การปรับปรุงประสิทธิภาพของการตรวจจับสิ่งผิดปกติสำหรับการวิเคราะห์ปูมแบบปรับขนาดได้

นายธนชัย จิระจันทร์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2557

PERFORMANCE IMPROVEMENT OF ANOMALY DETECTION FOR SCALABLE LOG ANALYSIS

Mr. Thanachai Jirachan

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2014

หัวข้อวิทยานิพนธ์	การปรับปรุงประสิทธิภาพของการตรวจจับสิ่งผิดปกติสำหรับการวิเคราะห์ปูมแบบปรับขนาดได้
โดย	นายธนชัย จิระจันทร์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร.บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา)

.....กรรมการ

(รองศาสตราจารย์ ดร.กุลธิดา โรจน์วิบูลย์ชัย)

.....กรรมการภายนอกมหาวิทยาลัย

(ดร.พงศ์ธวัช ชีพพิมลชัย)

ธนชัย จิระจันทร์ : การปรับปรุงประสิทธิภาพของการตรวจจับสิ่งผิดปกติสำหรับการวิเคราะห์ปูมแบบปรับขนาดได้ (PERFORMANCE IMPROVEMENT OF ANOMALY DETECTION FOR SCALABLE LOG ANALYSIS) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร. เกริก ภิรมย์โสภา, 97 หน้า.

ในงานวิจัยนี้ผู้วิจัยได้นำเสนอวิธีการปรับปรุงประสิทธิภาพในการวิเคราะห์สิ่งผิดปกติในปูมขนาดใหญ่ เพื่อให้มีความสามารถในการตรวจสอบการบุกรุกระบบแบบไม่มีการชี้นำ งานวิจัยนี้เป็นการประยุกต์ใช้ ความสามารถของวิธีการตรวจจับข้อมูลแปลกแยกที่เรียกว่า Kolmogorov-Smirnov and Efron Outlier Detection algorithm (KSE-test) และ การจัดกลุ่มข้อมูลด้วย K-Means algorithm ซึ่งทั้งสองวิธีนี้มีความซับซ้อนทางเวลาเป็นแบบเชิงเส้น เพื่อให้สามารถประมวลผลการตรวจจับข้อมูลแปลกแยกในปูมขนาดใหญ่ได้อย่างรวดเร็ว และ ยังคงประสิทธิภาพของผลลัพธ์ที่ดี คือ มีอัตราการตรวจพบข้อมูลแปลกแยกสูง และ อัตราการจำแนกผิดพลาดต่ำ ในการตรวจสอบความถูกต้อง ข้อมูลจาก KDD'99 ได้ถูกนำมาใช้ในการทดสอบ เพื่อหาค่า Threshold และ ประมาณค่า K ที่เหมาะสม สำหรับวิธีการที่นำเสนอ ผลที่ได้มีความเที่ยงตรงในการตรวจสอบการบุกรุกข้อมูลระบบมากขึ้น และ ความผิดพลาดน้อยลง กว่าการจำแนกโดยใช้วิธี KSE-test เพียงอย่างเดียว ในขณะที่ยังคงประสิทธิภาพเชิงเวลาเป็นเชิงเส้น นอกจากนี้ ผู้วิจัยยังได้แสดงการทดสอบประสิทธิภาพของงานที่นำเสนอว่าความสามารถขยายระบบ ด้วยวิธีการประมวลผลแบบขนาน บนแพลตฟอร์ม Apache Spark ทำให้สามารถลดระยะเวลาในการประมวลผลได้โดยการเพิ่มจำนวนเครื่องที่ใช้ในการประมวลผล

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2557

5670212121 : MAJOR COMPUTER ENGINEERING

KEYWORDS: OUTLIER DETECTION / LOG ANALYSIS / ANOMALY DETECTION / INTRUSION DETECTION / CLUSTERING

THANACHAI JIRACHAN: PERFORMANCE IMPROVEMENT OF ANOMALY DETECTION FOR SCALABLE LOG ANALYSIS. ADVISOR: ASST. PROF. KRERK PIROMSOPA, Ph.D., 97 pp.

We proposed a scalable outlier detection method to identify outliers in large datasets with a goal to create unsupervised intrusion detection. In our work, the strength of Kolmogorov-Smirnov and Efron Outlier Detection algorithm (KSE-test) and K-means clustering algorithm, both with linear time complexity, are combined to create fast outlier detection. While still maintaining high detection rate and low false alarm rate, our method can easily be paralleled for processing a large data set. The result is then applied with a predefined threshold in order to create efficient intrusion detection. We validate our method using the KDD'99 dataset. With the appropriate values of threshold and value of K in our proposed method, the results yield higher detection rate and lower false alarm rate. While scaling linearly, the accuracy of our method is also improved from those of pure KSE-test-based methods. Moreover, we propose a proof-of-concept parallel version of our proposed method that works on Apache Spark platform, which greatly reduces execution time and easily scales up by adding more machines to the cluster.

Department: Computer Engineering Student's Signature .....
Field of Study: Computer Engineering Advisor's Signature .....
Academic Year: 2014

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงลงด้วยดี มีองค์ประกอบมาจากส่วนสำคัญที่ขาดไม่ได้ คือ ความรู้ทางวิชาการที่คณาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ทุกท่านที่ได้ประสิทธิ์ประสาทให้แก่ข้าพเจ้า อีกทั้งประกอบกับคำแนะนำต่างๆที่เป็นประโยชน์ ต่อการดำเนินการวิจัย โดยเฉพาะอย่างยิ่ง การดูแล และ ให้ความช่วยเหลืออย่างสม่ำเสมอ จาก ผศ. ดร.เกริก ภิรมย์โสภา อาจารย์ที่ปรึกษาวิทยานิพนธ์ของข้าพเจ้า ซึ่งเป็นแบบอย่างทั้งในด้านการศึกษา และ ยังเป็นแบบอย่างสำคัญในด้านการดำเนินชีวิต ข้าพเจ้าจึงขอกราบแสดงความขอบพระคุณในความเมตตากรุณามา ณ ที่นี้

ขอขอบพระคุณคณาจารย์ และ คณะกรรมการสอบวิทยานิพนธ์ทุกท่าน ได้แก่ ผศ. ดร. ณัฐวุฒิ หนูไพโรจน์ (ประธานกรรมการสอบวิทยานิพนธ์), รศ. ดร.กุลธิดา โรจน์วิบูลย์ชัย (กรรมการสอบสอบวิทยานิพนธ์) และ ดร.พงศ์ธวัช ชีพพิมลชัย(กรรมการจากภายนอกมหาวิทยาลัย) ที่ได้ชี้แนะแนวคิดในการพัฒนาการวิจัย แนวทางการปรับปรุงแก้ไข เพิ่มเติมในส่วนที่บกพร่อง และ รวมถึงชี้แนะวิธีการนำเสนอ เพื่อให้งานวิจัยสำเร็จลุล่วงครบถ้วนตามเป้าหมาย

ขอขอบคุณทุนอุดหนุนการศึกษาระดับบัณฑิตศึกษาจากบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย เพื่อเฉลิมฉลองวโรกาสที่พระบาทสมเด็จพระเจ้าอยู่หัวภูมิพลอดุลยเดชทรงเจริญพระชนมายุครบ 72 พรรษา สำหรับการสนับสนุนค่าใช้จ่ายในระหว่างการศึกษาและการทำวิจัย และ ที่ขาดไม่ได้คือการสนับสนุนในทุกๆด้าน จากบิดาและมารดาของข้าพเจ้า ที่คอยส่งเสริม และสนับสนุนในทุกการตัดสินใจของข้าพเจ้า รวมถึงกำลังใจอันอบอุ่น ซึ่งมีคุณค่ามากมายต่อการเอาชนะอุปสรรคนานัปการ

ขอขอบคุณ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่เป็นบ้านอันแสนอบอุ่นตลอดเวลาที่ได้อยู่ในชายคาบ้านหลังที่สองหลังนี้ ที่เป็นศูนย์รวมทำให้ข้าพเจ้าได้มาพบเพื่อนผองพี่น้อง และ กัลยาณมิตรที่มีน้ำใจไมตรีอันดีต่อกัน

