	8	7	B
4	10	1	B
5	11	5	A
6	8	8	B

X	Y	class
9	4	B

ข้อมูลใหม่ที่เราไม่ทราบคลาสคำตอบ

กำหนดให้ค่าน้ำหนักของ
คุณลักษณะ
X = 1 และ Y = 0

รูปที่ 4.1 ตัวอย่างปัญหาในอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม

จากรูปที่ 4.1 เมื่อเราต้องการทราบว่าข้อมูลใหม่จะถูกจำแนกเป็นข้อมูลประเภทใดภายใต้เงื่อนไข K = 3 พบว่าการคำนวณระยะทางแบบยุคลิดกับชุดข้อมูลสำหรับเรียนรู้จะได้ค่าระยะทางดังนี้

$$\text{Id 1} = \sqrt{(11-9)^2 + (2-4)^2} = \sqrt{8}$$
$$\text{Id 2} = \sqrt{(7-9)^2 + (3-4)^2} = \sqrt{5}$$
$$\text{Id 3} = \sqrt{(8-9)^2 + (7-4)^2} = \sqrt{10}$$
$$\text{Id 4} = \sqrt{(10-9)^2 + (1-4)^2} = \sqrt{10}$$
$$\text{Id 5} = \sqrt{(11-9)^2 + (5-4)^2} = \sqrt{5}$$
$$\text{Id 6} = \sqrt{(8-9)^2 + (8-4)^2} = \sqrt{17}$$

หากข้อมูลจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมจะสรุปว่าข้อมูลใหม่เป็นข้อมูลประเภท A ตามคลาสคำตอบส่วนใหญ่ของเพื่อนบ้านที่ใกล้ที่สุดทั้ง 3 ตัว แต่จะเห็นได้ว่าประเภทคำตอบจริงของข้อมูลทดสอบคือข้อมูลประเภท B ปัญหาในการจำแนกประเภทนี้เนื่องมาจากอัลกอริทึมแบบดั้งเดิมมองความสำคัญของคุณลักษณะทั้งสองมีค่าเท่ากัน แต่หากมีการพิจารณาความสำคัญของแต่ละคุณลักษณะตามน้ำหนักที่กำหนดโดยคำนวณระยะทางใหม่จากสมการที่ 4.1 จะได้ค่าระยะทางดังนี้

$$\text{Id 1} = \sqrt{(11-9)^2 \times 1 + (2-4)^2 \times 0} = \sqrt{4}$$
$$\text{Id 2} = \sqrt{(7-9)^2 \times 1 + (3-4)^2 \times 0} = \sqrt{4}$$
$$\text{Id 3} = \sqrt{(8-9)^2 \times 1 + (7-4)^2 \times 0} = \sqrt{1}$$
$$\text{Id 4} = \sqrt{(10-9)^2 \times 1 + (1-4)^2 \times 0} = \sqrt{1}$$

Id 5 = $\sqrt{(11-9)^2 \times 1 + (5-4)^2 \times 0} = \sqrt{4}$

Id 6 = $\sqrt{(8-9)^2 \times 1 + (8-4)^2 \times 0} = \sqrt{1}$

จากค่าระยะห่างที่คำนวณใหม่พบว่า เมื่อทำการเรียงลำดับเพื่อนบ้านที่ใกล้ที่สุดแล้วข้อมูลใหม่จะถูกจำแนกเป็นข้อมูลประเภท B ซึ่งเป็นประเภทตามความจริงของข้อมูลที่ถูกจำแนก

จากตัวอย่างและงานวิจัยที่เกี่ยวข้อง จะเห็นได้ว่าวิธีการให้น้ำหนักสามารถเพิ่มประสิทธิภาพของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมได้ ผู้วิจัยจึงนำเอาวิธีการให้น้ำหนักกับคุณลักษณะมาใช้ร่วมกับการจำแนกประเภทแบบอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวโดยมีเป้าหมายในการค้นหาค่าน้ำหนักของคุณลักษณะที่เหมาะสม ดังนั้นอัลกอริทึมที่นำเสนอจะต้องมีวิธีการปรับละเอียดมาใช้งานแต่จะต้องป้องกันไม่ให้ใช้เวลาในการค้นหาค่าน้ำหนักที่เหมาะสมมากเกินไป ผู้วิจัยจึงให้ค่าน้ำหนักกับคุณลักษณะเบื้องต้น หรือ ค่าน้ำหนักแบบหยาบด้วยค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน เนื่องจากวิธีการนี้สามารถใช้ในการหาความสัมพันธ์ของคุณลักษณะของข้อมูลแต่ละตัวกับคลาสคำตอบของข้อมูล ผู้วิจัยได้นำค่าน้ำหนักแบบหยาบมาทดลองกับชุดข้อมูลสาธารณะ UCI บางชุดข้อมูลด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนักกับคุณลักษณะโดยไม่มีวิธีการปรับละเอียด ซึ่งผลการทดลองแสดงไว้ในตารางที่ 4.1

ตารางที่ 4.1 ผลความแม่นยำในการจำแนกประเภทของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยการให้น้ำหนักแบบหยาบจากค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน

Dataset	K	Traditional KNN	Weighted Pearson KNN (NON - PSO)
Hepatitis	3	69.44	88.89
	5	77.78	80.56
	7	86.11	83.33
Heart Statlog	3	80.88	80.88
	5	80.88	83.82
	7	83.82	80.88
Movement Libras	3	77.78	82.22
	5	75.56	76.67
	7	72.22	72.22

ตารางที่ 4.2 ผลความแม่นยำในการจำแนกประเภทของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว ด้วยการให้น้ำหนักแบบหยาบจากค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน (ต่อ)

Dataset	K	Traditional KNN	Weighted Pearson KNN (NON - PSO)
Ionosphere	3	82.95	86.36
	5	81.82	85.23
	7	80.68	85.23
Telecom	3	85.03	86.83
	5	86.83	89.82
	7	87.43	89.82
HCV	3	88.96	88.31
	5	88.96	88.96
	7	88.31	89.61

จากผลการทดลองจะเห็นว่า การนำค่าน้ำหนักค่าน้ำหนักแบบหยาบมาใช้สามารถปรับปรุงความแม่นยำในการจำแนกประเภทของอัลกอริทึมแบบดั้งเดิมได้ในค่า K ส่วนใหญ่ของชุดข้อมูลที่นำมาทดสอบ โดยเฉพาะอย่างยิ่งในชุดข้อมูล Telecom ซึ่งสามารถปรับปรุงความแม่นยำในทุกค่า K ด้วยเหตุนี้ผู้วิจัยจึงคิดว่าการกำหนดค่าน้ำหนักของคุณลักษณะโดยประมาณจากค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันจะช่วยลดเวลาในการค้นหาค่าน้ำหนักที่เหมาะสมด้วยวิธีการ PSO จึงเป็นที่มาของงานวิจัยที่นำเสนอ **การปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวนเคตัวด้วยการให้น้ำหนักกับคุณลักษณะของข้อมูลจากค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันและการปรับละเอียดด้วยอัลกอริทึมพีเอสโอ (Improving KNN Algorithm based on Weighted Attributes by Pearson Correlation Coefficient and PSO Fine Tuning)** ซึ่งสามารถอธิบายการทำงานเป็น 2 ขั้นตอนได้ดังนี้

1. การให้ค่าน้ำหนักกับคุณลักษณะแบบหยาบเพื่อคำนวณระยะทางในการค้นหาเพื่อนบ้านที่ใกล้ที่สุดจำนวน K ตัว
2. การปรับละเอียดค่าน้ำหนักเพื่อใช้ในการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนักกับคุณลักษณะ

4.1 การให้ค่าน้ำหนักกับคุณลักษณะแบบหยาบเพื่อคำนวณระยะทางในการค้นหาเพื่อนบ้านที่ใกล้ที่สุดจำนวน K ตัว

ขั้นตอนที่ 1 แบ่งชุดข้อมูลออกเป็นชุดข้อมูลการเรียนรู้และชุดข้อมูลการทดสอบ โดยสำหรับชุดข้อมูลการเรียนรู้อัลกอริทึมจะกำหนดให้ A เป็นเซตซึ่งประกอบด้วยเวกเตอร์ของคุณลักษณะข้อมูลทั้งหมด n คุณลักษณะ $A = \{a_1, a_2, a_3, \dots, a_n\}$ และกำหนด C เป็นเวกเตอร์คลาสคำตอบ

ขั้นตอนที่ 2 คำนวณความสัมพันธ์ระหว่างแต่ละชุดข้อมูลของเวกเตอร์คุณลักษณะใน A และเวกเตอร์คลาสคำตอบ C โดยใช้วิธีการหาสัมประสิทธิ์สหสัมพันธ์เพียร์สัน แล้วนำค่าดังกล่าวมาหาค่าสัมบูรณ์ (absolute value) เนื่องจากต้องการค่าที่บ่งถึงระดับความสัมพันธ์โดยไม่นำทิศทางความสัมพันธ์ของข้อมูลมาใช้ ซึ่งจะได้เซตของสัมประสิทธิ์ที่เป็นค่าบวก $P = \{pc_1, pc_2, pc_3, \dots, pc_n\}$

ขั้นตอนที่ 3 คำนวณค่าน้ำหนักแบบหยาบ w_i ของแต่ละคุณลักษณะเพื่อใช้เป็นค่าน้ำหนักเริ่มต้นโดยวิธีการปรับมาตรฐาน (Normalizing) จากสัมประสิทธิ์สหสัมพันธ์เพียร์สันแต่ละค่าในชุดข้อมูล P หารด้วยผลรวมของค่าสัมประสิทธิ์ทั้งหมดเพื่อให้ผลรวมของทั้งหมดน้ำหนักเมื่อปรับมาตรฐานแล้วเท่ากับ 1 ด้วยสมการที่ 4.2

$$w_i = \frac{pc_i}{\sum_{i=1}^{n} pc_i} \tag{4.2}$$

4.2 การปรับละเอียดค่าน้ำหนักเพื่อใช้ในการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนักกับคุณลักษณะ

ในขั้นตอนนี้วิธีการที่นำเสนอจะใช้อัลกอริทึม PSO เพื่อค้นหาค่าน้ำหนักที่เหมาะสมที่สุด โดยใช้ความแม่นยำในการจำแนกประเภทเป็นฟังก์ชั่นการหาค่าที่ดีที่สุด ซึ่งในหาคำนวณระยะทาง หรือ $d(p, q)$ เราใช้สมการหาระยะทางแบบยุคลิดร่วมกับค่าน้ำหนักของแต่ละคุณลักษณะซึ่งสามารถแสดงได้ตามสมการที่ 4.3

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2 \,.\, w_i^2} \tag{4.3}$$

การคำนวณจะนำเอาค่าน้ำหนักของแต่ละคุณลักษณะมาคูณเข้ากับระยะทางในแต่ละคุณลักษณะ โดยในอัลกอริทึมที่ได้นำเสนอผู้วิจัยได้คำนวณระยะทางจากสมการด้วยการกำหนดค่าน้ำหนักในรูปกำลัง

สองหรือ w_i^2 เพื่อให้ค่าน้ำหนักของคุณลักษณะมีผลมากยิ่งขึ้นในการคำนวณระยะทาง จากนั้นเมื่อได้ระยะห่างระหว่างข้อมูลใหม่ที่ยังไม่ถูกจำแนกและข้อมูลการเรียนรู้ อัลกอริทึมจะทำการพิจารณาระยะทางของเพื่อนบ้านที่ใกล้ที่สุด K ตัวนั้นจำแนกประเภทให้กับข้อมูลใหม่โดยคำตอบส่วนใหญ่ หลังจากการคำนวณความแม่นยำของผลการจำแนกประเภทรอบแรกแล้วเราจะนำอัลกอริทึมการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคมาใช้งานเพื่อหาค่าน้ำหนักที่เหมาะสม ซึ่งกำหนดฟังก์ชันการหาค่าที่ดีที่สุด (fitness function) ด้วยความแม่นยำของผลการจำแนกประเภทแต่ละรอบโดยใช้ข้อมูลข้อมูลการเรียนรู้เพื่อประเมินผล ในการปรับละเอียดแต่ละรอบจะใช้สมการที่ 4.4 และ 4.5

$$V_b[i+1] = (\omega * V_b) + c_1 r_1 (GBest - X_b) + c_2 r_2 (PBest_b - X_b) \quad (4.4)$$

$$X_b[i+1] = V_b[i+1] + X_b[i] \quad (4.5)$$

โดย $b = 1,2,3,\dots,n$ คืออนุภาค

V_b คือค่า Velocity ซึ่งจะเป็นตัวบอกทิศทางและระยะการเคลื่อนที่ของอนุภาค

X_b คือตำแหน่งอนุภาคแต่ละตัว โดยตำแหน่งในการคำนวณนี้คือ

$PBest_b$ คือตำแหน่งที่ดีที่สุดของอนุภาค (ค่าน้ำหนักของแต่ละอนุภาค)

$GBest$ คือตำแหน่งที่ดีที่สุดของฝูงอนุภาค (ค่าน้ำหนักที่ดีที่สุดปรับในแต่ละรอบ)

ω คือค่าถ่วงทิศทางโดยกำหนดค่าเป็น 0.729844

r_1, r_2 คือค่าสุ่มอยู่ในช่วง [0,1]

c_1 คือ Social Parameter และ c_2 คือ Cognitive Parameter โดยกำหนดค่าเป็น 1.49445

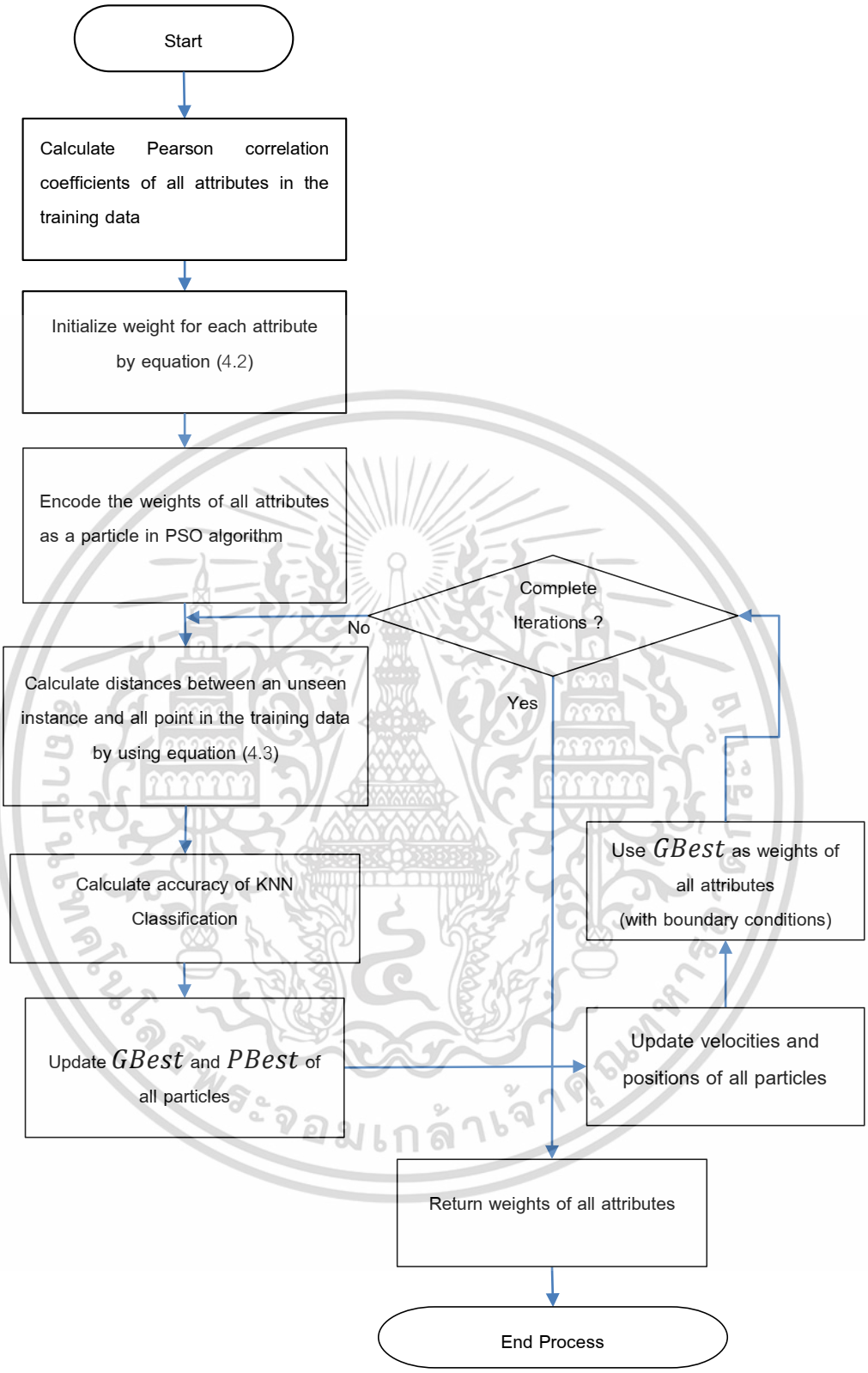
ซึ่งในการปรับตำแหน่ง (ค่าน้ำหนัก) ของอนุภาคจะมีการกำหนดขอบเขต (Boundary) เมื่อมีการเปลี่ยนแปลงค่าน้ำหนักทุกรอบการคำนวณเพื่อให้การค้นหาค่าน้ำหนักอยู่ในช่วงที่กำหนดจากค่าน้ำหนักแบบหยาบ โดยค่าที่ปรับจะต้องอยู่ระหว่างช่วงเพิ่มขึ้น 0.2 และลดลง 0.2 จากค่าเดิมดังนี้