สุดท้ายนี้ขอกล่าวคำขอบพระคุณ รายนามบุคคลดังต่อไปนี้ ซึ่งมีคุโณปการสำคัญยิ่งต่องานวิจัยฉบับนี้ อันประกอบด้วย นายสมิทธ์ ธรรมบำรุง, นายพิทยุตม์ ตั้งสัจจะธรรม, นายมานะ บวรผดุงกิตติ, นายภาสกร ยูถะสุนทร์, นางสาวกรกนก ขาวอำไพ และ นายภาสกร ทองสันตติ์ ที่ได้เอื้อเฟื้อทั้งแรงกาย กำลังสติปัญญา ความรู้ ประสบการณ์ และ คำแนะนำ ในการดำเนินงานและการแก้ไขปัญหาต่างๆตลอดมา งานวิจัยนี้คงสำเร็จลงได้ยาก หากปราศจากความเอื้อเฟื้อจากทุกท่าน จึงขอเอ่ยนามเพื่อแสดงความขอบคุณสำหรับความกรุณาไว้ ณ ที่นี้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญภาพ ..... ฎ

สารบัญตาราง ..... ณ

บทที่ 1 บทนำ 1

บทที่ 2 สมมติฐาน วัตถุประสงค์ และ ขอบเขตของการวิจัย 3

2.1 สมมติฐานการวิจัย 3

2.2 วัตถุประสงค์ 3

2.3 ขอบเขตของการดำเนินงาน 3

บทที่ 3 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 4

3.1 คำจำกัดความ 4

3.1.1 Intrusion 4

3.1.2 Intrusion Detection 4

3.1.3 IDS (Intrusion Detection System) 4

3.1.4 Misuse Detection 4

3.1.5 Anomaly Detection 4

3.1.6 Log file 4

3.1.7 Unsupervised learning 5

3.1.8 Clustering 5

3.1.9 Anomalies 5

หน้า

3.1.10 Noise 5
3.1.11 Outlier 5
3.2 ทฤษฎีที่เกี่ยวข้อง 6
3.2.1 IDS (Intrusion Detection System) 6
3.2.1.1 Host-based Intrusion Detection System 6
3.2.1.2 Network Intrusion Detection System 6
3.2.2 Anomaly Detection 6
3.2.2.1 Clustering Based Anomaly Detection 7
3.2.3 K-Means algorithm 9
3.2.4 Two sample Kolmogorov-Smirnov test 10
3.2.5 Kolmogorov-Smirnov and Efron Outlier Detection (KSE-test) 13
3.2.6 A MapReduce Programming Model 14
3.2.7 Apache Spark 15
3.2.8 A Confusion Matrix 16
3.3 งานวิจัยที่เกี่ยวข้อง 18
3.3.1 Robust, Scalable Anomaly Detection for Large Collections of Images 18
3.3.2 Pruning Based Method for Outlier Detection 18
3.3.3 Applying Hadoop for log analysis toward distributed IDS 20
3.3.4 An analysis of suitable parameters for efficiently applying K-means clustering to large TCPdump dataset using Hadoop framework 21
3.3.5 Log Analysis in Cloud Computing Environment with Hadoop and Spark 21
3.3.6 Improving the Quality of Clustering Using Cluster Ensembles 23
3.3.7 An Intrusion Detection System Based on the Clustering Ensemble 24

หน้า

3.3.8 A Novel Approach for Outlier Detection and Clustering Improvement 26
3.3.9 An Intrusion Detection Method Based on Outlier Ensemble Detection ... 27
3.3.10 A New Semi-unsupervised Intrusion Detection Method Based on Improved DBSCAN ... 27
3.3.11 สรุปงานวิจัยที่เกี่ยวข้อง ... 28
บทที่ 4 ขั้นตอนการดำเนินงาน ... 29
4.1 การออกแบบงานวิจัย ... 29
4.2 วิธีการดำเนินงาน ... 30
4.2.1 การจัดเก็บข้อมูล (Data Collection) ... 30
4.2.2 การเตรียมข้อมูล (Data Preprocessing) ... 30
4.2.3 Calculating KSE-score and Labeling Outlier Instances ... 31
4.2.4 K-means Clustering and Creating Normal Profile ... 38
4.2.5 Enhancing Accuracy ... 40
4.2.6 Labeling Clusters ... 41
4.2.7 ประเมินความแม่นยำโดยการคำนวณ Detection Rate(DR) และ False Positive Rate(FPR) ... 43
บทที่ 5 การวัดและประเมินผล ... 44
5.1 หลักเกณฑ์การวัดและประเมินผล ... 44
5.2 การเตรียมข้อมูลทดสอบ ... 44
5.3 รายละเอียดของระบบที่ใช้ในการทดสอบ ... 44
5.4 ทดสอบความสัมพันธ์ของขนาด Sample และ Threshold ที่มีผลต่อความแม่นยำในการจำแนกการบุกรุกระบบ ... 45
5.5 ความแม่นยำในการจำแนกการบุกรุกระบบ ... 47
5.6 เปรียบเทียบผลลัพธ์การจำแนกการบุกรุกกับวิธีการอื่นๆ ... 84

5.7 วิเคราะห์ประสิทธิภาพเชิงเวลาของวิธีการที่นำเสนอ และ เปรียบเทียบกับวิธีการประเภทอื่นๆ 84

5.8 การทดสอบเวลาในการประมวลผลจริงเปรียบเทียบกับวิธีการอื่นๆ 85

5.9 การทดสอบการประมวลผลแบบขนาน บน platform Apache Spark 87

บทที่ 6 สรุปผลการทดลอง 90

6.1 สรุปผลการทดลอง 90

6.2 ประโยชน์ที่ได้รับจากงานวิจัย 91

6.3 แนวทางการวิจัยในอนาคต 92

รายการอ้างอิง 93

ประวัติผู้เขียนวิทยานิพนธ์ 97

สารบัญภาพ

ภาพที่ 1 สำดับขั้นตอนการทำงานของ K-Means Algorithm 9

ภาพที่ 2 แสดงวิธีการเปรียบเทียบคะแนน ของ Kolmogorov – Smirnov One Sample Test[8] 11

ภาพที่ 3 ภาพตัวอย่างขั้นตอนการคำนวณการคิดระยะห่างสูงสุดระหว่างสองฟังก์ชัน ของ KS-test[9] 12

ภาพที่ 4 ส่วนขยายของ Apache Spark[16] 15

ภาพที่ 5 Table of confusion and relationships among terms[19] 16

ภาพที่ 6 ขั้นตอนการทำงานของวิธีการ Cluster pruning[21] 19

ภาพที่ 7 แสดงความสัมพันธ์ของเวลาการทำงาน กับ ขนาดภาระงาน และ จำนวนเครื่องที่ใช้ 20

ภาพที่ 8 ผลการทดลองการจัดจำแนกการบุกรุกด้วย K-Means algorithm ที่ K=25 21

ภาพที่ 9 platform ที่ใช้ในการทดสอบ[25] 22

ภาพที่ 10 เวลาที่ใช้ประมวลผลระหว่าง Hadoop เปรียบเทียบ Spark ด้วย K-Means และ PageRank algorithm 23

ภาพที่ 11 Cluster ensemble framework[26] 24

ภาพที่ 12 The Diagram of EA algorithm[7] 24

ภาพที่ 13 The flowchart of EA algorithm[7] 25

ภาพที่ 14 The Diagram of EAIDS[7] 25

ภาพที่ 15 Algorithm ODC[4] 26

ภาพที่ 16 แผนผังการเดินทางของข้อมูล และ ผลลัพธ์ที่ได้จากแต่ละขั้นตอน 29

ภาพที่ 17 ลำดับขั้นตอนการทำงานของ KSE-test ที่มีการปรับปรุงแก้ไขให้เหมาะกับงานวิจัยนี้ 31

ภาพที่ 18 ทำการสุ่มข้อมูล Sample1 และ Sample2 จากชุดข้อมูล 32

ภาพที่ 19 การคำนวณ Euclidean distance เพื่อสร้าง Distance Matrix 33

ภาพที่ 20 คำนวณ Euclidean distance เก็บลงในแถวถัดไปของ Distance Matrix 33

ภาพที่ 21 Distance Matrix หลังจากการคำนวณเสร็จสิ้น 34
ภาพที่ 22 คำนวณ KS-score ระหว่าง จุดที่เราสนใจ เทียบกับข้อมูลตัวแรกของ Sample1 35
ภาพที่ 23 คำนวณ KS-score ระหว่าง จุดที่เราสนใจ เทียบกับข้อมูลของตัวที่สองใน Sample1..... 35
ภาพที่ 24 การคิด KSE-score หลังจากคำนวณ KS-score ครบทุกคู่ 36
ภาพที่ 25 ภาพแสดงวิธีการใช้ Threshold สร้างรายการจำแนกชนิดข้อมูลจากคะแนน KSE-score 37
ภาพที่ 26 การแบ่งขอบเขตข้อมูลปกติและข้อมูลการบุกรุก โดยการกำหนดสัดส่วน Nsize 39
ภาพที่ 27 การใช้ Normal Profile เพิ่มความถูกต้องแม่นยำของผลลัพธ์ 41
ภาพที่ 28 ตัวอย่างการทำ Majority vote ในกลุ่มข้อมูลผลลัพธ์ 42
ภาพที่ 29 รายละเอียดของเครื่องที่ใช้ในการทดสอบ 45
ภาพที่ 30 ความสัมพันธ์ ระหว่าง Detection Rate กับ Threshold 46
ภาพที่ 31 ความสัมพันธ์ ระหว่าง False Positive Rate กับ Threshold 46
ภาพที่ 32 ความสัมพันธ์ ระหว่าง Accuracy กับ Threshold 46
ภาพที่ 33 ROC ระหว่าง TPR และ FPR ของ KSE-test 53
ภาพที่ 34 ROC ระหว่าง TNR และ FNR ของ KSE-test 53
ภาพที่ 35 ผลการทดลอง Detection Rate ของ KSE-test 54
ภาพที่ 36 ผลการทดลอง False Positive Rate ของ KSE-test 54
ภาพที่ 37 ผลการทดลอง Accuracy ของ KSE-test 54
ภาพที่ 38 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=5) 55
ภาพที่ 39 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=5) 55
ภาพที่ 40 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=5) 56
ภาพที่ 41 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=5) 56
ภาพที่ 42 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=5) 56
ภาพที่ 43 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=10) 57

ภาพที่ 44 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=10)..... 57
ภาพที่ 45 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=10)..... 58
ภาพที่ 46 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=10)..... 58
ภาพที่ 47 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=10) 58
ภาพที่ 48 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=15)..... 59
ภาพที่ 49 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=15)..... 59
ภาพที่ 50 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=15)..... 60
ภาพที่ 51 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=15)..... 60
ภาพที่ 52 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=15) 60
ภาพที่ 53 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=20)..... 61
ภาพที่ 54 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=20)..... 61
ภาพที่ 55 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=20)..... 62
ภาพที่ 56 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=20)..... 62
ภาพที่ 57 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=20) 62
ภาพที่ 58 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=25)..... 63
ภาพที่ 59 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=25)..... 63
ภาพที่ 60 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=25)..... 64
ภาพที่ 61 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=25)..... 64
ภาพที่ 62 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=25) 64
ภาพที่ 63 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=30)..... 65
ภาพที่ 64 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=30)..... 65
ภาพที่ 65 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=30)..... 66
ภาพที่ 66 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=30)..... 66
ภาพที่ 67 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=30) 66

ภาพที่ 68 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=35)..... 67
ภาพที่ 69 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=35)..... 67
ภาพที่ 70 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=35)..... 68
ภาพที่ 71 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=35)..... 68
ภาพที่ 72 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=35) 68
ภาพที่ 73 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=40)..... 69
ภาพที่ 74 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=40)..... 69
ภาพที่ 75 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=40)..... 70
ภาพที่ 76 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=40)..... 70
ภาพที่ 77 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=40) 70
ภาพที่ 78 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=45)..... 71
ภาพที่ 79 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=45)..... 71
ภาพที่ 80 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=45)..... 72
ภาพที่ 81 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=45)..... 72
ภาพที่ 82 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=45) 72
ภาพที่ 83 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=50)..... 73
ภาพที่ 84 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=50)..... 73
ภาพที่ 85 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=50)..... 74
ภาพที่ 86 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=50)..... 74
ภาพที่ 87 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=50) 74
ภาพที่ 88 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=55)..... 75
ภาพที่ 89 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=55)..... 75
ภาพที่ 90 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=55)..... 76
ภาพที่ 91 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=55)..... 76

ภาพที่ 92 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=55) 76
ภาพที่ 93 เปรียบเทียบ Detection Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 1..... 78
ภาพที่ 94 เปรียบเทียบ False Positive Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 1..... 78
ภาพที่ 95 เปรียบเทียบ Accuracy ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 1..... 78
ภาพที่ 96 เปรียบเทียบ Detection Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 2..... 79
ภาพที่ 97 เปรียบเทียบ False Positive Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 2..... 79
ภาพที่ 98 เปรียบเทียบ Accuracy ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 2..... 79
ภาพที่ 99 เปรียบเทียบ Detection Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 3..... 80
ภาพที่ 100 เปรียบเทียบ False Positive Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 3..... 80
ภาพที่ 101 เปรียบเทียบ Accuracy ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 3 80
ภาพที่ 102 เปรียบเทียบ Detection Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 4..... 81
ภาพที่ 103 เปรียบเทียบ False Positive Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 4..... 81
ภาพที่ 104 เปรียบเทียบ Accuracy ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 4 81
ภาพที่ 105 เปรียบเทียบ Detection Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 5..... 82
ภาพที่ 106 เปรียบเทียบ False Positive Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 5..... 82
ภาพที่ 107 เปรียบเทียบ Accuracy ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 5 82
ภาพที่ 108 กราฟเปรียบเทียบเวลาในการประมวลผลจริงของวิธีการแบบต่างๆ..... 86
ภาพที่ 109 ลำดับขั้นตอนการทำงานของโปรแกรมคำนวณ KSE-Score แบบขนาน 87
ภาพที่ 110 แผนภูมิแสดงความสัมพันธ์ระหว่าง ขนาดของข้อมูล จำนวน node และ เวลาที่ใช้..... 89
ภาพที่ 111 แผนภาพแสดง Speed Up ต่อจำนวนของ Worker Node 89

สารบัญตาราง

ตารางที่ 1 เปรียบเทียบความสัมพันธ์ของความแม่นยำ กับ ขนาด Sample และ Threshold 45
ตารางที่ 2 Confusion Matrix แสดงผลการทดลองของข้อมูลชุดที่ 1 (90/10) 48
ตารางที่ 3 Confusion Matrix แสดงผลการทดลองของข้อมูลชุดที่ 2 (92/8) 49
ตารางที่ 4 Confusion Matrix แสดงผลการทดลองของข้อมูลชุดที่ 3 (94/6) 50
ตารางที่ 5 Confusion Matrix แสดงผลการทดลองของข้อมูลชุดที่ 4 (96/4) 51
ตารางที่ 6 Confusion Matrix แสดงผลการทดลองของข้อมูลชุดที่ 5 (98/2) 52
ตารางที่ 7 ค่าเฉลี่ยของผลลัพธ์จากวิธีการประมาณค่าตัวแปรที่นำเสนอ 83
ตารางที่ 8 เปรียบเทียบความสามารถในการจำแนกการบุกรุกระบบ 84
ตารางที่ 9 เปรียบเทียบคุณภาพผลลัพธ์ และ ความซับซ้อนเชิงเวลา ของวิธีการต่างๆ 85
ตารางที่ 10 เปรียบเทียบเวลาในการประมวลผลจริง แปรผันตามขนาดของข้อมูล 85
ตารางที่ 11 เวลาในการประมวลผล แบบต่างๆแปรผันตามขนาดของข้อมูล และจำนวน node..... 88