$$(w_i - 0.2) \leq w_i \leq (w_i + 0.2)$$

หลังจากครบรอบที่กำหนดของการค้นหา อัลกอริทึมจะได้ค่าน้ำหนักที่เหมาะที่สุดจากรอบการค้นหาเพื่อนำไปใช้ในการค้นหาเพื่อนบ้านที่ใกล้ที่สุด K ตัวต่อไป รหัสเทียมของขั้นตอนการทำงานของอัลกอริทึมที่ได้นำเสนอนี้สามารแสดงได้ดังรูปที่ 4.2 และผังงานอธิบายขั้นตอนการทำงานการหาค่าน้ำหนักของอัลกอริทึมที่ได้นำเสนอนี้ได้ดังรูปที่ 4.3

```
(1)  START
(2)  //**Find the rough weight of each attributes
(3)  Choose the value of K
(4)  FOR i = 1 to  the number of attribute in training data
(5)        corr_mat = np.corrcoef(att_i, train_tar)
(6)        corr_score = abs(corr_score)
(7)        pearson_score add corr_score[i]
(8)  END FOR
(9)  FOR i = 1 to  the number of attribute in training data
(10)       Weight[i] = pearson_score[x]/pearson_score_sum
(11)       weight_pearson. add weight[i]  // to P set
(12) END FOR
(13) //**Initial particles, counter of each particle, number of iteration**
(14) FOR each particle
(15)    Initialize particle position with rough search weight
(16) END FOR
(17) WHILE (i < number of iteration)
(18) Do
(19)    For each particle
(20)       Calculate distance between test and training data with equation (4.3) and then
(21)       classify test set
(22)       Calculate Accuracy result (fitness value) // fitness function
(23)       If the Accuracy result is better than Accuracy result its old Accuracy result
(24)          set current position (weight) as the new pBest
(25)    END FOR
(26)    Choose the particle with the best fitness value of all as gBest
(27)    For each particle
(28)       Calculate particle velocity according equation (a)
(29)       Update particle position according equation (b)
(30)    END FOR
(31) RETURN gbest (set of weights) of work process
(32) Load test data
(33) For each point in test data:
(34)       - find the Euclidean distance to all training data points with equation (4.3)
(35) assign a class to the test data by the majority class of class nearest neighbors
(36) END
```

รูปที่ 4.2 รหัสจำลองแสดงการทำงานของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวโดยให้น้ำหนักกับคุณลักษณะของข้อมูลจากค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันและการปรับละเอียด



รูปที่ 4.3 ผังงานอธิบายขั้นตอนการหาค่าน้ำหนักในอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวโดยให้น้ำหนักกับคุณลักษณะของข้อมูลจากค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันและการปรับละเอียด

จากรูปที่ 4.2 รหัสจำลองแสดงการทำงานของอัลกอริทึมที่นำเสนอในช่วงเริ่มต้นของการทำงานจะเป็นการนำเข้าข้อมูลจากชุดข้อมูลการเรียนรู้ จากนั้น ในบรรทัดที่ 4 จะทำการหาค่าสัมประสิทธิ์สหสัมพันธ์เพียร์สันแล้วนำค่าดังกล่าวมาหาค่าสัมบูรณ์โดยจะได้เซตของสัมประสิทธิ์ P หลังจากนั้นในบรรทัดที่ 9 จะเป็นการคำนวณค่าน้ำหนักเริ่มต้น w_i ของแต่ละคุณลักษณะโดยวิธีการปรับมาตรฐานจากสัมประสิทธิ์สหสัมพันธ์เพียร์สันแต่ละค่า ขั้นตอนการปรับละเอียดจะเริ่มในบรรทัดที่ 14 ซึ่งใช้อัลกอริทึม PSO แบบดั้งเดิมเพื่อค้นหาค่าน้ำหนักที่เหมาะสมที่สุด โดยนำค่าน้ำหนักแบบหยาบมาใช้เป็นตำแหน่งเริ่มต้น จากนั้นใช้ความแม่นยำจากการจำแนกประเภทของข้อมูลทดสอบในชุดข้อมูลการเรียนรู้เป็นฟังก์ชั่นการหาค่าที่ดีที่สุด จนเมื่อถึงรอบที่กำหนดหรือค่าที่ค้นหาเป็นค่าที่ดีที่สุด (fitness value) อัลกอริทึมจะคืนค่าตำแหน่งที่ดีที่สุด ($GBest$) หรือก็คือค่าน้ำหนักที่ได้ทำการปรับให้เหมาะสมแล้ว อัลกอริทึมจะนำชุดข้อมูลทดสอบมาทำการจำแนกประเภทต่อไปด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบให้น้ำหนักกับคุณลักษณะในบรรทัดที่ 34 หลังจากได้เพื่อนบ้านที่ใกล้ที่สุดจำนวน K ตัวแล้ว อัลกอริทึมจะกำหนดคลาสคำตอบให้กับข้อมูลทดสอบโดยใช้คำตอบส่วนใหญ่ของเพื่อนบ้านเหล่านั้น ซึ่งหากคำตอบส่วนใหญ่มีจำนวนมากกว่าหนึ่งคลาส (จำนวนคำตอบเท่ากันในแต่ละคลาส) อัลกอริทึมจะพิจารณาจากคำตอบส่วนใหญ่ของเพื่อนบ้านที่ใกล้ที่สุด K ตัวเหล่านั้น

4.3 ตัวอย่างการคำนวณเพื่อให้น้ำหนักกับคุณลักษณะแบบหยาบและการปรับละเอียดค่าน้ำหนัก

ผู้วิจัยใช้ชุดข้อมูลสาธารณะ UCI จำนวน 2 ชุดในตัวอย่างการคำนวณคือชุดข้อมูล Vehicle silhouette และชุดข้อมูล Climate Simulation โดยจะทำการคำนวณค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันระหว่างคุณลักษณะทั้งหมด n คุณลักษณะกับเวกเตอร์คลาสคำตอบแล้วนำค่าดังกล่าวมาหาค่าสัมบูรณ์ หากคลาสคำตอบของชุดข้อมูลเป็นข้อมูลเชิงคุณภาพ (qualitative variable) วิธีการจะทำการปรับข้อมูลให้อยู่ในรูปแบบของข้อมูลตัวเลข (numeric variable) เพื่อให้สามารถนำมาคำนวณได้ โดยจะได้ผลลัพธ์ดังรูปที่ 4.4 และ รูปที่ 4.5

pc_1	pc_2	pc_3	pc_4	pc_5	pc_6	pc_7	pc_8	pc_9
0.03	0.192	0.07	0.183	0.068	0.237	0.303	0.343	0.273

pc_{10}	pc_{11}	pc_{12}	pc_{13}	pc_{14}	pc_{15}	pc_{16}	pc_{17}	pc_{18}
0.068	0.324	0.306	0.256	0.203	0.171	0.025	0.063	0.242

Vehicle silhouette

รูปที่ 4.4 ค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันระหว่างแต่ละคุณลักษณะและเวกเตอร์คลาสคำตอบของชุดข้อมูล Vehicle silhouette เมื่อนำมาหาค่าสัมบูรณ์

pc_1	pc_2	pc_3	pc_4	pc_5	pc_6	pc_7	pc_8	pc_9
0.304	0.304	0.023	0.054	0.106	0.034	0.01	0.012	0.043

pc_{10}	pc_{11}	pc_{12}	pc_{13}	pc_{14}	pc_{15}	pc_{16}	pc_{17}	pc_{18}
0.022	0.015	0.05	0.204	0.162	0.061	0.086	0.068	0.014

Climate Simulation

รูปที่ 4.5 ค่าสัมประสิทธิ์สหสัมพันธ์เพียร์สันระหว่างแต่ละคุณลักษณะและเวกเตอร์คลาสคำตอบของชุดข้อมูล Climate Simulation เมื่อนำมาหาค่าสัมบูรณ์

จากนั้นวิธีการจะคำนวณค่าน้ำหนักแบบหยาบ W_i ของแต่ละคุณลักษณะโดยวิธีการปรับมาตรฐานจากค่าสัมประสิทธิ์สหสัมพันธ์เพียร์สันแต่ละค่าด้วยสมการที่ 4.2 ซึ่งจะได้ผลลัพธ์ดังรูปที่ 4.6

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
0.009	0.057	0.021	0.055	0.02	0.071	0.09	0.102	0.081

w_{10}	w_{11}	w_{12}	w_{13}	w_{14}	w_{15}	w_{16}	w_{17}	w_{18}
0.02	0.097	0.091	0.076	0.06	0.051	0.008	0.019	0.072

Vehicle silhouette

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
0.193	0.193	0.015	0.034	0.067	0.021	0.007	0.008	0.028

w_{10}	w_{11}	w_{12}	w_{13}	w_{14}	w_{15}	w_{16}	w_{17}	w_{18}
0.014	0.009	0.032	0.13	0.103	0.039	0.055	0.043	0.009

Climate Simulation

รูปที่ 4.6 ค่าน้ำหนักแบบหยาบจากการคำนวณปรับมาตรฐานของชุดข้อมูล Vehicle silhouette และชุดข้อมูล Climate Simulation

ในขั้นตอนการปรับละเอียดซึ่งนำเอาอัลกอริทึม PSO แบบดั้งเดิมมาใช้ค้นหาค่าน้ำหนักที่เหมาะสม อัลกอริทึมจะนำค่าน้ำหนักแบบหยาบจากในรูปที่ 4.6 มาใช้เป็นตำแหน่งเริ่มต้น จากนั้นคำนวณระยะทางระหว่างข้อมูลทดสอบของชุดข้อมูลการเรียนรู้กับชุดข้อมูลการเรียนรู้เพื่อหาเพื่อนบ้านที่ใกล้ที่สุดจำนวน K ตัวด้วยสมการที่ 4.3 เมื่อได้ระยะห่างระหว่างทดสอบและข้อมูลการเรียนรู้ อัลกอริทึมจะทำการพิจารณาระยะทางของเพื่อนบ้านที่ใกล้ที่สุด K ตัวตามลำดับแล้วกำหนดคลาสคำตอบให้กับข้อมูลทดสอบโดยใช้คำตอบส่วนใหญ่ของเพื่อนบ้านเหล่านั้น ความแม่นยำของการจำแนกประเภทด้วยชุดข้อมูลการเรียนรู้จะถูกใช้เป็นค่าประเมินในฟังก์ชั่นการหาค่าที่ดีที่สุด โดยในตัวอย่างของชุดคือชุดข้อมูล Vehicle silhouette และชุดข้อมูล Climate Simulation เมื่อกำหนดค่า K = 5 จะได้ค่าน้ำหนักซึ่งผ่านการปรับละเอียดแล้วดังรูปที่ 4.7

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
0.1454	0.2571	0.2209	0.2546	0.2202	0	0.2903	0.2139	0.2813

w_{10}	w_{11}	w_{12}	w_{13}	w_{14}	w_{15}	w_{16}	w_{17}	w_{18}
0.2204	0.2967	0.291	0.2763	0	0	0	0	0

Vehicle silhouette

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
0.2839	0.0672	0	0	0.1198	0	0	0	0.2276

w_{10}	w_{11}	w_{12}	w_{13}	w_{14}	w_{15}	w_{16}	w_{17}	w_{18}
0	0.0329	0.0048	0.2223	0.0886	0.1984	0	0	0

Climate Simulation

รูปที่ 4.7 ค่าน้ำหนักแต่ละคุณลักษณะของชุดข้อมูล Vehicle silhouette และชุดข้อมูล Climate Simulation ที่ผ่านการปรับละเอียดเมื่อกำหนดค่า K = 5

จากผลลัพธ์ในรูปที่ 4.7 จะเห็นได้ว่าค่าน้ำหนักแบบหยาบในรูปที่ 4.6 มีการเปลี่ยนแปลงเมื่อผ่านการปรับละเอียดเพื่อค้นหาค่าน้ำหนักที่เหมาะสม จากนั้นอัลกอริทึมจะนำค่าน้ำหนักที่ผ่านการปรับละเอียดไปคำนวณระยะทางระหว่างข้อมูลทดสอบและชุดข้อมูลการเรียนรู้เพื่อค้นหาเพื่อนบ้านที่ใกล้ที่สุดจำนวน K ตัวด้วยสมการที่ 4.3 และทำการกำหนดคลาสคำตอบให้กับข้อมูลทดสอบโดยใช้

คำตอบส่วนใหญ่ของเพื่อนบ้าน K ตัวเหล่านั้น ซึ่งผลการจำแนกประเภทด้วยอัลกอริทึมที่นำเสนอเมื่อนำค่าน้ำหนักที่ปรับละเอียดแล้วจากรูปที่ 4.7 ไปใช้จะได้ความแม่นยำของการจำแนกเพิ่มขึ้นเมื่อนำไปเปรียบเทียบกับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม ผลการเปรียบเทียบความแม่นยำเมื่อกำหนดค่า K = 5 สามารถแสดงได้ดังตารางที่ 4.3

ตารางที่ 4.3 ผลการเปรียบเทียบความแม่นยำในการจำแนกประเภทระหว่างอัลกอริทึมที่นำเสนอโดยใช้ค่าน้ำหนักที่ผ่านการปรับละเอียดจากข้อมูลตัวอย่างกับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมเมื่อกำหนดค่า K = 5

Dataset	Traditional KNN	Weighted Pearson-PSO KNN	Percent of Acc Improvement
Vehicle silhouette	68.39	73.11	6.90%
Climate Model	90.37	92.59	2.46%