บทที่ 1
บทนำ

การตรวจสอบข้อมูลการบุกรุกเครือข่ายคอมพิวเตอร์ โดยใช้การวิเคราะห์จากข้อมูลการใช้งานที่ถูกเก็บบันทึกไว้ใน Log file ด้วยเทคนิค Anomaly detection ช่วยให้เราสามารถตรวจสอบรูปแบบของการบุกรุกใหม่ๆที่ไม่เคยรู้จักมาก่อนได้ ซึ่งแตกต่างจากวิธีการตรวจสอบการบุกรุกแบบ Misuse detection ซึ่งเป็นการตรวจสอบลักษณะข้อมูลที่เข้ามาเปรียบเทียบกับรูปแบบการจู่โจมที่เรารู้รูปแบบมาก่อน จึงจำเป็นต้องมีพื้นข้อมูลความรู้เกี่ยวกับการโจมตีรูปแบบต่างๆเพื่อสร้างกฎเกณฑ์ในการตรวจจำแนกไว้ล่วงหน้า แต่ในวิธีการของ Anomaly detection เพื่อตรวจสอบการบุกรุกระบบอาศัยหลักการที่ว่า พฤติกรรมการบุกรุก คือ พฤติกรรมที่มีลักษณะเบี่ยงเบนไปจากลักษณะการใช้งานทั่วไปในภาวะปกติ ทำให้การตรวจจำแนกพฤติกรรมการบุกรุกด้วยวิธี Anomaly detection สามารถทำได้โดยไม่ต้องอาศัยองค์ความรู้เกี่ยวกับรูปแบบพฤติกรรมการบุกรุกแต่ละชนิดมาก่อน ดังนั้นการตรวจจำแนกโดยการนำข้อมูลการใช้งานของผู้ใช้ที่ถูกเก็บบันทึกไว้ใน Log file มาทำการวิเคราะห์ และ สร้างองค์ความรู้ด้วยวิธีการเรียนรู้แบบไม่มีการชี้นำ (Unsupervised learning) ซึ่งเป็นวิธีการที่สะดวก ที่จะทำให้เราค้นพบรูปแบบหรือการโจมตีใหม่ๆที่ไม่เคยพบมาก่อน เพื่อนำไปใช้ในการตรวจจำแนกและคัดกรองพฤติกรรมการใช้งานต่างๆที่เข้าสู่ระบบ

การตรวจจับข้อมูลแปลกแยก (Anomaly detection) [1] เป็นอีกวิธีหนึ่งในการตรวจจำแนกข้อมูลผิดปกติ ที่มีลักษณะเบี่ยงเบนไปจากข้อมูลส่วนใหญ่ในชุดข้อมูล ซึ่งในปัจจุบันมีแนวคิดและวิธีการหลากหลายวิธี มีความนิยมในการนำมาประยุกต์ใช้ในแง่มุมที่หลากหลาย ตามความเหมาะสมกับคุณลักษณะเฉพาะของวิธีการ รวมถึงสามารถนำมาใช้ในการประมวลผลหาสิ่งผิดปกติในข้อมูลขนาดใหญ่ เพื่อหาสิ่งแปลกปลอม หรือ ข้อมูลการบุกรุกระบบ

การจัดกลุ่มข้อมูล(Clustering) [1] เป็นการเรียนรู้ Unsupervised learning วิธีหนึ่ง ซึ่งสามารถใช้ในการจัดจำแนกข้อมูลได้ง่ายและรวดเร็วมีความสามารถใช้งานได้กับลักษณะข้อมูลหลากหลายประเภท เหมาะสำหรับการนำมาใช้ในการแบ่งข้อมูลขนาดใหญ่ออกเป็นกลุ่มย่อยๆโดยที่ข้อมูลในแต่ละกลุ่มมีความสัมพันธ์ไปในทิศทางเดียวกัน โดยจัดจำแนกข้อมูลที่มีลักษณะใกล้เคียงกันไว้ในกลุ่มเดียวกัน และ จัดข้อมูลที่มีคุณลักษณะต่างกันอยู่คนละกลุ่มกัน โดยเราสามารถนำผลลัพธ์การจัดกลุ่มที่ได้มาทำการหารูปแบบของพฤติกรรมการใช้งานที่มีความผิดปกติจากข้อมูลการใช้งานที่ถูกบันทึกไว้ใน Log file เพื่อนำมาใช้ประโยชน์ในการสร้างกฎเพื่อคัดกรองพฤติกรรมบุกรุกเครือข่ายต่อไป

วิธีการที่นำเสนอในงานวิจัย “การปรับปรุงประสิทธิภาพของการตรวจจับสิ่งผิดปกติสำหรับการวิเคราะห์ปูมแบบปรับขนาดได้” ใช้วิธีการตรวจจับข้อมูลแปลกแยกที่เรียกว่า Kolmogorov-Smirnov and Efron Outlier Detection algorithm (KSE-test) ร่วมกับ การจัดกลุ่มข้อมูล เพื่อคัดกรองให้ผลการจำแนกกลุ่มข้อมูลมีความถูกต้องแม่นยำมากขึ้น เพื่อทำให้ผลลัพธ์มี Detection Rate ที่สูงขึ้น และ มี False Positive Rate ที่ต่ำลง โดยนำเสนอขั้นตอนวิธีการเตรียมข้อมูลที่เหมาะสมกับลักษณะของข้อมูลที่จะนำมาทำการประมวลผล นำมาประยุกต์เทคนิคในการทำ Anomaly detection สำหรับการทำ Intrusion detection บนชุดข้อมูล Log file ที่นำมาใช้ทดสอบ คือ KDD’99 (TCPdump log) โดยวิธีการประยุกต์ใช้การจัดกลุ่มข้อมูลด้วย K-Means algorithm มาทำการคัดกรองผลลัพธ์ที่ได้จากการทำ KSE-test เพื่อให้มีความแม่นยำมากขึ้นและมีความผิดพลาดน้อยลง และ มีความสามารถในการประมวลผลที่รวดเร็ว ใช้ทรัพยากรน้อย โดยทำการทดลองเปรียบเทียบผลลัพธ์จากวิธีการที่นำเสนอกับผลลัพธ์จากวิธีการเดิมแบบต่างๆ ที่มีอยู่ก่อนหน้า ผลลัพธ์ที่ได้คือมี Detection Rate ที่สูงกว่าหรือใกล้เคียงกับวิธีที่มีอยู่ก่อนหน้า และ มี False Positive Rate ที่ต่ำลงกว่าวิธีการเดิม โดยข้อดีของวิธีการที่สำเสนอสามารถคำนวณได้ภายในประสิทธิภาพเชิงเวลาเชิงเส้น อีกทั้งยังสามารถพัฒนาให้สามารถทำการประมวลผลแบบขนานเพื่อความสามารถในการขยายระบบเพื่อรองรับภาระงาน และ ลดเวลาที่ใช้ในการประมวลผลให้น้อยลง

บทที่ 2
สมมติฐาน วัตถุประสงค์ และ ขอบเขตของการวิจัย

2.1 สมมติฐานการวิจัย

1. วิธีการที่ออกแบบขึ้นมีประสิทธิภาพในการตรวจสอบและจำแนกพฤติกรรมการใช้งานปกติออกจากพฤติกรรมการบุกรุกระบบได้อย่างแม่นยำ
2. วิธีการที่ออกแบบมีความสามารถในการวิเคราะห์ปูม(Log) ขนาดใหญ่ และ มีจำนวนคุณลักษณะ (Attribute/Feature/Dimension) มากๆได้ดี

2.2 วัตถุประสงค์

1. เพื่อนำเสนอวิธีการตรวจจับสิ่งผิดปกติแบบไม่มีการชี้นำ(Unsupervised anomaly detection) ที่มีความสามารถในการใช้วิเคราะห์ปูมแบบปรับขนาดได้เพื่อทำการตรวจสอบการบุกรุกเครือข่าย
2. เพื่อนำเสนอวิธีที่มีความสามารถวิเคราะห์ข้อมูลที่มีจำนวน Dimension มากๆ ได้อย่างมีประสิทธิภาพ
3. เพื่อปรับปรุงประสิทธิผลของวิธีการที่ใช้ในการตรวจจำแนกพฤติกรรมบุกรุกระบบออกจากพฤติกรรมการใช้งานปกติให้มีความแม่นยำมากยิ่งขึ้น และ มีอัตราการผิดพลาดน้อยลง

2.3 ขอบเขตของการดำเนินงาน

1. ตรวจสอบการบุกรุกระบบด้วยการวิเคราะห์ข้อมูลการใช้งานจาก TCPdump log (KDD'99)
2. ตรวจสอบการบุกรุกระบบโดยประยุกต์ใช้วิธีการจัดกลุ่มข้อมูล (Clustering) และ เทคนิคการตรวจสอบข้อมูลแปลกแยก (Outlier detection) แบบไม่มีการชี้นำ (Unsupervised learning)
3. ข้อมูลที่ใช้ต้องมีสัดส่วนของ จำนวนข้อมูลการใช้งานปกติมากกว่าจำนวนข้อมูลที่เป็นการจู่โจมระบบ เป็นจำนวนมากๆ และ ข้อมูลการบุกรุกมีลักษณะแตกต่างจากข้อมูลปกติอย่างชัดเจน
4. เป็นการประมวลผลแบบ batch ไม่เป็น real-time
5. วัดผลด้วยความแม่นยำของผลลัพธ์ที่เพิ่มขึ้นเท่านั้น

บทที่ 3
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

3.1 คำจำกัดความ

3.1.1 Intrusion

Intrusion[2] คือ การบุกรุกระบบ การแทรกซึม หรือ การทำอันตรายต่อระบบ ในแง่มุมของ computer security

3.1.2 Intrusion Detection

Intrusion detection[2] คือการตรวจสอบพฤติกรรมที่ผิดปกติ ที่เป็นอันตรายต่อระบบ อาทิ การบุกรุกระบบ การแทรกซึม หรือ การทำอันตรายต่อระบบ ในแง่มุมของ computer security

3.1.3 IDS (Intrusion Detection System)

IDS[2] คือ ระบบตรวจสอบพฤติกรรมจู่โจมระบบ ทำงานโดยการนำข้อมูลกิจกรรมต่างๆในระบบมาทำการวิเคราะห์ โดยหาพฤติกรรมผิดปกติ แต่จะยังไม่มีการดำเนินการตอบสนองต่อการจู่โจมที่ปรากฏ เพียงแต่จะทำการรายงานการตรวจพบไปยังผู้ดูแลระบบให้ทราบเท่านั้น

3.1.4 Misuse Detection

Misuse Detection[2] คือ วิธีการตรวจสอบพฤติกรรมผิดปกติโดยนำคุณลักษณะของพฤติกรรมที่ต้องการตรวจสอบมาเปรียบเทียบกับคุณลักษณะในฐานความรู้ที่มีอยู่ เพื่อระบุว่ากิจกรรมนั้นเป็นพฤติกรรมจู่โจมระบบหรือไม่

3.1.5 Anomaly Detection

Anomaly Detection[1, 2] คือ วิธีการหารูปแบบพฤติกรรมที่มีความเบี่ยงเบนไปจากปกติ ไม่สอดคล้องกับข้อมูลส่วนใหญ่ที่เป็นรูปแบบของข้อมูลปกติ เพื่อให้สามารถแยกแยะพฤติกรรมการจู่โจมระบบออกจากรูปแบบปกติได้ด้วยวิธีการที่เหมาะสม

3.1.6 Log file

Log file[2] คือ ไฟล์ข้อมูลที่บันทึกกิจกรรมภายในระบบที่ถูกบันทึกไว้ ซึ่งสามารถใช้อ้างอิงถึงพฤติกรรมการใช้งานระบบของผู้ใช้ ซึ่งการเก็บบันทึกจะมีหลายรูปแบบ(format) ในแต่ละแถวข้อมูลมีการบันทึกค่า attribute ต่างๆแตกต่างกันไปตามจุดประสงค์การใช้งาน

3.1.7 Unsupervised learning

Unsupervised learning[1, 3] การเรียนรู้ที่สร้างองค์ความรู้ใหม่โดยไม่มีการชี้นำล่วงหน้า คือ ดำเนินการกับชุดข้อมูลที่ไม่ได้ถูกจัดจำแนกมาก่อน ไม่มีการระบุความรู้พื้นฐานเกี่ยวกับตัวข้อมูล เช่น ไม่มีการระบุ class label มากับตัวชุดข้อมูล

3.1.8 Clustering

Clustering[3] คือ การจัดกลุ่มข้อมูลโดยจัดจำแนกข้อมูลที่มีลักษณะใกล้เคียงกันไว้ในกลุ่มเดียวกัน

3.1.9 Anomalies

Anomalies[4] คือ รูปแบบของข้อมูลที่มีลักษณะไม่สอดคล้องกับรูปแบบข้อมูลส่วนใหญ่ที่เรานิยามว่าเป็นข้อมูลปกติ

3.1.10 Noise

Noise[4] คือ ปรากฏการณ์ข้อมูลที่แตกต่างแปลกแยกที่เกิดในข้อมูล อาจเป็นส่วนที่เราไม่สนใจที่จะนำมาวิเคราะห์

3.1.11 Outlier

Outlier [4] คือ ข้อมูลที่มีความแปลกแยกจากข้อมูลอื่นๆรอบข้าง เป็นลักษณะข้อมูลที่พบน้อยแต่เป็นข้อมูลที่มีความสำคัญ และเราให้ความสนใจ

3.2 ทฤษฎีที่เกี่ยวข้อง

3.2.1 IDS (Intrusion Detection System)

ระบบตรวจสอบพฤติกรรมจู่โจมระบบ[2] ทำงานโดยการนำข้อมูลกิจกรรมต่างๆในระบบมาทำการวิเคราะห์ โดยหาพฤติกรรมผิดปกติจากข้อมูลที่เก็บไว้ในระบบ เช่น Network traffic และ Log file มาวิเคราะห์หาความสำพันธ์ระหว่างกิจกรรมในระบบ กับ ข้อมูลความรู้ที่ถูกจัดเก็บไว้ เมื่อพบข้อมูลที่มีความเป็นไปได้ที่จะเป็นการจู่โจมระบบจะยังไม่มีการดำเนินการตอบสนองเหตุการณ์ที่ปรากฏ เพียงแต่จะทำการรายงานการตรวจพบไปยังผู้ดูแลระบบให้ทราบเท่านั้น ซึ่งการทำงานของ IDS สามารถทำได้หลายวิธี เช่น การตรวจสอบกิจกรรมภายในระบบเครือข่าย การสแกนหาช่องโหว่ของส่วนต่างๆภายในเครือข่าย ทำการทดสอบ Integrity ของข้อมูลส่วนที่สำคัญๆ โดย IDS มีด้วยกันหลายชนิด เช่น Host-based intrusion detection system และ Network based intrusion detection system ที่จะกล่าวถึงต่อไป

3.2.1.1 Host-based Intrusion Detection System

ระบบตรวจสอบการบุกรุกที่ทำงานอยู่บนแต่ละ Host [1, 2] ทำหน้าที่ตรวจสอบหรือคอยสอดส่องข้อมูล (Monitoring) เช่น Application log, System log และ Operation log เป็นต้น หากพบพฤติกรรมที่น่าจะเป็นพฤติกรรมที่ผิดปกติจะทำการแจ้งไปยังผู้ใช้ หรือ ผู้ดูแลระบบได้รับทราบ

3.2.1.2 Network Intrusion Detection System

ระบบที่คอยตรวจสอบการบุกรุกบนเครือข่าย [1, 2] โดยสอดส่องและติดตามการจราจรบนเครือข่ายทั้งระหว่างอุปกรณ์ และ ระหว่างส่วนต่างๆของเครือข่าย ซึ่งสามารถตรวจสอบกิจกรรมทั้งหมดในระบบที่ผ่านเครือข่ายได้ และ จะแจ้งเตือนผู้ใช้เมื่อพบพฤติกรรมที่มีแนวโน้มว่าจะเป็นการบุกรุกเพื่อให้ผู้ใช้หรือผู้ดูแลระบบได้รับทราบและตัดสินใจดำเนินมาตรการต่างๆต่อไป

3.2.2 Anomaly Detection

Anomaly detection[1] คือ วิธีการในการตรวจจำแนกข้อมูลที่มีลักษณะเบี่ยงเบนไปแปลกแยกไปจากข้อมูลปกติ ซึ่งสามารถประยุกต์ใช้เทคนิคต่างๆเข้ามาช่วย ไม่ว่าจะเป็นเทคนิคทางสถิติ เทคนิคของการทำ Data Mining และ Machine Learning เพื่อแบ่งแยก