บทที่ 5

การทดลอง

ในบทนี้จะอธิบายถึงวิธีในการทดลอง ผลการทดลองการจำแนกประเภทของอัลกอริทึมที่นำเสนอและผลของอัลกอริทึมที่เกี่ยวข้อง ซึ่งจะมีการเปรียบเทียบผลลัพธ์ของแต่ละอัลกอริทึมเพื่อวัดประสิทธิภาพในการจำแนกประเภทจากชุดข้อมูลสาธารณะ UCI โดยสามารถอธิบายรายละเอียดของวิธีการทดลองได้ดังนี้

5.1 พารามิเตอร์

ค่าพารามิเตอร์ต่าง ๆ ที่ถูกใช้ในการทดลองนี้จะเป็นค่าที่ใช้ในอัลกอริทึมที่นำเสนอ โดยจะมีการกำหนดค่าพารามิเตอร์ดังตารางที่ 5.1

ตารางที่ 5.1 ค่าพารามิเตอร์พื้นฐานที่ใช้ในอัลกอริทึมการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค

พารามิเตอร์	ค่าพารามิเตอร์
จำนวนอนุภาค สำหรับ1กลุ่มอนุภาค	20 อนุภาค
ค่าความเฉื่อย ω	0.72984
ค่าคงที่การเรียนรู้ (C_1, C_2)	1.496172
ข้อมูลที่ใช้ในการทดสอบ	ตารางที่ 5.2
จำนวนรอบการค้นหาที่กำหนด (Generation)	100 รอบ

5.2 วิธีการทดลอง

งานวิจัยชิ้นนี้ได้ทำการทดลองและเปรียบเทียบเพื่อวัดประสิทธิภาพในการจำแนกระหว่างวิธีการจำแนกประเภท 3 วิธีการดังนี้

1. อัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม
2. วิธีการปรับมาตรฐานเพื่อหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคและการประยุกต์ใช้ในปัญหาการให้น้ำหนักกับคุณลักษณะ (The Normalized-PSO and Its Application in Attribute Weighted Optimal Problem)
3. วิธีการปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยการให้น้ำหนักกับคุณลักษณะของข้อมูลจากค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันและการปรับละเอียด (Improving KNN Algorithm based on Weighted Attributes by Pearson Correlation Coefficient and Fine Tuning)

5.3 ชุดข้อมูลทดสอบ

ในงานวิจัยชิ้นนี้ได้ทำการทดลองบนชุดข้อมูลสาธารณะ UCI ซึ่งสามารถอธิบายลักษณะของชุดข้อมูลแต่ละชุดตามตารางที่ 5.2

ตารางที่ 5.2 อธิบายลักษณะของชุดข้อมูลแต่ละชุดที่ใช้ในการทดลอง

Data set	Size	Number of Attributes	Number of Classes
Hepatitis	155	19	2
Heart – Statlog	270	13	2
Sonar	208	60	2
Climate Simulation	540	18	2
Movement Libras	360	90	15
Vehicle silhouette	946	18	4
Ionosphere	351	34	2
Telecom	668	19	2
Credit Approval	684	14	2
HCV dataset	615	12	5

5.4 ผลการทดลอง

ในส่วนนี้จะเป็นการแสดงผลของทั้ง 3 วิธีการ ด้วยผลการทดสอบความแม่นยำของการจำแนกประเภทด้วยชุดข้อมูลทดสอบทั้ง 10 ชุดข้อมูลในหัวข้อที่ 5.3 และเวลาที่ใช้ในการจำแนกประเภทของทั้ง 3 อัลกอริทึม โดยทำการแบ่งข้อมูลเป็นชุดข้อมูลการเรียนรู้ (training data) และชุดข้อมูลทดสอบ (test data) เป็นร้อยละ 75 และร้อยละ 25 ตามลำดับ ผลทดลองในส่วนแรกเป็นผลความแม่นยำของการจำแนกประเภทชุดข้อมูลทดสอดที่ถูกแบ่งไว้ของอัลกอริทึมที่ได้นำเสนอ (Weighted Pearson-PSO KNN) โดยข้อมูลเป็นค่าเฉลี่ยความแม่นยำของการจำแนกประเภท 5 ครั้ง ความแม่นยำของการจำแนกประเภทด้วยค่าน้ำหนักแบบหยาบเพียงอย่างเดียว (ไม่มีการปรับละเอียดค่าน้ำหนัก) และความแม่นยำสูงที่สุดจากค่าเฉลี่ย ซึ่งแสดงผลได้ดังตารางที่ 5.3 และตารางที่ 5.4

ตารางที่ 5.3 ความแม่นยำของการจำแนกประเภทด้วยชุดข้อมูลทดสอบของอัลกอริทึมที่นำเสนอ (Weighted Pearson-PSO KNN)

Dataset	K	Weighted Pearson – PSO KNN (SD)	Weighted Pearson KNN (non-PSO)	Weighted Pearson – PSO KNN (Maximum)
Hepatitis	3	91.11 (3.24)	88.89	94.44
	5	85.0 (2. 22)	80.56	88.89
	7	88.33 (2.08)	83.33	88.89
Heart Statlog	3	82.65 (2.16)	80.88	85.29
	5	84.12 (1.72)	83.82	86.77
	7	84.12 (2.16)	80.88	86.77
Sonar	3	89.62 (1.96)	82.69	92.31
	5	86.92 (1.44)	80.77	88.46
	7	86.15 (2.83)	76.92	90.38
Movement Libras	3	82.67 (1.13)	82.22	84.44
	5	81.11 (1.22)	76.67	82.22
	7	77.56 (1.47)	72.22	80.0
Climate Simulation	3	90.67 (0.89)	89.63	92.59
	5	91.41 (0.81)	89.63	92.59
	7	91.41 (0.89)	88.89	91.85
Vehicle silhouette	3	71.98 (2.96)	66.51	73.11
	5	72.83 (0.38)	72.64	73.11
	7	71.46 (2.88)	69.339	74.06

ตารางที่ 5.4 ความแม่นยำของการจำแนกประเภทด้วยชุดข้อมูลทดสอบของอัลกอริทึมที่นำเสนอ (Weighted Pearson-PSO KNN) (ต่อ)

Dataset	K	Weighted Pearson – PSO KNN (SD)	Weighted Pearson KNN (non-PSO)	Weighted Pearson – PSO KNN (Maximum)
Ionosphere	3	86.36 (1.44)	86.36	89.77
	5	85.91 (0.56)	85.23	86.36
	7	84.55 (0.91)	85.23	85.23
Telecom	3	88.38 (1.04)	86.83	89.82
	5	89.22 (0.85)	89.82	90.42
	7	90.42 (0.38)	89.82	91.02
Credit Approval	3	88.54 (2.29)	84.21	91.23
	5	87.37 (1.20)	85.38	89.47
	7	87.25 (1.01)	87.72	88.89
HCV dataset	3	90.52 (0.66)	88.31	91.56
	5	91.29 (1.34)	88.96	92.86
	7	91.17 (1.13)	89.61	92.86

จากผลของการทดสอบของอัลกอริทึมที่นำเสนอตามตารางที่ 5.3 และ 5.4 จะเห็นได้ว่าเมื่อนำการปรับละเอียดค่าน้ำหนักมาใช้งาน สามารถปรับปรุงประสิทธิภาพของอัลกอริทึมซึ่งใช้ค่าน้ำหนักแบบหยาบเพียงอย่างเดียว ประสิทธิภาพที่เพิ่มขึ้นนี้เป็นจุดที่แสดงให้เห็นว่าค่าน้ำหนักที่ผ่านการปรับละเอียดมีผลต่อการจำแนกประเภทที่แม่นยำมากยิ่งขึ้น โดยเฉพาะอย่างยิ่งเมื่อเปรียบเทียบกับความแม่นยำสูงที่สุดจากค่าเฉลี่ยในคอลัมน์ที่ 5 จะเห็นว่าอัลกอริทึมที่นำเสนอสามารถปรับปรุงประสิทธิภาพของการให้ค่าน้ำหนักแบบหยาบเพียงอย่างเดียวได้เป็นอย่างดี ผลทดลองในส่วนที่สองจะการเปรียบเทียบความแม่นยำของการจำแนกประเภทชุดข้อมูลทดสอดที่ถูกแบ่งไว้ของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม (Traditional KNN) และค่าเฉลี่ยของอัลกอริทึมที่นำเสนอ ซึ่งแสดงผลได้ดังตารางที่ 5.5 และ 5.6

ตารางที่ 5.5 แสดงผลการเปรียบเทียบความแม่นยำของการจำแนกประเภทด้วยชุดข้อมูลทดสอบระหว่างอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมและอัลกอริทึมที่นำเสนอ (Weighted Pearson-PSO KNN)

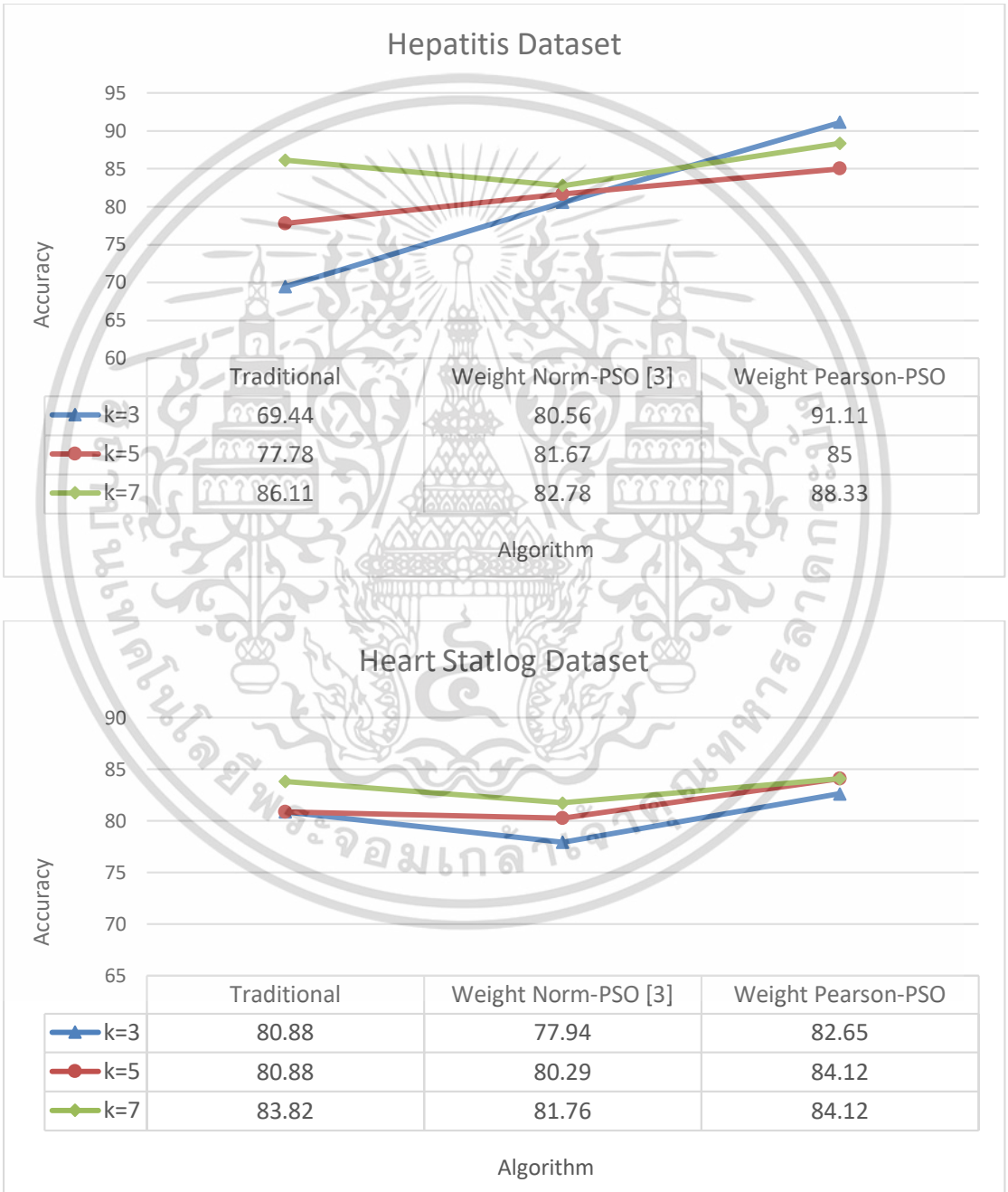
Dataset	K	Traditional KNN	Weighted Pearson – PSO KNN (SD)	Percent of Acc. Improvement
Hepatitis	3	69.44	91.11 (3.24)	24.0%
	5	77.78	85.0 (2.22)	9.29%
	7	86.11	88.33 (2.08)	2.58%
Heart Statlog	3	80.88	82.65 (2.16)	2.18%
	5	80.88	84.12 (1.72)	4.0%
	7	83.82	84.12 (2.16)	0.35%
Sonar	3	84.62	89.62 (1.96)	5.91%
	5	86.54	86.92 (1.44)	0.44%
	7	84.62	86.15 (2.83)	1.82%
Movement Libras	3	77.78	82.67 (1.13)	6.29%
	5	75.56	81.11 (1.22)	7.35%
	7	72.22	77.56 (1.47)	7.39%
Climate Simulation	3	89.63	90.67 (0.89)	1.16%
	5	90.37	91.41 (0.81)	1.15%
	7	89.63	91.41 (0.89)	1.98%
Vehicle silhouette	3	69.34	71.98 (2.96)	3.81%
	5	68.39	72.83 (0.38)	6.48%
	7	70.75	71.46 (2.88)	1.0%

ตารางที่ 5.6 แสดงผลการเปรียบเทียบความแม่นยำของการจำแนกประเภทด้วยชุดข้อมูลทดสอบระหว่างอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมและอัลกอริทึมที่นำเสนอ (Weighted Pearson-PSO KNN) (ต่อ)

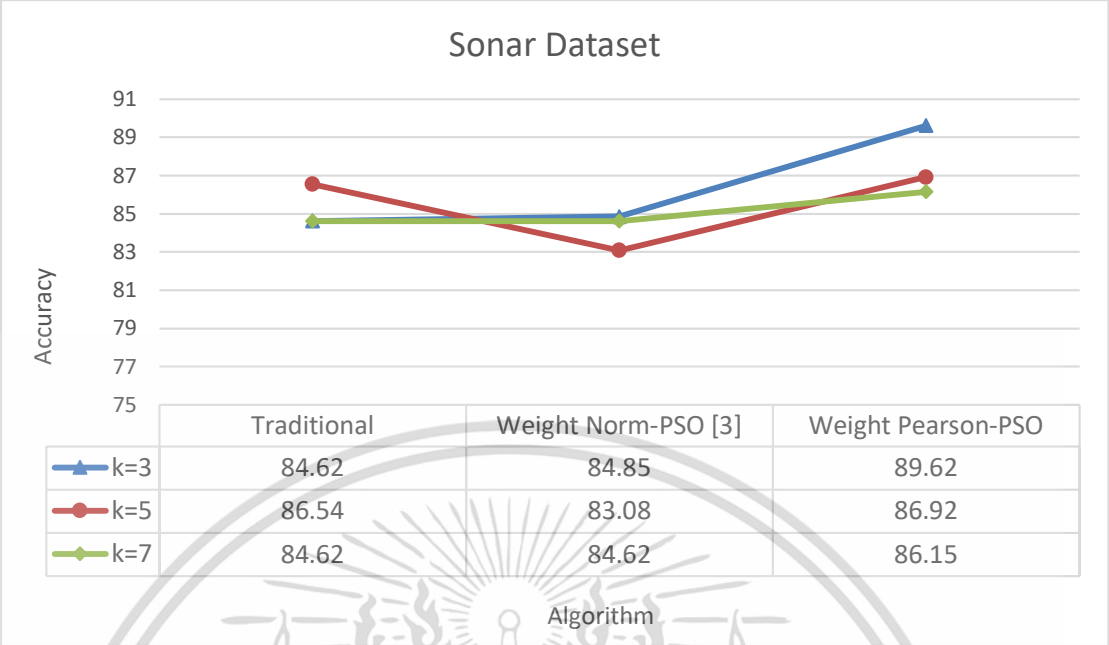
Dataset	K	Traditional KNN	Weighted Pearson – PSO KNN (SD)	Percent of Acc. Improvement
Ionosphere	3	82.954	86.36 (1.44)	4.11%
	5	81.82	85.91 (0.56)	5.00%
	7	80.68	84.55 (0.91)	4.79%
Telecom	3	85.03	88.38 (1.04)	3.94%
	5	86.83	89.22 (0.85)	2.76%
	7	87.42	90.42 (0.38)	3.42%
Credit Approval	3	85.38	88.54 (2.29)	3.70%
	5	85.96	87.37 (1.20)	1.63%
	7	86.55	87.25 (1.01)	0.81%
HCV dataset	3	88.96	90.52 (0.66)	1.75%
	5	88.96	91.29 (1.34)	2.63%
	7	88.31	91.17 (1.13)	3.24%