และแสดงข้อมูลที่แปลกแยกออกมา ซึ่งการทำ Anomaly detection สามารถนำมาประยุกต์ใช้ในงานด้านการตรวจสอบการบุกรุกเพื่อรักษาความปลอดภัยของระบบได้อีกด้วย

3.2.2.1 Clustering Based Anomaly Detection

การจัดกลุ่มข้อมูล (Clustering) เป็นการเรียนรู้แบบไม่มีการชี้นำโดยจัดจำแนกรูปแบบของข้อมูลที่มีลักษณะใกล้เคียงกันไว้ในกลุ่มเดียวกัน เพื่อหารูปแบบที่ซ่อนอยู่ในข้อมูล สุดท้ายกลุ่มของข้อมูลที่ได้ออกมาเป็นผลลัพธ์ โดยสมาชิกในกลุ่มเดียวกันมีลักษณะใกล้เคียงกันมากๆ และ สมาชิกในคนละกลุ่มข้อมูลจะมีความแตกต่างกัน

ในแง่มุมของการทำ Anomaly Detection ได้กล่าวถึงเทคนิคการทำ Anomaly Detection ไว้มากมายหลายวิธี ซึ่งการจัดกลุ่มข้อมูล(Clustering) เป็นวิธีการหนึ่งที่สามารถนำมาใช้ในการจำแนกพฤติกรรมของข้อมูลที่มีความเบี่ยงเบนไปจากปกติ โดยวิธีการจัดกลุ่มสามารถทำได้ทั้งแบบ Unsupervised และ Semi-supervised คือ สามารถทำการจัดกลุ่มข้อมูลได้โดยไม่จำเป็นต้องอาศัยข้อมูลตั้งต้นในการสอน หรือ อาจใช้ข้อมูลที่มี Class label บางส่วนมาสอน (Training) เพื่อเพิ่มความแม่นยำในการจำแนกได้ และ การจัดกลุ่มข้อมูลยังมีความสามารถในการจำแนก Anomaly ออกมาได้หลายวิธี ทั้งในรูปแบบของ Noise ในข้อมูล หรือ กลุ่มของข้อมูลที่มีขนาดเล็กๆก็สามารถพิจารณาให้เป็นกลุ่มของ Anomaly ได้ จึงทำให้วิธีการจัดกลุ่มสามารถใช้งานได้กับหลากหลายลักษณะข้อมูล

Partitional Clustering

วิธีการจัดกลุ่มแบบแบ่งส่วน[3] การจัดกลุ่มข้อมูลแบบแบ่งส่วน คือจำแนกข้อมูลที่มีลักษณะใกล้เคียงกันเอาไว้ใน Region เดียวกันโดยจะพิจารณาข้อมูลให้อยู่ในกลุ่มที่ใกล้ที่สุด โดยพิจารณาจากระยะทางจากข้อมูลไปยังจุดตัวแทนของแต่ละกลุ่ม Partitional clustering แบ่งออกเป็นสองประเภทหลักๆคือ แบบ Centroid และ แบบ Medroid โดยแบบ Centroid จะแทน Cluster ด้วย จุดศูนย์กลาง(Gravity Centre of Instances) แบบ Medroid จะแทน Cluster ด้วยค่าเฉลี่ยของจุดที่อยู่ใกล้กับจุด Centre of instances มากที่สุด วิธีการนี้มีข้อดีคือ มีมีขั้นตอนวิธีที่ง่าย และ รวดเร็ว ใช้ทรัพยากรของระบบน้อย แต่มีข้อเสียคือ ต้องมีการกำหนดจำนวนกลุ่มล่วงหน้า ซึ่งจำนวนกลุ่มที่ใช้ มีผลต่อคุณภาพของผลลัพธ์จึงควรเลือกจำนวนกลุ่มที่เหมาะสมต่อชุดข้อมูล ผลลัพธ์ที่ได้จากการจัดกลุ่มไม่ทนทานต่อ Noise อันเกิดมาจากข้อบังคับของ Algorithm ที่กำหนดให้ข้อมูลทุกตัวต้องมีกลุ่ม กลุ่มข้อมูลที่ได้มีลักษณะเป็นทรงกลมกระจายตัวในวงรัศมีออกจากศูนย์กลางของกลุ่ม

Hierarchical Clustering

วิธีการจัดกลุ่มการจัดกลุ่มข้อมูลแบบลำดับขั้น[3] คือ การจัดกลุ่มข้อมูลในลักษณะของการจัดแบ่งข้อมูลออกเป็นลำดับขั้น โดยจะได้ผลลัพธ์เป็น ต้นไม้ของกลุ่มข้อมูล (Tree of Clusters) หรือเรียกว่า Dendrogram โดยแต่ละ Node ของต้นไม้ที่ได้ ทำให้สามารถแสดงผลลัพธ์ในระดับ granularity ต่างๆได้ ประกอบด้วยสองวิธีหลักๆ คือ Top-down (Divisive method) ที่จะเริ่มจากการแบ่งข้อมูลกลุ่มใหญ่ออกเป็นกลุ่มเล็กๆที่มีความใกล้เคียงกันภายในกลุ่มมากๆขนาดของกลุ่มจะค่อยๆเล็กลงตามแต่ลำดับขั้น และ Bottom-up (Agglomerative method) ที่จะเริ่มจากการรวมกลุ่มขนาดเล็กที่มีความใกล้เคียงกันขยายขึ้นมาเป็นกลุ่มขนาดใหญ่ขึ้นทีละขั้น มีข้อดีคือ มีความยืดหยุ่นสามารถแสดงความละเอียดของผลลัพธ์ได้หลายระดับ ได้ผลลัพธ์เป็นต้นไม้ของกลุ่มข้อมูล ที่สามารถเข้าใจได้ง่าย แต่มีข้อเสียคือ ใช้เวลาและ ทรัพยากรของระบบในการประมวลผลมาก มี Time Complexity $O(n^2 \log n)$ และ Space Complexity $O(n^2)$

Density-based Clustering

วิธีการจัดกลุ่มตามความหนาแน่น[3] คือ การจับกลุ่มของข้อมูลที่อยู่ใกล้เคียงกัน บริเวณใดที่สมาชิกมีสมาชิกล้อมรอบมากๆมีความหนาแน่นสูงจะถูกจัดเป็น cluster และบริเวณที่ความหนาแน่นต่ำเป็น noise โดยอาศัยหลักการ density reachable และ density connected ในการแบ่ง instance ออกเป็น3 ชนิดคือ

Core points เป็นจุดที่มีจำนวน Neighbor มากกว่าค่าที่กำหนดไว้ (Neighbor หาได้จากจุดที่มีระยะห่างจาก instance ที่เราสนใจ อยู่ในค่า Threshold ที่กำหนด) เป็น Density connected

Border points เป็นจุดที่มีจำนวน Neighbor น้อยกว่าจำนวนที่กำหนดไว้ แต่ยังคงมี Neighbor ที่เป็น Core point ทำให้เรียกได้ว่าเป็น Density reachable คือยังคงอยู่ในรัศมีของ Core point บางจุดแต่ไม่ครบตามจำนวน

Noise points เป็นจุดที่ไม่เป็นทั้ง Core points และ Border points

มีข้อดีคือ สามารถจัดกลุ่มข้อมูลให้ผลลัพธ์เป็นกลุ่มของข้อมูลที่มีรูปทรงใดก็ได้ และสามารถจำแนก Noise ของการจัดกลุ่มออกมาได้ แต่มีข้อเสียคือ ใช้เวลาและ ทรัพยากรของระบบในการประมวลผลมาก มี Time Complexity $O(n^2)$ และ Space Complexity $O(n^2)$

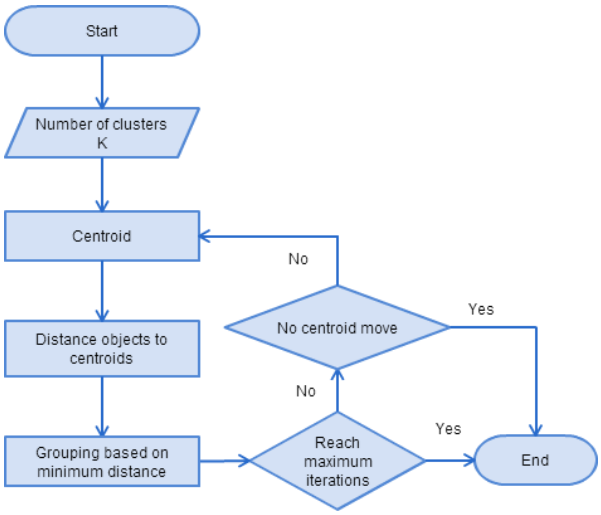
3.2.3 K-Means algorithm

K-Means algorithm[5] เป็น วิธีการจัดจำแนกกลุ่มของข้อมูลที่นิยมใช้กันแพร่หลาย สามารถจัดกลุ่มข้อมูลแบบไม่มีการชี้นำที่สามารถจัดจำแนกกลุ่มข้อมูลได้อย่างรวดเร็ว วิธีการจัดกลุ่มข้อมูลมีการทำงานขั้นตอนดังนี้ ขั้นตอนแรกกำหนดค่า K คือ จำนวนของกลุ่มข้อมูลผลลัพธ์ที่ต้องการจะจำแนก ซึ่งค่า K นี้จะถูกกำหนดโดยผู้ใช้ จากนั้นทำการสุ่มจุดกึ่งกลางของกลุ่มตามจำนวนค่า K ที่กำหนด แล้วทำการจัดข้อมูลแต่ละตัวในชุดข้อมูลไปยังกลุ่มที่มีจุดศูนย์กลางใกล้กับข้อมูลนั้นที่สุด การประเมินระยะห่างใช้การคำนวณระยะ Euclidean distance ระหว่างข้อมูลที่เราสนใจกับจุดกึ่งกลางของกลุ่มแต่ละกลุ่ม โดยการคำนวณระยะห่างระหว่างจุดสองจุด สามารถคำนวณได้ตามสมการดังนี้

$$d(x,y) = \sqrt{\sum_{i=0}^{m}(x_i - y_i)^2} \qquad (1)$$

โดย d(x,y) คือระยะ Euclidean distance ระหว่างจุด x คือจุดข้อมูลแรก และ จุด y คือจุดข้อมูลที่สอง และ m คือ จำนวนของคุณลักษณะ ของแต่ละจุดข้อมูล

นำระยะห่างที่คำนวณได้มาใช้ในการจัดข้อมูลเข้าไปยังกลุ่มที่ใกล้ที่สุด (คือ มีระยะห่างระหว่างจุดถึงจุดศูนย์กลางของกลุ่มนั้นสั้นที่สุด) หลังจากนั้นทำการคำนวณตำแหน่งจุดกึ่งกลางของกลุ่มใหม่โดยใช้การคำนวณค่าเฉลี่ยของสมาชิกทั้งหมดในกลุ่มได้เป็นพิกัดของจุดศูนย์กลางกลุ่มใหม่ แล้วทำการจัดกลุ่มอีกครั้งตามลำดับขั้นตอนเดิม ทำกระบวนการข้างต้นซ้ำจนกว่า จุดกึ่งกลางไม่เคลื่อนย้ายตำแหน่งไปจากรอบก่อนหน้า หรือ หยุดกระบวนการเมื่อจำนวนรอบของการทำซ้ำถึงจำนวนรอบที่มากสุดที่กำหนดไว้ ก็จะหยุดการประมวลผลและ ผลลัพธ์ที่ได้คือกลุ่มของข้อมูลแต่ละตัวในชุดข้อมูล สรุปขั้นตอนได้ดังแผนภาพการทำงานของ K-means algorithm ที่แสดงในภาพที่ 1



ภาพที่ 1 สำดับขั้นตอนการทำงานของ K-Means Algorithm

ลักษณะผลลัพธ์สุดท้ายที่ได้จะมีลักษณะเป็น กลุ่มของข้อมูลที่ประกอบด้วยสมาชิกในกลุ่ม ที่มีความคล้ายคลึงกันมากถูกจัดอยู่ภายในกลุ่มเดียวกัน และ สมาชิกที่อยู่ต่างกลุ่มมีความแตกต่างกันกันกับสมาชิกในกลุ่มอื่นๆ ซึ่งการบวนการประมวลผลทั้งหมดสามารถทำได้ในเวลาเชิงเส้น มีประสิทธิภาพเชิงเวลา O(nKL) มีค่า K และ ค่า L เป็นค่าคงที่

ข้อดี เป็นวิธีการที่ง่ายไม่ซับซ้อน และ มีประสิทธิภาพในด้านความเร็วในการประมวลผล ใช้ทรัพยากรน้อย สามารถปรับปรุงให้ทำการประมวลผลแบบขนานได้

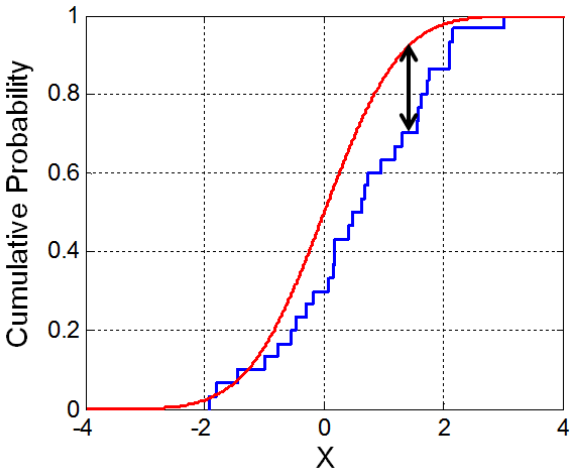
ข้อด้อย ผลลัพธ์ความแม่นยำขึ้นอยู่กับการเลือกค่า K ที่เหมาะสม การสุ่มจุดศูนย์กลางในตอนเริ่มต้น มีผลต่อผลลัพธ์การจัดกลุ่ม ผลลัพธ์สุดท้ายนั้นข้อมูลทุกตัวต้องถูกจัดอยู่ในกลุ่ม จัดกลุ่มแล้วกลุ่มที่ได้จะมีลักษณะทรงกลมรอบกระจายจุดศูนย์กลาง

เนื่องจากเป็นวิธีการที่ไม่ซับซ้อน และ มีความรวดเร็วในการประมวลผล จึงได้มีความพยายามนำเอา K-Means algorithm มาประยุกต์ใช้กับการประมวลผลงานขนาดใหญ่ รวมถึงการทำการตรวจจับข้อมูลแปลกแยก และ การบุกรุกระบบ อย่างหลากหลาย อาทิเช่นงานวิจัยที่นำเสนอใจ [4, 6, 7] แต่ก็ยังไม่ให้ผลลัพธ์ที่ดีเท่าที่ควร เนื่องจาก K-Means เป็นวิธีการที่ออกแบบมาเพื่อใช้ในการจัดกลุ่ม มากกว่าการนำมาใช้ในการตรวจจับข้อมูลที่มีความเบี่ยงเบนไปจากปกติ ดังนั้น ด้วยข้อจำกัดหลายๆประการ ดังที่ได้กล่าวถึงข้อด้อยของวิธีการจัดกลุ่มด้วยวิธี K-means มาแล้วในขั้นต้น เช่น ข้อมูลทุกตัวต้องมีกลุ่ม ทำให้ไม่สามารถใช้การจำแนก สิ่งแปลกปลอมที่อยู่นอกกลุ่มได้ และ การที่กลุ่มที่จัดได้จะมีรูปทรงได้จำกัด(ทรงกลมตามเป็นรัศมีรอบจุดศูนย์กลาง) ทำให้ไม่สามารถจัดกลุ่มในระดับที่ละเอียด

3.2.4 Two sample Kolmogorov-Smirnov test

Kolmogorov–Smirnov test (K–S test หรือ KS test) [8-10] เป็นวิธีการทดสอบทางสถิติแบบ นอนพาราเมตริก เพื่อทดสอบว่าสิ่งตัวอย่างอิสระสองกลุ่ม มีการกระจายตัว(การแจกแจง)เหมือนกันหรือไม่ สามารถแบ่งได้เป็นสองประเภทย่อยๆตามวิธีการใช้งาน ได้ดังนี้

Kolmogorov – Smirnov One Sample Test ใช้ในการเปรียบเทียบ การแจกแจงของกลุ่มตัวอย่าง ว่ามีการแจกแจงเดียวกับ Distribution function ที่ใช้อ้างอิงหรือไม่ (เปรียบเทียบความถี่ที่สังเกต กับ ความถี่ที่คาดหวัง) แสดงได้ดังภาพที่ 2