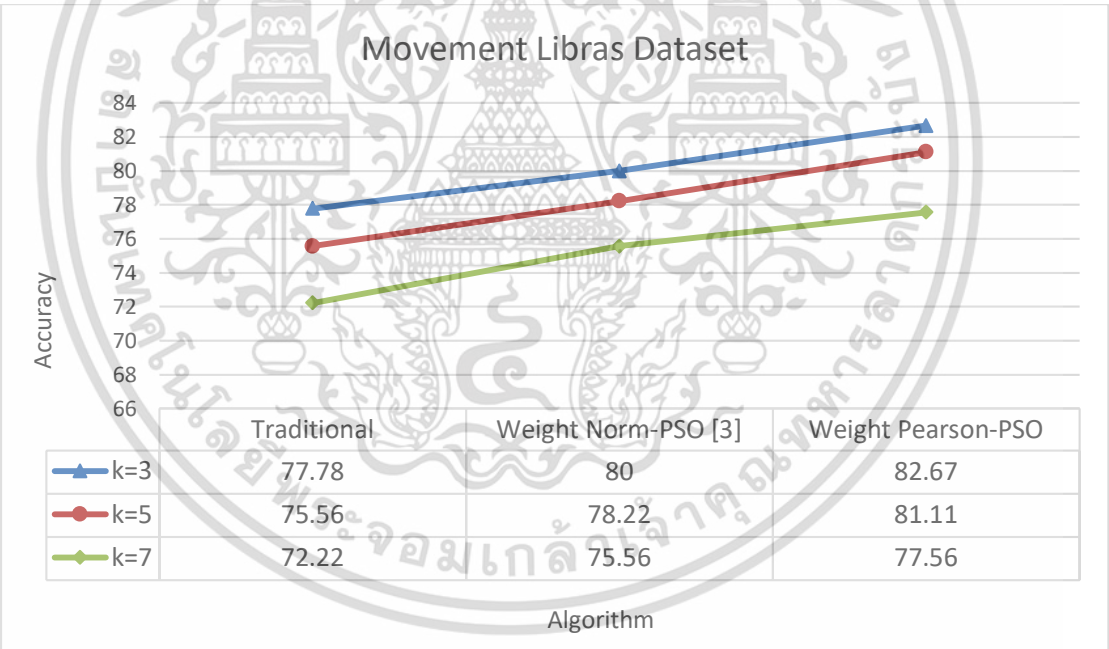
ผลของการทดสอบจากตารางที่ 5.5 และ 5.6 จะเห็นได้ว่าอัลกอริทึมที่นำเสนอสามารถปรับปรุงความแม่นยำของการจำแนกประเภทในอัลกอริทึมแบบดั้งเดิมได้ในทุกชุดข้อมูล โดยเฉพาะอย่างยิ่งในชุดข้อมูลบางชุดจะเห็นว่าอัลกอริทึมที่นำเสนอสามารถปรับปรุงประสิทธิภาพได้อย่างดี เช่น ชุดข้อมูล Hepatitis ชุดข้อมูล Movement Libras เป็นต้น ในการทดลองส่วนที่สามผู้วิจัยได้นำเสนอผลทดสอบความแม่นยำของการจำแนกประเภทในรูปแบบกราฟความสัมพันธ์ โดยเปรียบเทียบจาก 3 อัลกอริทึมคือ อัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิม วิธีการปรับมาตรฐานเพื่อหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคสำหรับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว (Weight Norm-PSO KNN) ซึ่งถูกนำเสนอโดย Guo และคณะ [15] และอัลกอริทึมที่ได้นำเสนอโดยกำหนดวิธีการ

คำนวณโดยใช้ระยะทางยุคลิดแบบให้น้ำหนักกับคุณลักษณะ กำหนดค่าพารามิเตอร์ของอัลกอริทึมการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคในอัลกอริทึมที่นำเสนอตามตารางที่ 5.1 ส่วนอัลกอริทึม Weight Norm-PSO กำหนดจำนวนอนุภาคเป็น 20 อนุภาค ω = 0.3 ค่า c_1 และ c_2 กำหนดเป็น 1.49445 รอบการค้นหา 100 รอบ อัลกอริทึมจะจบการทำงานหลังจากเสร็จสิ้นการวนซ้ำของรอบค้นหา ผลความแม่นยำในการจำแนกประเภทของ 3 อัลกอริทึมสามารถแสดงในรูปแบบกราฟความสัมพันธ์ดังรูปที่ 5.1 ถึงรูปที่ 5.5

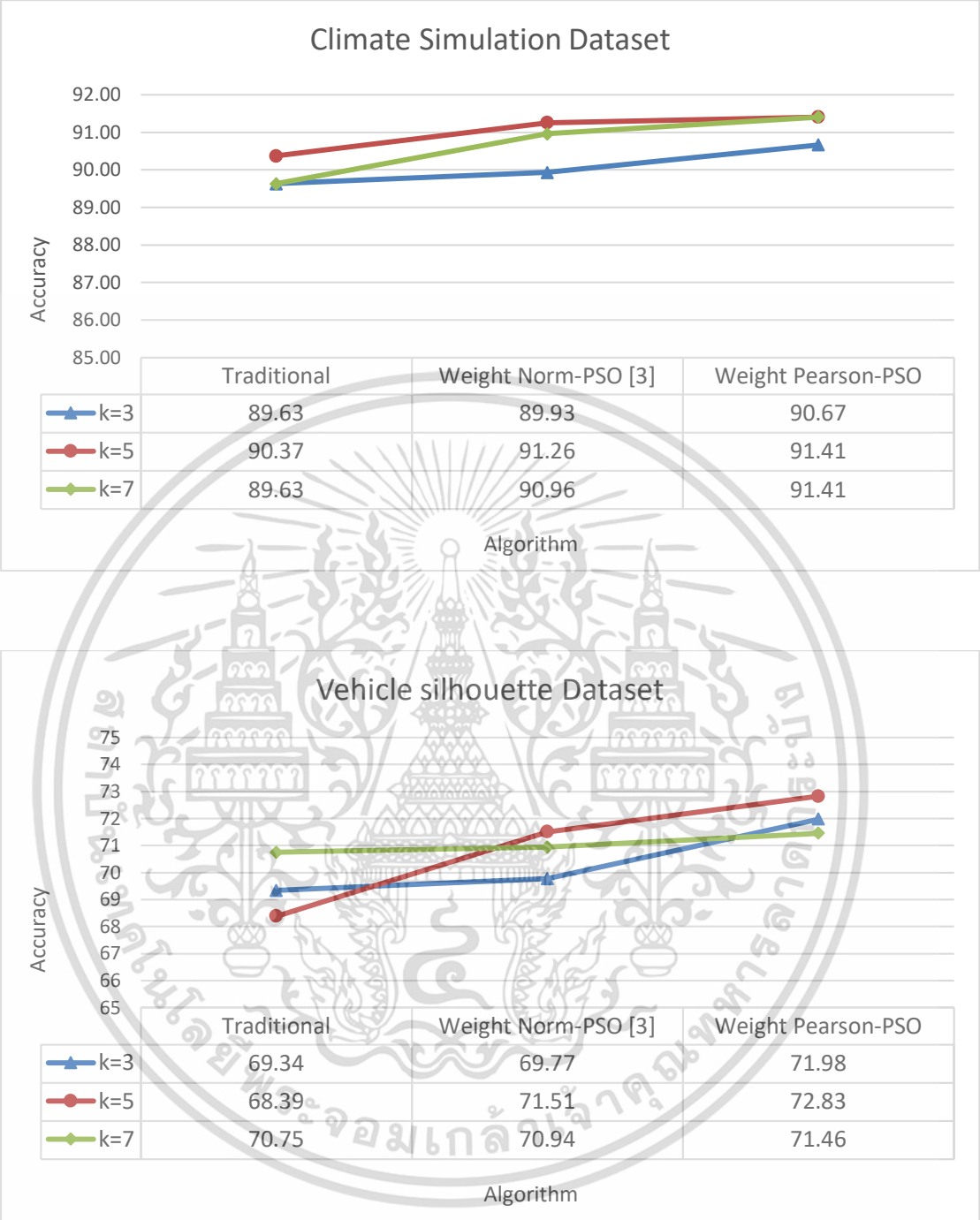


รูปที่ 5.1 กราฟแสดงความสัมพันธ์ของผลทดสอบความแม่นยำในการจำแนกประเภทของทั้ง 3 วิธีด้วยชุดข้อมูล Hepatitis และ Heart Statlog

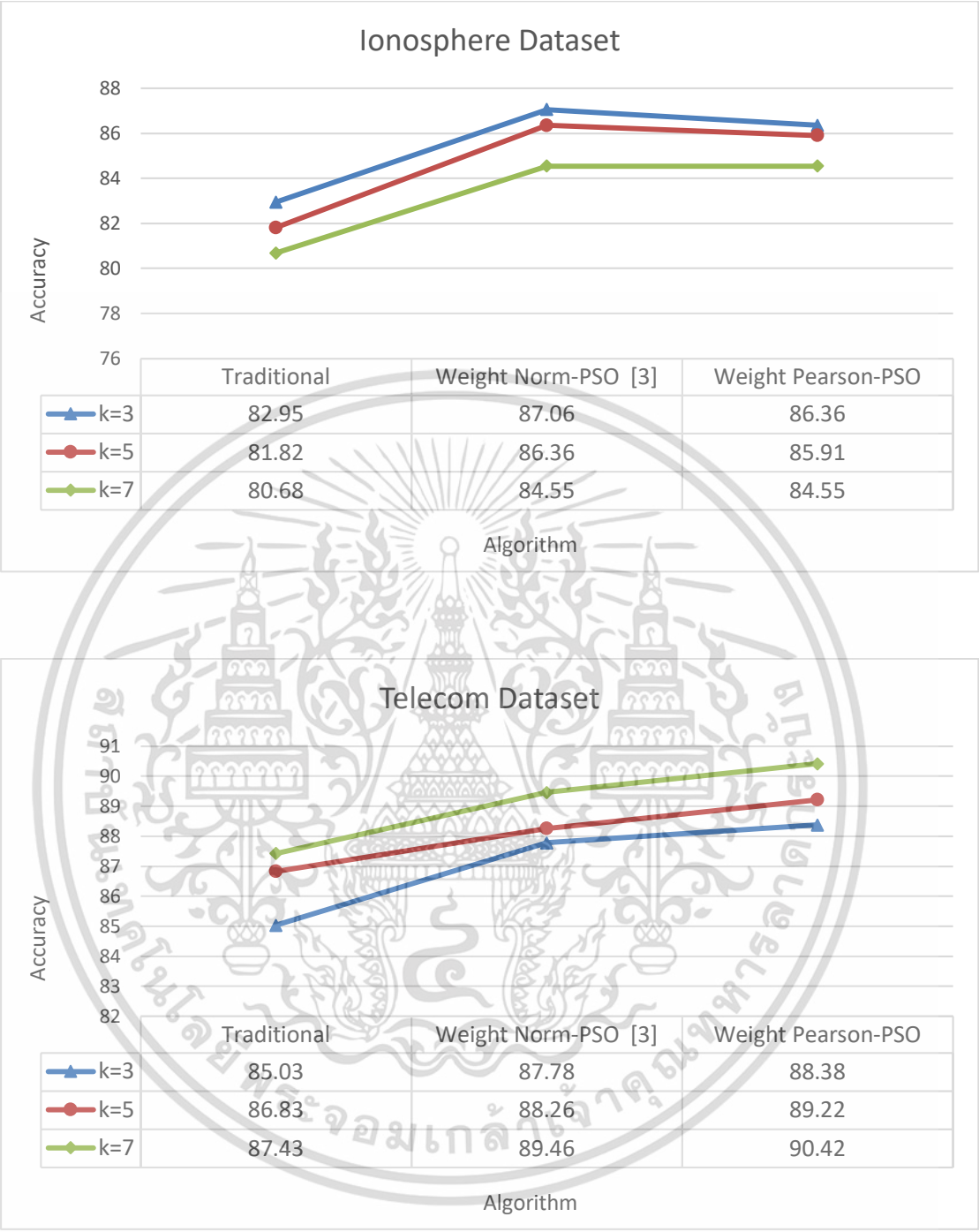




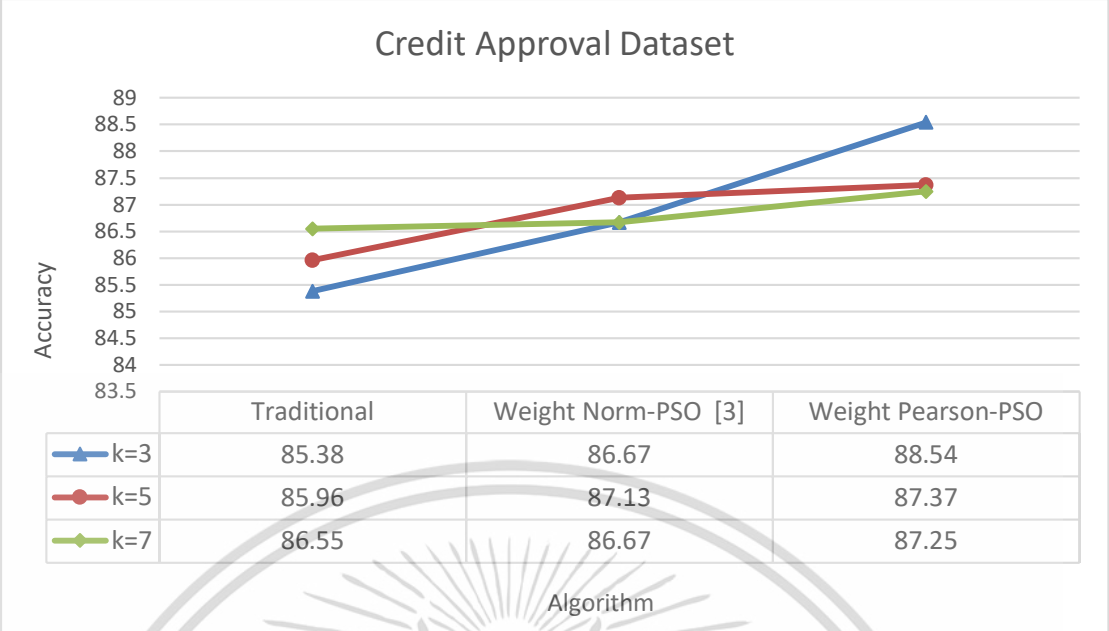
รูปที่ 5.2 กราฟแสดงความสัมพันธ์ของผลทดสอบความแม่นยำในการจำแนกประเภทของทั้ง 3 วิธี ด้วยชุดข้อมูล Sonar และ Movement Libras

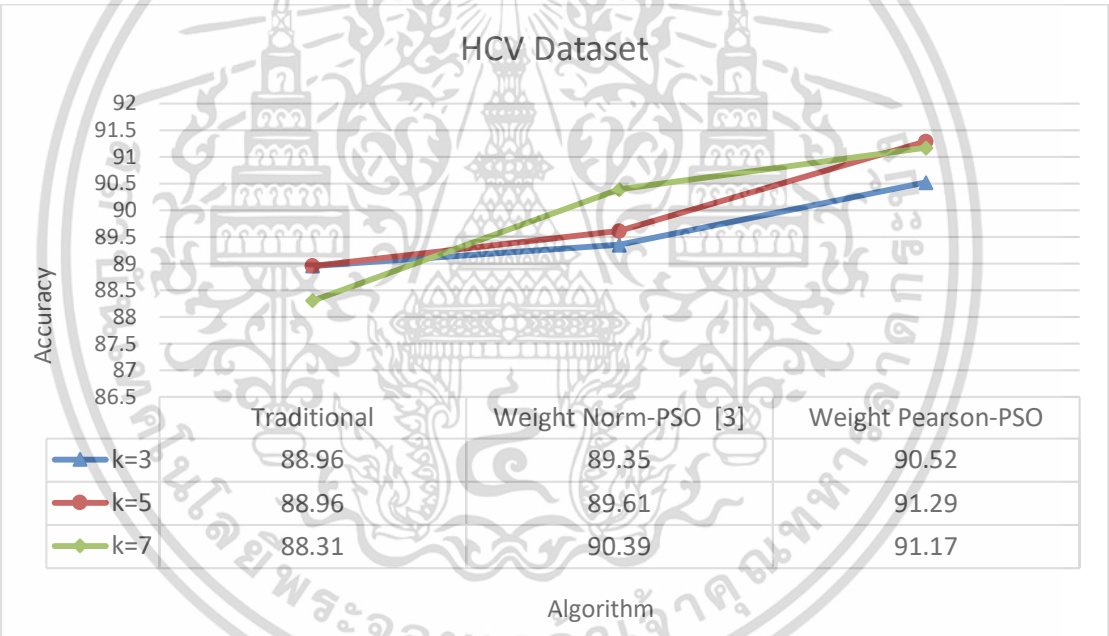


รูปที่ 5.3 กราฟแสดงความสัมพันธ์ของผลทดสอบความแม่นยำในการจำแนกประเภทของทั้ง 3 วิธี ด้วยชุดข้อมูล Climate Simulation และ Vehicle silhouette



รูปที่ 5.4 กราฟแสดงความสัมพันธ์ของผลทดสอบความแม่นยำในการจำแนกประเภทของทั้ง 3 วิธี ด้วยชุดข้อมูล Ionosphere และ Telecom

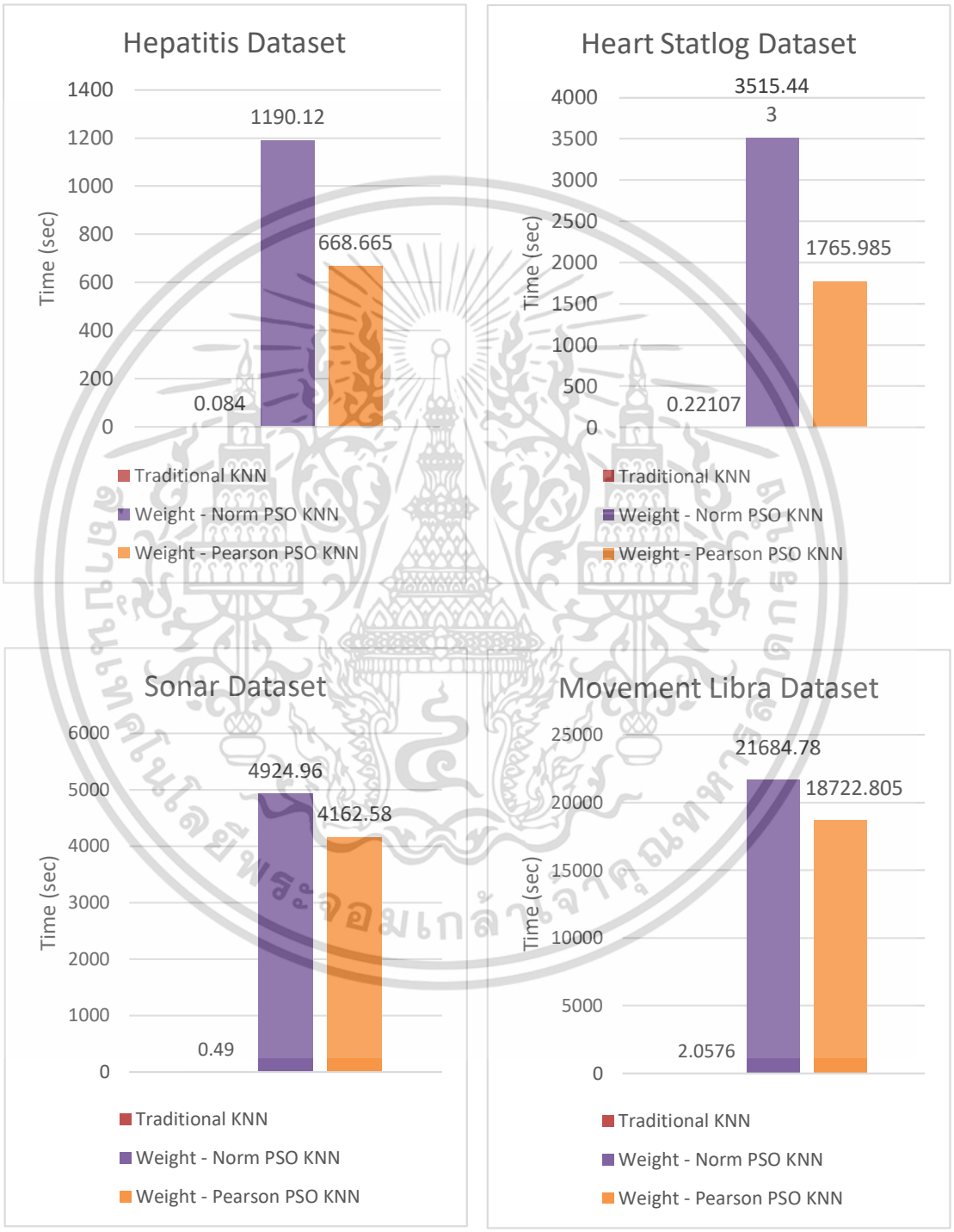




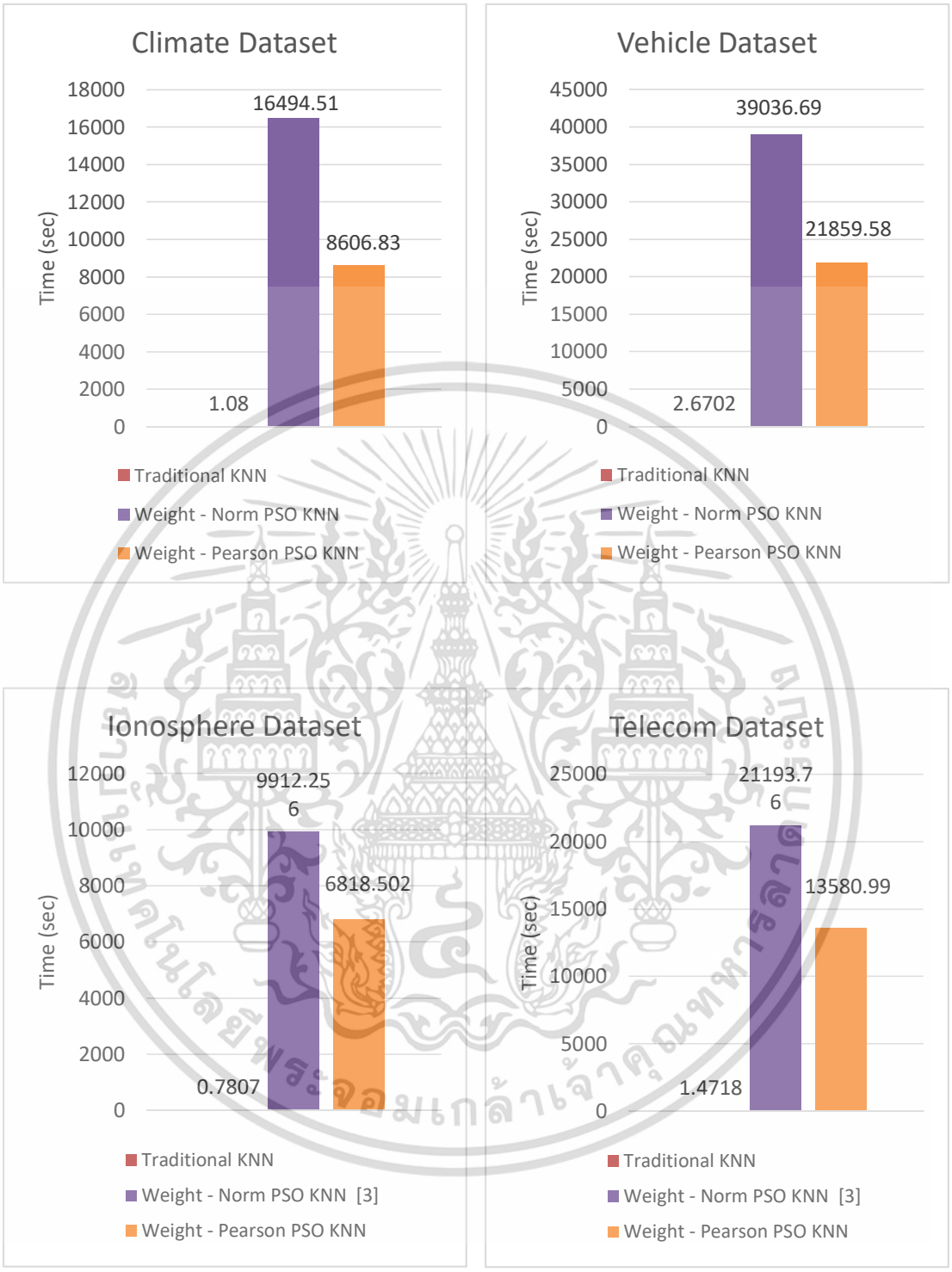
รูปที่ 5.5 กราฟแสดงความสัมพันธ์ของผลทดสอบความแม่นยำในการจำแนกประเภทของทั้ง 3 วิธี ด้วยชุดข้อมูล Credit Approval และ HCV

จากผลทดลองจากตารางที่ 5.3 และรูปที่ 5.1 ถึงรูปที่ 5.5 จะเห็นได้ว่าอัลกอริทึมทั้ง 2 ที่มีการนำค่าน้ำหนักที่เหมาะสมมาใช้งาน สามารถเพิ่มความแม่นยำในการจำแนกประเภทได้อย่างมีประสิทธิภาพเมื่อเปรียบเทียบกับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมจากชุดข้อมูลที่นำมาทดสอบ แสดงให้เห็นถึงประโยชน์ของการนำค่าน้ำหนักมาใช้งานในการคำนวณระยะทางเพื่อจำแนกประเภท ยิ่งไปกว่านั้นเมื่อเปรียบเทียบระหว่าง 2 อัลกอริทึมที่มีการนำค่าน้ำหนักที่เหมาะสมมาใช้

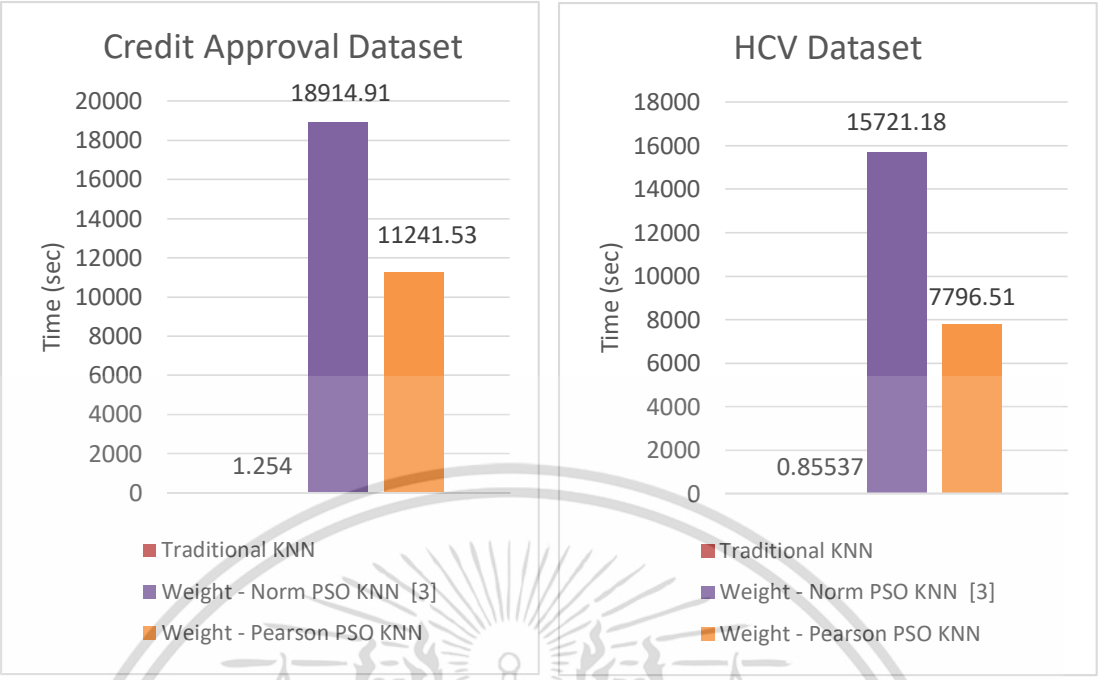
อัลกอริทึมที่นำเสนอยังมีความแม่นยำของการจำแนกประเภทที่ดีกว่าวิธีการปรับมาตรฐานเพื่อหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว [15] ในชุดข้อมูลส่วนใหญ่อีกด้วย แต่จุดที่น่าสนใจคือความสามารถของอัลกอริทึมที่ได้นำเสนอซึ่งสามารถใช้เวลาในการทำงานเพื่อจำแนกประเภทข้อมูลทดสอบได้ดีกว่ากับทุกชุดข้อมูลดังแสดงในรูปที่ 5.6 ถึงรูปที่ 5.8



รูปที่ 5.6 แสดงผลของเวลาที่ใช้ในการจำแนกประเภทข้อมูลของทั้ง 3 อัลกอริทึมด้วยชุดข้อมูล Hepatitis, Heart Statlog, Sonar และ Movement Libras



รูปที่ 5.7 แสดงผลของเวลาที่ใช้ใช้ในการจำแนกประเภทข้อมูลของทั้ง 3 อัลกอริทึมด้วยชุดข้อมูล Climate, Vehicle silhouette, Ionosphere และ Telecom



รูปที่ 5.8 แสดงผลของเวลาที่ใช้ในการจำแนกประเภทข้อมูลของทั้ง 3 อัลกอริทึมด้วยชุดข้อมูล Credit Approval และ HCV

ผลการทดลองในรูปที่ 5.6 ถึงรูปที่ 5.8 จะเป็นเวลาที่ดีที่สุดจากการทำงานของแต่ละอัลกอริทึมที่นำมาเปรียบเทียบ ซึ่งจะเห็นว่าสำหรับอัลกอริทึมที่มีการปรับละเอียดค่าน้ำหนัก อัลกอริทึมที่นำเสนอสามารถใช้เวลาในการทำงานน้อยกว่าวิธีการปรับมาตรฐานเพื่อหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวในทุกชุดข้อมูลทดสอบ โดยเฉพาะอย่างยิ่งในชุดข้อมูล HCV และ Heart Statlog อัลกอริทึมที่นำเสนอสามารถใช้เวลาในการจำแนกประเภทได้ดีกว่าถึง 50.41% และ 49.76% ตามลำดับ ผลของเวลาที่ใช้ในการจำแนกประเภทนี้แสดงให้เห็นถึงประโยชน์ของการนำเอาค่าน้ำหนักแบบหยาบมาใช้งาน เพราะแม้จะใช้เวลาน้อยกว่าในการทำงานอัลกอริทึมที่นำเสนอกลับสามารถค้นหาค่าน้ำหนักที่เหมาะสมได้ อีกทั้งส่งผลให้การจำแนกประเภทของอัลกอริทึมที่นำเสนอมีความแม่นยำมากกว่าวิธีการปรับมาตรฐานเพื่อหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวในชุดข้อมูลส่วนใหญ่ เนื่องด้วยค่าน้ำหนักแบบหยาบนั้นสามารถอธิบายความสำคัญของแต่ละคุณลักษณะเบื้องต้นส่งผลให้การค้นหาค่าน้ำหนักที่เหมาะสมสามารถใช้เวลาได้อย่างเหมาะสม และแม้ว่าในการทำงานของอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวแบบดั้งเดิมจะใช้เวลาน้อยที่สุดเนื่องจากไม่มีการปรับละเอียดค่าน้ำหนัก แต่จะเห็นได้ว่าเวลาที่ใช้มากขึ้นของอัลกอริทึมที่นำเสนอก็คุ้มค่าเมื่อเปรียบเทียบกับประสิทธิภาพความแม่นยำที่เพิ่มขึ้นในการจำแนกประเภท นอกจากนี้ผู้วิจัยยังได้ศึกษาลักษณะของชุดข้อมูลซึ่งส่งผลต่อประสิทธิภาพในการจำแนกเมื่อใช้อัลกอริทึมที่นำเสนอ หรือเมื่อใช้กับอัลกอริทึมที่มีการกำหนดค่าน้ำหนักของคุณลักษณะเข้ามาเกี่ยวข้อง ซึ่งสามารถแสดงการวิเคราะห์ลักษณะของข้อมูลที่มีผลต่อประสิทธิภาพได้ดังตารางที่ 5.7

ตารางที่ 5.7 แสดงการวิเคราะห์ลักษณะของชุดข้อมูลที่เหมาะกับอัลกอริทึมที่นำเสนอ

Data set	Size	No. of Attributes	No. of Classes	No. of Attributes which have same weight values (Mean)	Percent of Acc Improvement (K = 5)
Hepatitis	155	19	2	8	9.29%
Heart Statlog	270	13	2	4.4	4.0%
Sonar	208	60	2	32	0.44%
Climate Simulation	540	18	2	11.4	1.15%
Movement Libras	360	90	15	43	7.35%
Vehicle silhouette	946	18	4	6.8	6.48%
Ionosphere	351	34	2	15.4	5.00%
Telecom	668	19	2	8.2	2.76%
Credit Approval	684	14	2	7.2	1.63%
HCV dataset	615	12	5	6	2.63%

จากตารางที่ 5.7 จะแสดงการวิเคราะห์ลักษณะของชุดข้อมูลที่เหมาะกับอัลกอริทึมที่นำเสนอ โดยทำการเปรียบเทียบเมื่อกำหนดค่า K เป็น 5 ข้อสังเกตที่น่าสนใจคือค่าเฉลี่ยของจำนวนคุณลักษณะที่มีค่าน้ำหนักเท่ากันในคอลัมน์ที่ 5 ซึ่งหากจำนวนค่าน้ำหนักของคุณลักษณะที่เท่ากันมีน้อยกว่าหรือเท่ากับครึ่งหนึ่งของจำนวนคุณลักษณะทั้งหมดของชุดข้อมูลนั้น จะส่งผลให้ประสิทธิภาพการจำแนกประเภทข้อมูลทดสอบเพิ่มขึ้นอย่างมีนัยสำคัญ เช่น ชุดข้อมูล Hepatitis และ ชุดข้อมูล Vehicle silhouette เป็นต้น ซึ่งทั้ง 2 ชุดข้อมูลมีค่าเฉลี่ยของจำนวนคุณลักษณะที่มีค่าน้ำหนักเท่ากันเป็น 8 และ 6.8 เมื่อนำมาจำแนกประเภทด้วยอัลกอริทึมที่นำเสนอสามารถปรับปรุงประสิทธิภาพของความแม่นยำของอัลกอริทึมแบบดั้งเดิมได้ถึง 9.29% และ 6.48% ในทางกลับกันหากจำนวนค่าน้ำหนักของ

คุณลักษณะที่เท่ากันมีมากกว่าครึ่งหนึ่งของจำนวนคุณลักษณะทั้งหมด จะส่งผลให้ประสิทธิภาพการจำแนกประเภทข้อมูลทดสอบไม่ได้ผลดีเท่าที่ควรเมื่อเปรียบเทียบกับข้อมูลในลักษณะก่อนหน้าดังชุดข้อมูล Climate Simulation และ ชุดข้อมูล Credit Approval ซึ่งมีค่าเฉลี่ยของจำนวนคุณลักษณะที่มีค่าน้ำหนักเท่ากันเป็น 11.4 และ 7.2 ตามลำดับ อัลกอริทึมที่นำเสนอสามารถปรับปรุงประสิทธิภาพของความแม่นยำได้ 1.15% และ 1.63% จากตัวอย่างของชุดข้อมูลที่กล่าวถึงนี้สามารถอธิบายได้ว่าลักษณะของชุดข้อมูลแต่ละชุดมีประสิทธิภาพแตกต่างกันเมื่อนำมาใช้กับอัลกอริทึมที่นำเสนอหรืออัลกอริทึมที่มีการกำหนดค่าน้ำหนักของคุณลักษณะ โดยชุดข้อมูลซึ่งมีจำนวนค่าน้ำหนักของคุณลักษณะที่เท่ากันมากกว่าครึ่งหนึ่งไม่เหมาะกับการนำมาใช้ในการระบุค่าน้ำหนักให้กับคุณลักษณะ เนื่องจากวิธีการให้น้ำหนักไม่สามารถอธิบายความแตกต่างของแต่ละคุณลักษณะในชุดข้อมูลเหล่านั้นได้อย่างเหมาะสม ส่งผลให้การจำแนกประเภทของอัลกอริทึมที่มีการให้น้ำหนักเห็นว่าแต่ละคุณลักษณะมีความสำคัญเท่ากัน และอาจนำไปสู่การจำแนกประเภทที่ไม่มีประสิทธิภาพ นอกจากนี้การกำหนดค่า K เพื่อพิจารณาเพื่อนบ้านที่ใกล้ที่สุดจำนวน K ตัวยังส่งผลต่อประสิทธิภาพในการอธิบายความสำคัญของแต่ละคุณลักษณะและประสิทธิภาพในการจำแนกประเภทอีกด้วย ตัวอย่างที่น่าสนใจคือชุดข้อมูล Sonar โดยเมื่อกำหนดค่า K เป็น 5 จะมีค่าเฉลี่ยจำนวนค่าน้ำหนักของคุณลักษณะเท่ากัน 32 ค่าและสามารถปรับปรุงประสิทธิภาพของความแม่นยำได้เพียง 0.44% แต่หากกำหนดค่า K เป็น 3 จำนวนค่าน้ำหนักของคุณลักษณะที่เท่ากันจะมีค่าเฉลี่ยอยู่ที่ 27.2 และอัลกอริทึมที่นำเสนอสามารถปรับปรุงประสิทธิภาพของอัลกอริทึมแบบดั้งเดิมได้ถึง 5.91%

บทที่ 6
สรุปผลการทดลองและข้อเสนอแนะ

6.1 สรุปผลการทดลอง

วิทยานิพนธ์ฉบับนี้ได้ทำการพัฒนาและปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว แบบดั้งเดิมซึ่งยังมีข้อบกพร่อง โดยผู้วิจัยได้เลือกแก้ไขปัญหาในการให้ความสำคัญของคุณลักษณะ เนื่องจากอัลกอริทึมแบบดั้งเดิมให้ความสำคัญกับแต่ละคุณลักษณะเท่ากันทั้งหมด ซึ่งในความเป็นจริง คุณลักษณะของข้อมูลแต่ละตัวมีความสำคัญแตกต่างกัน ผู้วิจัยจึงทำการปรับปรุงอัลกอริทึมด้วยการให้น้ำหนักกับคุณลักษณะของข้อมูลจากค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันจากนั้นหาค่าน้ำหนักที่เหมาะสมที่สุดด้วยวิธีการแบบกลุ่มอนุภาค (Particle Swarm Optimization หรือ PSO) ในงานวิจัยจะนำผลการทดลองซึ่งเป็นการเปรียบเทียบประสิทธิภาพการทำงานของการจำแนกประเภทข้อมูลทดสอบทั้ง 3 อัลกอริทึมคือ วิธีการจำแนกประเภทด้วยอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว แบบดั้งเดิม วิธีการปรับมาตรฐานเพื่อหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคและการประยุกต์ใช้ในปัญหาการให้น้ำหนักกับคุณลักษณะสำหรับอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัว (The Normalized-PSO and Its Application in Attribute Weighted Optimal Problem) [15] และวิธีการปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยการให้น้ำหนักกับคุณลักษณะของข้อมูลจากค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันและการปรับละเอียด (Improving KNN Algorithm based on Weighted Attributes by Pearson Correlation Coefficient and PSO Fine Tuning) หรืออัลกอริทึมที่นำเสนอ ซึ่งจากผลการทดลองจะเห็นได้ว่าการปรับปรุงอัลกอริทึมด้วยวิธีการที่นำเสนอในงานชิ้นนี้สามารถการแก้ปัญหาข้อจำกัดของอัลกอริทึมแบบดั้งเดิมในด้านความแม่นยำของผลการทดลอง โดยหากพิจารณาผลการทดลองในบทที่ 5 หัวข้อที่ 5.4 จะพบว่าความสามารถของอัลกอริทึมที่นำเสนอสามารถจำแนกข้อมูลทดสอบได้ประสิทธิภาพมากที่สุดเมื่อเทียบกับอัลกอริทึมอื่น อีกทั้งยังสามารถใช้เวลาในการหาค่าน้ำหนักได้อย่างเหมาะสมเมื่อเปรียบเทียบกับอัลกอริทึมที่มีวิธีการปรับละเอียดเช่นเดียวกัน

6.2 ข้อเสนอแนะและแนวทางในการปรับปรุง

วิธีการปรับปรุงอัลกอริทึมเพื่อนบ้านใกล้ที่สุดจำนวน K ตัวด้วยการให้น้ำหนักกับคุณลักษณะของข้อมูลจากค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันและการปรับละเอียดที่ได้นำเสนอเป็นวิธีการที่พัฒนามาจากการทำงาน 2 ขั้นตอน โดยขั้นตอนที่ 1 เป็นการให้ค่าน้ำหนักกับคุณลักษณะแบบหยาบเพื่อคำนวณระยะทางในการค้นหาเพื่อนบ้านที่ใกล้ที่สุดจำนวน K ตัว และในขั้นตอนที่ 2 เป็นการปรับละเอียดค่าน้ำหนัก ซึ่งการพัฒนาอัลกอริทึมนี้ได้ถูกพัฒนาขึ้นเพื่อให้สามารถค้นหาค่าน้ำหนักที่ดีที่สุด

ได้ดียิ่งขึ้นเมื่อเปรียบเทียบกับอัลกอริทึมแบบดั้งเดิมและสามารถใช้เวลาการทำงานได้เร็วยิ่งขึ้นเมื่อเปรียบเทียบกับอัลกอริทึมที่มีการปรับละเอียด ในการพัฒนาอัลกอริทึมที่นำเสนอนี้จำเป็นต้องทำการทดลองปรับเปลี่ยนเพื่อให้สามารถค้นหาค่าน้ำหนักที่เหมาะสมได้ดียิ่งขึ้น ซึ่งผู้วิจัยได้ระบุแนวทางการศึกษาและวิจัยเพิ่มเติมเพื่อเป็นแนวทางในการปรับปรุงไว้ดังนี้

1) วิธีการให้น้ำหนักคุณลักษณะในขั้นตอนการค้นหาแบบหยาบ เนื่องจากน้ำหนักเริ่มต้นนั้นจะส่งผลต่อการปรับค่าน้ำหนักเพื่อทำให้ค่ามีความเหมาะสม และมีผลต่อเวลาในการค้นหาของขั้นตอนการปรับละเอียด ซึ่งจะส่งผลต่อประสิทธิภาพของอัลกอริทึม ดังนั้นหากในอนาคตมีวิธีการที่สามารถอธิบายความสัมพันธ์ของข้อมูลระหว่างแต่ละคุณลักษณะและคลาสคำตอบได้ดียิ่งขึ้น อาจจะมีผลทำให้ประสิทธิภาพของอัลกอริทึมมีประสิทธิภาพมากยิ่งขึ้นด้วย
2) ตัวแปรในขั้นตอนการปรับละเอียด ในส่วนนี้การกำหนดค่าตัวแปรขอบเขตของการให้ค่าน้ำหนักอย่างเหมาะสมยังคงเป็นสิ่งที่ต้องศึกษาเพื่อเพิ่มประสิทธิภาพของการค้นหาค่าน้ำหนัก โดยจะต้องศึกษาว่าข้อมูลที่นำมาทำการทดลองหรือเรียนรู้เหมาะสมกับขอบเขตการให้น้ำหนักหรือไม่ โดยตัวแปรขอบเขตที่ถูกนำมาใช้ปรับละเอียดค่าน้ำหนักเหล่านั้นมีเพื่อให้การค้นหาค่าน้ำหนักอยู่ในช่วงที่กำหนด แต่ข้อมูลบางประเภทแม้สามารถปรับละเอียดค่าน้ำหนักได้แต่ค่าน้ำหนักบางค่ากลับถูกจำกัดในค่าขอบเขตนั้น ส่งผลให้การกำหนดค่าน้ำหนักทำมีประสิทธิภาพไม่ดีเท่าที่ควร ในทางกลับกันหากกำหนดค่าของช่วงที่มากเกินไปก็จะส่งผลให้ช่วงของการค้นหากว้างยิ่งขึ้นเช่นกัน
3) การศึกษาลักษณะข้อมูลที่เหมาะสมกับการทำงานของอัลกอริทึม ในงานวิจัยชิ้นนี้ได้นำเอาชุดข้อมูลสาธารณะ UCI ซึ่งมีลักษณะของข้อมูลแตกต่างกัน หากชุดข้อมูลมีความเหมาะสมกับการหาค่าน้ำหนักจะส่งผลให้การจำแนกประเภทมีความแม่นยำเพิ่มขึ้นอย่างมีนัยสำคัญ จากการวิเคราะห์ลักษณะของชุดข้อมูลในตารางที่ 5.5 และ ตารางที่ 5.6 จะเห็นได้ว่าลักษณะของชุดข้อมูลมีผลต่อประสิทธิภาพในการอธิบายความสำคัญของแต่ละคุณลักษณะ ซึ่งชุดข้อมูลที่มีความเหมาะสมกับอัลกอริทึมที่นำเสนอหรืออัลกอริทึมที่มีการให้น้ำหนักกับคุณลักษณะจะสามารถอธิบายความสำคัญของแต่ละคุณลักษณะด้วยค่าน้ำหนักที่แตกต่างกัน ในทางกลับกันชุดข้อมูลซึ่งไม่เหมาะสมอัลกอริทึมที่นำเสนอหรืออัลกอริทึมที่มีการให้น้ำหนักกับคุณลักษณะจะไม่สามารถอธิบายความแตกต่างของแต่ละคุณลักษณะในชุดข้อมูลเหล่านั้นได้อย่างเหมาะสม

เอกสารอ้างอิง

[1] Pooja Rani, Aug. 2017. “A Review of various KNN Techniques.” International Journal for Research in Applied Science & Engineering Technology (IJRASET). 5(8) : 1174-1179.

[2] Jingwen Sun, Weixing Du and Niancai Shi. 2018. “A Survey of kNN Algorithm.” Information Engineering and Applied Computing (2018).

[3] DR.O. Obulesu, M. Mahendra and M. ThrilokReddy. Jul. 2018. “Machine Learning Techniques and Tools: A survey.” 2018 International Conference on Inventive Research in Computing Applications (ICIRCA). pp.605-611.

[4] Xian Liang, Fuheng Qu, Yong Yang and Hua Cai. 2015. “An Improved ID3 Decision Tree Algorithm Based on Attribute Weighted.” International Conference on Civil, Materials and Environmental Sciences (CMES 2015). pp.613-615.

[5] Sona Taheri, John Yearwood, Musa Mammadov and Sattar Seifollahi. Apr. 2014. “Attribute Weighted Naive Bayes Classifier Using a Local Optimization.” Neural Computing and Applications. pp.995–1002.

[6] Zhang Li, Zhang Chengjin, Xu Qingyang and Liu Chunfa. Aug. 2015. “Weigted-KNN and its application on UCI.” 2015 IEEE International Conference on Information and Automation. pp.1748-1750.

[7] Jie Huang, Yongqing Wei, Jing Yi and Mengdi Liu. Feb. 2018. “An Improved kNN Based on Class Contribution and Feature Weighting.” 2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA). pp.313-316.

[8] Diego P. Vivencio Estevam R. Hruschka M. do Carmo Nicoletti Edimilson B. dos Santos and Sebastian D.C.O. Galvao. Apr. 2007. “Feature-weighted k-Nearest Neighbor Classifier.” Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence (FOCI 2007). pp.481-486

[9] Xingjiang Xiao and Huafeng Ding. Oct. 2012. “Enhancement of K-nearest Neighbor Algorithm Based on Weighted Entropy of Attribute Value.” 2012 5th International Conference on BioMedical Engineering and Informatics (BMEI 2012). pp.1261-1264.

[10] Jianping Gou, Lan Du b, Yuhong Zhang and Taisong Xiong. Jun. 2012. “A New Distance-weighted k -nearest Neighbor Classifier.” Journal of Information & Computational Science 9. pp. 1429-1436.

[11] Sahibsingh A. Dudani. Apr. 1976. “The Distance-Weighted k-Nearest Neighbor Rule.” IEEE Transactions on Systems, Man, and Cybernetics. 6(4) : 325–327.

[12] Jia Wu, Zhihua Cai and Zhechao Gao. Aug. 2010. “Dynamic K-Nearest-Neighbor with Distance and Attribute Weighted for Classification.” International Conference on Electronics and Information Engineering (ICEIE 2010). pp.356-360.

[13] Shweta Taneja, Charu Gupta, Kratika Goyal and Dharna Gureja. Feb. 2014. “An Enhanced K-Nearest Neighbor Algorithm Using Information Gain and Clustering.” 2014 Fourth International Conference on Advanced Computing & Communication Technologies. pp.325-329.

[14] J. Kennedy and R. Eberhart. Dec. 1995. “Particle Swarm Optimization.” In

Proceedings of IEEE International Conference on Neural Networks IV. pp.1942-1948.

[15] Junjie Guo, Jin Gou, Cheng Wang and Wei Luo. Sep. 2016. “The Normalized-PSO and Its Application in Attribute Weighted Optimal Problem.” 2016 Third International Conference on Trustworthy Systems and their Applications (TSA). pp.48-53.

[16] Chi Yu-hong, Sun Fu-chun, Wang and Wei-ju. 2011. “An Improved Particle Swarm Optimization Algorithm with Search Space Zoomed Factor and Attractor [J].” Chinese Journal of Computers. 34(1) : 115-130.

[17] Qinghua Cao and Yu Liu. Aug. 2010. “A KNN Classifier with PSO Feature Weight Learning Ensemble [C].” 2010 International Conference on Intelligent Control and Information Processing. pp.110-114.

[18] Michel Marie Deza and Elena Deza. 2009. Encyclopedia of Distances. p.94. 1st ed. Springer-Verlag Berlin Heidelberg.

[19] Chen Wei, Xiang Tie-ming and Xu jie. Jun. 2015 “Team Evolutionary Algorithm Based on PSO[J].” Pattern Recognition And Artificial Intelligence. 28(6) : 521-527.

[20] Gene V. Glass and Kenneth D. Hopkins. Jun. 2008. “Statistical Methods in Education and Psychology” 3rd ed. Pearson.

[21] Muhammad Ejazuddin Syed. Nov. 2014. “Attribute weighting in K-nearest neighbor classification.” M.Sc. thesis University of Tampere Computer Science School of Information Sciences.

[22] Shenglan Liu Ping Zhu and Sujuan Qin. Dec. 2018 “An Improved Weighted KNN Algorithm for Imbalanced Data Classification.” 2018 IEEE 4th International Conference on Computer and Communications. pp.1814-1819.

[23] N. Garcia-Pedrajas, JA. Del-Castillo and G. Cerruela-Garcia. Feb. 2017. "Aproposal for local k values for k-Nearest Neighbor rule." IEEETransactions on Neural Networks & Learning Systems. 28(10) : 470-475.

[24] Tom M. Mitchell. Mar. 1997. Machine Learning. pp.231-248. McGraw-Hill Education.

ภาคผนวก

ภาคผนวก ก ผลงานวิจัยที่ได้รับการตีพิมพ์

ภาคผนวก ก

ผลงานวิจัยที่ได้รับการตีพิมพ์

Wanarase Sinhashthita, and Kietikul Jearanaitanakij, “Improving KNN Algorithm Based on Weighted Attributes by Pearson Correlation Coefficient and PSO Fine Tuning”, The 5th International Conference on Information Technology: InCIT2020 The Tide Resort, Bangsaen, Chonburi, Thailand, pp.CIT49-CIT52.

Improving KNN Algorithm Based on Weighted Attributes by Pearson Correlation Coefficient and PSO Fine Tuning

Wanarase Sinhashthita
Department of Computer Engineering
Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
59601094@kmitl.ac.th

Kietikul Jearanaitanakij
Department of Computer Engineering
Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
kietikul.je@kmitl.ac.th

Abstract— **Assigning proper weights to attributes in some datasets according to their importances can significantly improve the classification accuracy. Weighted attributes can support the classification methods effectively if their weights truly represent by their importances. In this research, we improve the K - Nearest Neighbors (KNN) algorithm by using Pearson correlation coefficient along with Particle Swarm Optimization (PSO) to find the optimal set of weights for attributes in the dataset. The experimental results show that the proposed method can significantly improve the classification accuracy when compared to the traditional KNN algorithm.**

Keywords- K Nearest Neighbors algorithm; Particle Swarm Optimization; Classification; Attribute Weighting; Pearson correlation coefficient

I. INTRODUCTION

K-Nearest Neighbors (KNN) classification is a method that assigns an answer class to an unseen instance by calculating Euclidean distances from the number of nearest neighbors and assigning the category to an unseen instance by the majority answer [1-3].

Assigning the proper weights to attributes can significantly improve the classification accuracy. Attribute weighting is a delicate process and should be done appropriately because attributes are not equally important. Liang et al. [4] apply attribute weighting to decision tree classification by using gain ratio method for assigning the weights of attributes. Similarly, Taheri et al. [5] present the combination of attribute weighting method, which assigns weight by a local optimization using the Quasisecant method, and Naive Bayes classification.

In addition, Li et al. [6] present weighted-KNN algorithm which uses the error rate to measure the importance of attribute. If the important attributes are removed from the learning data, the error rate must be largely increased. On the other hand, if irrelevant attributes are removed, error rate must be decreased. Huang et al. [7] propose the attribute removing method to find the important attributes by measuring accuracy after each attribute is removed.

Finding an appropriate set of weights for attributes is challenging research issue. Syed [8] uses several existing methods to find weights of attributes such as gain ratio, Pearson correlation coefficient, etc. Furthermore, Gou et al. [9] present weighted KNN classification which calculates the weights of the nearest neighbor according to their distances by dual weighting based on WKNN algorithm [10].

In order to further improve the accuracy, some works try to find the optimal attribute's weights by using PSO algorithm [11] for fine tuning. They assign weights as initial positions of particles and use classification accuracy as a value of a fitness function. Guo et al. [12] present algorithm to find optimal weights which uses random weights as initial positions and then fine tune the weights with the Normalized-PSO [13]. In addition, they also mentioned about Cao and Liu's work [14], which also uses PSO algorithm, but they use cloud model as initial weights instead. Among various weight fine tuning algorithms mentioned above, none of them can find the optimal weights within a reasonable running time. Besides finding the optimal weight for each attribute, another goal of our research is to diminish the running time issue.

In this research, we use Pearson correlation coefficient to assign initial weights to attributes by finding the relationship between each attribute and its target class, as we will call it as a rough search. The reason of using Pearson correlation coefficient will be explained in the discussion section. Afterward we fine tune those weights with PSO to adjust weights of attributes for the best classification accuracy. The experimental results show that the proposed method can improve the classification accuracy of the traditional K Nearest Neighbors algorithm within a reasonable running time.

In the following contents, the topics are organized as follows. In section 2, we discuss about related backgrounds. In section 3, we explain how the proposed algorithm works. In section 4, we provide the experimental results on the UCI datasets and compare

978-1-7281-6694-0/20/$31.00 ©2020 IEEE

the results with the traditional KNN algorithm and another previous work. Finally, a summary of our research is given in section 5.

II. Related Backgrounds

A. K Nearest Neighbor Algorithm

K Nearest Neighbors (KNN) is a classification algorithm that widely used in machine learning and data mining. KNN is a supervised learning which is a classification method that is not complicated. The algorithm can classify without creating a parametric model.

The process of KNN algorithm can be explained in 2 steps. Firstly, algorithm calculates the distance between an unseen instance and the learning data. Secondly, algorithm selects the smallest distance of the nearest k neighbors and assigns the answer class to an unseen instance by the majority answer.

KNN algorithm uses Euclidean distance to calculate how far an unseen instance to each point in the training data. This method is a basic measurement of distance between two points. The origin of the method comes from the Pythagorean Theorem [15] and commonly used in various types of work.

The Euclidean distance between two points P and Q is the length of the line segment P and Q, i.e. $Distance\ (P,Q)$, where $P = (p_1, p_2, \ldots, p_n)$ and $Q = (q_1, q_2, \ldots, q_n)$ are points in the Cartesian coordinate system. The distance between the points P and Q is calculated as equation (2-1) Let p_i be the i^{th} coordinate of P, q_i be the i^{th} coordinate of Q, and n be number of dimensions of the data

$$Distance\ (P,Q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \qquad (2\text{-}1)$$

However, the assumption of KNN that each attribute is equally important may lead to a wrong classification if attributes relate to the target class in different degrees. The Euclidean distance between two point (P and Q) for the weighted attributes can be calculated by multiplying attribute's weight with the squared between p_i and q_i [6-7].

B. Pearson correlation coefficient

Correlation coefficient is the value that explains the relationship between two variables such as the relationships between height and weight, wind strength and temperature, etc. The value of correlation is between -1 and 1 which indicates the direction of the relationship. If the value is positive, the relationship of both variables is in the same direction. In contrast, if the value is negative, the relationship of two variables is in the opposite direction [16]. The well-known methods to calculate the correlation coefficient are Pearson correlation, Spearman rank correlation, Tetrachoric correlation.

Karl Pearson proposed the Pearson correlation coefficient method which measures the relationship between two variables. The correlation values are shown in the ratio scale R_{XY}. Pearson correlation coefficient between vectors X and Y on N attributes can be calculated from the following equation.

$$R_{XY} = \frac{N\sum XY - (\sum X)(\sum Y)}{\sqrt{[N\sum X^2 - (\sum X)^2][N\sum Y^2 - (\sum Y)^2]}} \qquad (2\text{-}2)$$

III. Methodology

The proposed attribute weighting method for KNN algorithm can be described as follows. We defined W_i as the weight of the i^{th} attribute which conforms to its importance. Our goal is to find the optimal weights without spending too much searching time. We will explain the proposed method in 2 steps.

A. Rough search and the weighted K nearest neighbors method

- Divide the dataset into training data and test data. For training data, let A be a set of n attributes $\{a_1, a_2, \ldots, a_n\}$ in the data set and C be the target class vector. Determine the relationship between each attribute in A and the class vector C by calculating the absolute value of Pearson correlation coefficient, resulting in a complete set of correlation coefficients $PC = \{pc_1, pc_2, \ldots, pc_n\}$.
- Use equation (3-1) to calculate the initial weight w_i for each attribute by normalizing values in the set PC by their summation so that the sum of all normalized weights is equal to 1.

$$w_i = \frac{pc_i}{\sum_{i=1}^{n} pc_i} \qquad (3\text{-}1)$$

B. Fine tuning

We use the PSO algorithm to find the optimal weights by using the classification accuracy as the fitness function. In order to calculate the Euclidean distance, we use the weighted attribute equation (3-2) Let n be the number of attributes, w_i represents weight of the i^{th} attribute, p_i represents the i^{th} attribute of p, and q_i represents the i^{th} attribute of q. Among the nearest neighbors, we assign the category to an unseen instance by the majority answer.

$$Distance(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2 \cdot w_i^2} \qquad (3\text{-}2)$$

The following equations are used to update velocity and position (weight) of each particle.

$$v_b[i+1] = (\omega * v_b) + c_1 r_1 (GBest - v_b) + c_2 r_2 (PBest_b - w_b) \qquad (3\text{-}3)$$

$$w_b[i+1] = v_b[i+1] + w_b[i] \qquad (3\text{-}4)$$

Where $b = 1,2,3,\ldots,n$ represent indexes of all particles.

v_b is the velocity value which tells direction and distance of the particle.

w_b is the weight of each attribute.

$PBest_b$ is the best location of the particle b.

$GBest$ is the best location of the particle swarm.

ω is the inertia weight which equals to 0.729844.

r_1, r_2 are random values in the range [0,1].

c_1 and $c_2$are social and cognitive parameters which are equal to 1.49445.

It is worth to note that the boundary of each weight tuning in every iteration is limited within the range of ± 0.2 to prevent too large search space in running PSO algorithm which may result in a long running time.

$$(w_i - 0.2) \leq w_i \leq (w_i + 0.2) \qquad (3\text{-}5)$$

The flowchart explaining the process of the proposed method is shown in Figure 1.

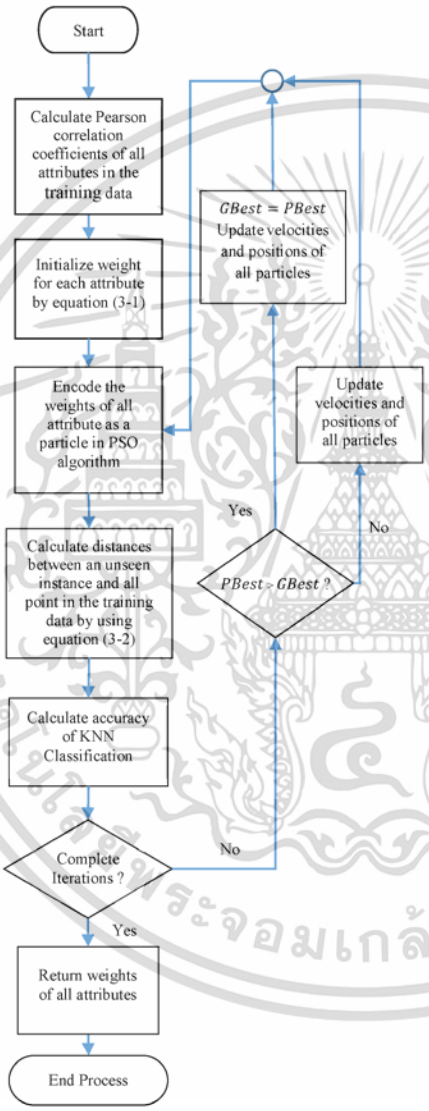


Figure 1. The flow chart of the proposed method.

IV. EVALUATION

The experiments are conducted on 6 standard datasets from the UCI repository. The characteristics of each dataset are shown in Table I.

TABLE I. DATASETS AND THEIR CHARACTERISTICS

Dataset	Number of instances	Number of Attributes	Number of Classes
Hepatitis	155	19	2
Heart Stat log	270	13	2
Climate Model Simulation	540	18	2
Movement Libras	360	90	15
Vehicle silhouette	946	18	4
Ionosphere	351	34	2

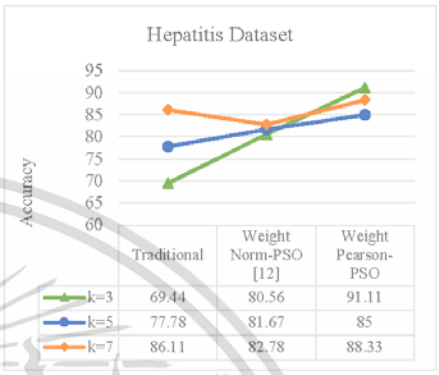
In the first experiment, we compare the classification accuracy between the traditional KNN algorithm and the proposed algorithm. The setting of the proposed method are as follows: the number of particles = 20, ω = 0.729844, c_1 and c_2 = 1.49445, number of PSO iterations = 100. We divide each dataset into training and test sets by the fraction of 0.75 and 0.25, respectively.

The number of nearest neighbors (K) for both algorithms are varied as 3, 5 and 7 in order to investigate the consistency of their performance. The experimental results are shown in Table II.

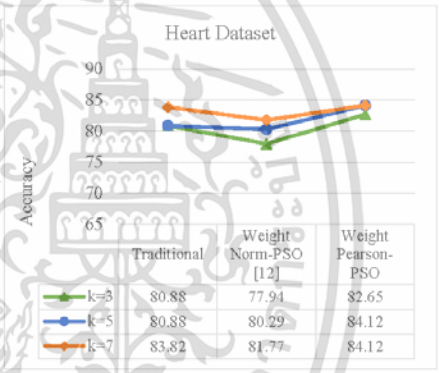
TABLE II. AVERAGE ACCURACY OF THE TRADITIONAL KNN AND THE PROPOSED METHOD.

Dataset	K	Traditional KNN	Pearson - Weighted KNN (SD)	Percent of Average Acc Improvement
Hepatitis	3	69.44	91.11 (3.239)	31.199%
	5	77.78	85.0 (2. 222)	9.286%
	7	86.11	88.33 (2.079)	2.581%
Heart Stat log	3	80.88	82.65 (2.161)	2.182%
	5	80.88	84.12 (1.715)	4.0004%
	7	83.82	84.12 (2.161)	0.351%
Movement Libras	3	77.78	82.67 (1.133)	6.286%
	5	75.56	81.11 (1.217)	7.353%
	7	72.22	77.56 (1.474)	7.385%
Climate Model Simulation	3	89.63	90.67 (0.889)	1.157%
	5	90.37	91.41 (0.811)	1.148%
	7	89.63	91.41 (0.889)	1.983%
Vehicle silhouette	3	69.34	71.98 (2.962)	3.810%
	5	68.39	72.83 (0.377)	6.48%
	7	70.75	71.46 (2.879)	1.000%
Ionosphere	3	82.95	86.36 (1.437)	4.11%
	5	81.82	85.91 (0.719)	5.00%
	7	80.68	84.54 (0.850)	4.79%

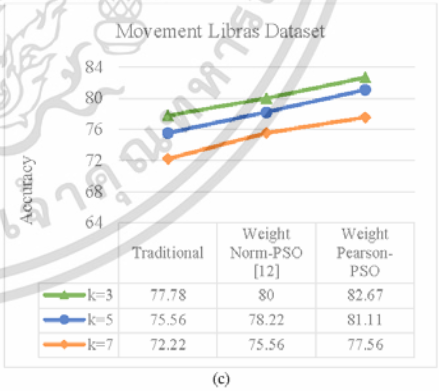
It is obvious that the proposed method can improve the classification accuracy of the traditional KNN algorithm in all datasets. In order to show efficiency of the proposed method, we give comparisons with another attribute weighting algorithm, the Normalized-PSO [10]. For a fair comparison, we control the common parameters of both algorithms as follows: the number of particles = 20, ω = 0.3, c_1 and c_2= 1.49445. The search cycle of PSO algorithm = 100 iterations. The comparisons of three algorithms on all datasets are shown in Figures 2 and 3.



(a)

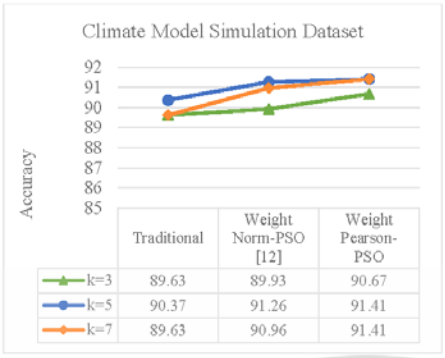


(b)



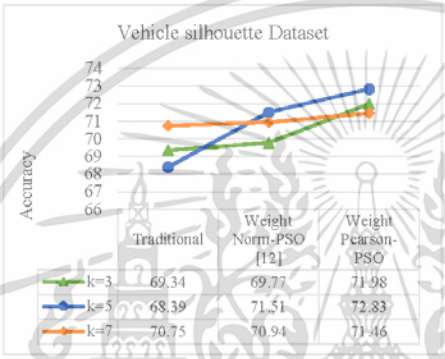
(c)

Figure 2. The average accuracy comparisons on Hepatitis, Heart and Movement Libras Datasets



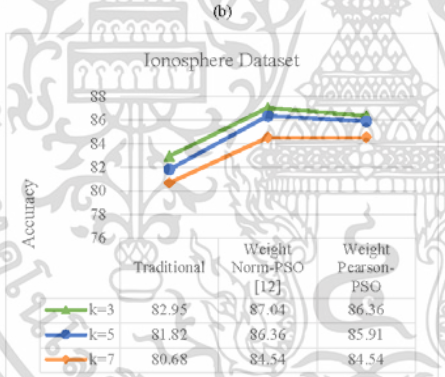
	Traditional	Weight Norm-PSO [12]	Weight Pearson-PSO
k=3	89.63	89.93	90.67
k=5	90.37	91.26	91.41
k=7	89.63	90.96	91.41

(a)



	Traditional	Weight Norm-PSO [12]	Weight Pearson-PSO
k=3	69.34	69.77	71.98
k=5	68.39	71.51	72.83
k=7	70.75	70.94	71.46

(b)

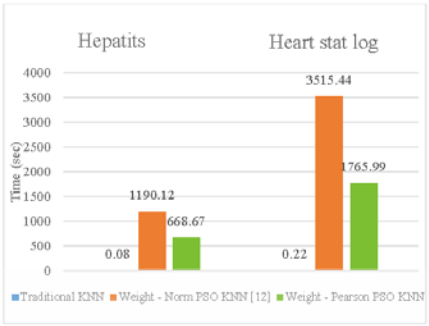


	Traditional	Weight Norm-PSO [12]	Weight Pearson-PSO
k=3	82.95	87.04	86.36
k=5	81.82	86.36	85.91
k=7	80.68	84.54	84.54

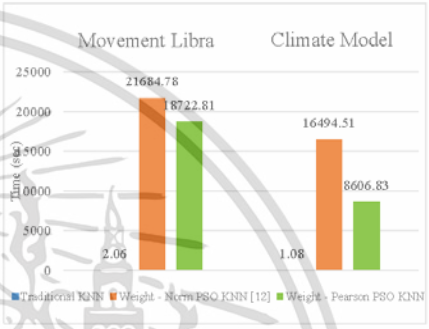
(c)

Figure 3. The average accuracy comparisons on Climate Model Simulation, Vehicle silhouette and Ionosphere Datasets

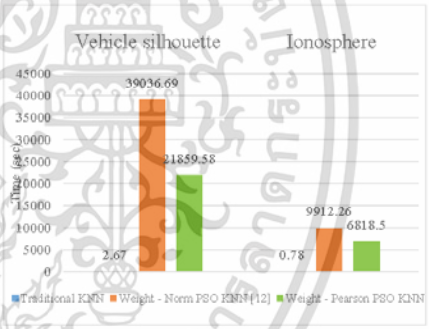
Two attribute weighting algorithms, weight Pearson-PSO and Weight Norm-PSO, significantly improve the classification accuracy of the traditional algorithm. Although the classification accuracies of both attribute weighting algorithms are competitive, the proposed method spends significantly less running time than Weight Norm-PSO. We illustrate the running time comparisons in Figures 4 and 5.



(a)



(b)



(c)

Figure 4. The running time comparisons

The comparisons in Figures 4 indicate that the proposed method (Weighted-Pearson PSO KNN) spends much less running time than that of Weight-Norm PSO KNN about 50% in most datasets. Although the running time of the traditional KNN is smallest, the superior classification accuracy of the proposed method is a reasonable trade-off.

V. Discussion

In this section, we explain about the importance of Pearson correlation coefficient which can significantly increase the efficiency of the KNN algorithm. The correlation values represent the relationship between each attribute and its target class. We have

experimented with some datasets and found that using Pearson correlation coefficient with weighted-KNN algorithm without fine-tuning method can improve the classification accuracy of the traditional KNN algorithm in almost datasets. Therefore, it is wise to roughly initialize weights of attributes by Pearson correlation coefficients before fine tuning as the running time of PSO can be reduced. The experimental results are shown in Table III.

TABLE III. THE COMPARISONS WHEN PEARSON CORRELATION COEFFICIENTS ARE APPLIED TO WEIGHTED-KNN ALGORITHM AND TRADITIONAL KNN

Dataset	K	Traditional KNN	Pearson Weighted KNN (NON - PSO)
Hepatitis	3	69.44	88.889
	5	77.78	80.556
	7	86.11	83.334
Movement Libras	3	77.78	82.222
	5	75.56	76.667
	7	72.22	72.222
Ionosphere	3	82.95	86.3636
	5	81.82	85.227
	7	80.68	85.227

VI. Conclusion

In this research, we proposed the algorithm to improve the KNN classification by using attribute weighting calculated from Pearson correlation coefficient and PSO fine tuning. From the experimental results, we found that the proposed algorithm can improve the accuracy of traditional KNN method and use less running time than the related work.

The future work of the proposed method is to find an appropriate boundary for weight adjusting in PSO algorithm. We currently use $\pm 2\%$ of the present weight value as the boundary to prevent too large search space of PSO algorithm. However, different problems may have their own boundaries. Finding the proper value of boundary is the next challenging issue that can improve the proposed method.

References

[1] Pooja Rani, "A Review of various KNN Techniques", International Journal for Research in Applied Science & Engineering Technology (IJRASET), Vol. 5, No. 8, August 2017, pp. 1174-1179.

[2] Jingwen Sun, Weixing Du and Niancai Shi, "A Survey of kNN Algorithm", Information Engineering and Applied Computing (2018), 2018.

[3] DR.O. Obulesu, M. Mahendra and M. ThrilokReddy, "Machine Learning Techniques and Tools: A survey", 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), July 2018, pp. 605-611.

[4] Xian Liang, Fuheng Qu, Yong Yang and Hua Cai, "An Improved ID3 Decision Tree Algorithm Based on Attribute Weighted", International Conference on Civil, Materials and Environmental Sciences (CMES 2015), pp. 613 –615.

[5] Sona Taheri, John Yearwood, Musa Mammadov and Sattar Seifollahi, "Attribute Weighted Naive Bayes Classifier Using a Local Optimization", Neural Computing and Applications, April 2014.I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[6] Zhang Li, Zhang Chengjin, Xu Qingyang and Liu Chunfa, "Weigted-KNN and its application on UCI", 2015 IEEE International Conference on Information and Automation, Aug. 2015, pp.1748-1750.

[7] Jie Huang, Yongqing Wei, Jing Yi and Mengdi Liu, "An Improved kNN Based on Class Contribution and Feature Weighting", 2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Feb. 2018

[8] Muhammad Ejazuddin Syed, "Attribute weighting in K-nearest neighbor classification", M.Sc. thesis University of Tampere Computer Science School of Information Sciences, November 2014.

[9] Jianping Gou, Lan Du b, Yuhong Zhang and Taisong Xiong, "A New Distance-weighted k -nearest Neighbor Classifier", Journal of Information & Computational Science 9, June 2012, pp.1429–1436

[10] Sahibsingh A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rule", IEEE Transactions on Systems, Man, and Cybernetics, Volume: SMC-6 , Issue: 4 , April 1976, pp.325 – 327.

[11] J. Kennedy and R. Eberhart, "Particle swarm optimization", Proceedings of ICNN'95 - International Conference on Neural Networks, 27 Nov.-1 Dec. 1995.

[12] Junjie Guo, Jin Gou, Cheng Wang and Wei Luo, "The Normalized-PSO and Its Application in Attribute Weighted Optimal Problem", 2016 Third International Conference on Trustworthy Systems and their Applications (TSA), Sept. 2016, pp.48-53.

[13] Chi Yu-hong, Sun Fu-chun, Wang and Wei-jun, "An Improved Particle Swarm Optimization Algorithm with Search Space Zoomed Factor and Attractor[J]", Chinese Journal of Computers,2011,34(1), pp.115-130.

[14] Qinghua Cao and Yu Liu, "A KNN Classifier with PSO Feature Weight Learning Ensemble [C]", 2010 International Conference on Intelligent Control and Information Processing, Aug. 2010, pp.110-114.

[15] Michel Marie Deza Michel Marie and Elena Deza, "Encyclopedia of Distances", Springer-Verlag Berlin Heidelberg, 2009, pp.94

[16] Gene V. Glass and Kenneth D. Hopkins, "Statistical Methods in Education and Psychology, 3rd Edition", Pearson 3 edition, Jun. 2008.

[17] Halil Yigit, "A weighting approach for KNN classifier", 2013 International Conference on Electronics, Computer and Computation (ICECCO), Nov. 2013, pp.228-23