ภาพที่ 2 แสดงวิธีการเปรียบเทียบคะแนน ของ Kolmogorov – Smirnov One Sample Test[8]

จากภาพ เส้นสีแดงคือ Cumulative Distribution Function (CDF) ของ Distribution function ที่คาดหวัง และ เส้นสีน้ำเงิน คือ Empirical Distribution Function (ECDF) ของกลุ่มข้อมูลตัวอย่าง คะแนนที่ได้คือระยะห่างที่มากที่สุดของสอง Function เพื่อนำคะแนนที่ได้ไปเปรียบเทียบกับค่าวิกฤติ ถ้าค่าที่ได้ไม่เกินค่าวิกฤติ ถือว่าชุดข้อมูลตัวอย่างมีการกระจายตัวแบบเดียวกันกับ การกระจายตัวที่คาดหวัง โดย Empirical Distribution Function (ECDF) [11] สามารถหาได้ดังนี้ กำหนดกลุ่มข้อมูลขนาด N ตัว การหาค่า ECDF ได้จาก สมการที่(2)

$$ECDF(x) = \frac{number\ of\ element\ in\ the\ sample\ \leq x}{N} \qquad (2)$$

โดยที่ x คือ ค่าของข้อมูลค่าหนึ่ง ดังนั้น ค่า ECDF(x) คือ สัดส่วนของข้อมูลที่มีค่าน้อยกว่า x เมื่อเทียบกับจำนวนข้อมูลทั้งหมดในชุดข้อมูล

Kolmogorov-Smirnov Two Sample Test ใช้ในการเปรียบเทียบ การแจกแจงระหว่างสองกลุ่มตัวอย่าง ว่ามีการแจกแจงเดียวกันหรือไม่ (เปรียบเทียบความถี่ของประชากรกลุ่มแรก กับความถี่ของประชากรกลุ่มที่สอง) ซึ่งสามารถแบ่งย่อยได้เป็น การทดสอบแบบทางเดียว(ทดสอบว่าประชากรในชุดข้อมูลที่หนึ่งมีค่ามากกว่า หรือ น้อยกว่า ประชากรในชุดข้อมูลที่สอง) หรือ การทดสอบแบบสองทาง(ทดสอบว่า ประชากรในชุดข้อมูลที่หนึ่งเหมือน หรือ แตกต่างจากประชากรในชุดข้อมูลที่สอง)

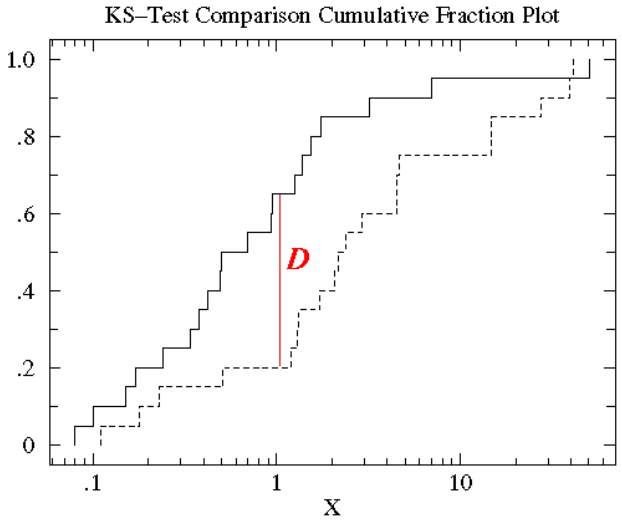
ในที่นี้เราจะบรรยายรายละเอียดในส่วนของวิธีการที่จะนำมาใช้ คือ Kolmogorov-Smirnov Two Sample Test แบบสองทาง ซึ่งใช้หลักการพิจารณาจาก Empirical Cumulative Distribution Function (ECDF) ของข้อมูลตัวอย่างทั้งสอง ว่ามีลักษณะใกล้เคียงกันหรือไม่ ถ้าลักษณะการแจกแจงความถี่สะสมเหมือนกัน ระยะห่างที่มากที่สุดระหว่างสองฟังก์ชันต่างกันไม่เกินค่าวิกฤติ ก็สามารถสรุปว่าสิ่งตัวอย่างทั้งสองมาจากประชากรเดียวกัน ถ้าลักษณะระยะห่างที่มากที่สุดระหว่างสองฟังก์ชันต่างกันมากกว่าค่าวิกฤติ ก็แสดงว่าสิ่งตัวอย่างมาจากประชากรที่ต่างกัน โดยในการทดสอบ จะใช้สมมติฐานหลักคือ H_0 คือ กลุ่มตัวอย่างทั้งสองชุดมีการกระจายตัวเหมือนกัน H_1 กลุ่มตัวอย่างทั้งสองชุดมีการกระจายตัวต่างกัน ดังเขียนแสดงได้ต่อไปนี้

สมมติฐาน H_0 : $F(X) = G(X)$ และ สมมติฐาน H_1 : $F(X) \neq G(X)$

F(X) คือ Distribution function ของกลุ่มข้อมูลชุดที่หนึ่ง

G(X) คือ Distribution function ของกลุ่มข้อมูลชุดที่สอง

วิธีการทดสอบสมมติฐานมีดังนี้ ในการทดสอบ สำหรับข้อมูลอย่างสองกลุ่ม จะพิจารณาจากการแจกแจงความถี่สะสมที่แตกต่างกันมากที่สุด หรือ ECDF ของกลุ่มตัวอย่าง สามารถแสดงวิธีการคำนวณคะแนนด้วยภาพได้ดังแสดงในภาพที่ 3



ภาพที่ 3 ภาพตัวอย่างขั้นตอนการคำนวณการคิดระยะห่างสูงสุดระหว่างสองฟังก์ชัน ของ KS-test[9]

KS-score คือ ระยะห่างสูงสุดระหว่างสองฟังก์ชันที่ได้จาก กรณี Kolmogorov-Smirnov Two Sample Test ทำการทดสอบ 2 ทาง ตามสมการที่(3)

$$\text{KS-score} = \sup_x |S_1(x) - S_2(x)| = \max |S_1(x) - S_2(x)| \qquad (3)$$

โดยที่

$S_1(x)$ = empirical cumulative distribution function ของกลุ่ม 1

= สัดส่วนของค่ากลุ่มที่หนึ่งที่มีค่า < x

$S_2(x)$ = empirical cumulative distribution function ของกลุ่ม 2

= สัดส่วนของค่ากลุ่มที่สองที่มีค่า < x

โดยจะยอมรับสมมติฐาน H_0 เมื่อ ค่า KS-score คือ ระยะห่างสูงสุดระหว่างฟังก์ชัน F(x)และ G(x) ค่าที่ได้จะมีค่าอยู่ในช่วง 0-1 หาก KS-score ที่ได้ไม่เกิน ค่าวิกฤติค่าหนึ่ง ที่ระดับนัยสำคัญ α (ค่าวิกฤติหาได้จากการเปิดตารางตามระดับนัยสำคัญที่กำหนด) แล้วทำการเปรียบเทียบค่า KS-score ที่คำนวณได้เทียบกับค่าวิกฤติจากตาราง หากค่าที่คำนวณได้มีค่าน้อยกว่าค่าวิกฤติ จะยอมรับสมมติฐาน H_0 คือ ข้อมูลตัวอย่างทั้งสองชุดมีการกระจายตัวเหมือนกัน แต่หากค่า KS-score ที่คำนวณได้มากกว่าค่าวิกฤติ จะปฏิเสธสมมติฐาน H_0 และยอมรับสมมติฐาน H_1 คือ ข้อมูลตัวอย่างทั้งสองมีการกระจายตัวที่แตกต่างกัน

แต่ในการนำมาใช้ในงานของ outlier detection นั้นเราไม่จำเป็นต้องทดสอบสมมติฐาน เพียงแต่ เรานำค่าของผลคะแนนมากน้อยที่ได้นั้น ไปประยุกต์ใช้ เพื่อวัดค่าเป็นระดับคะแนนที่แสดงความแปลกแยกแตกต่างของข้อมูลที่เราสนใจเมื่อเปรียบเทียบกับข้อมูลตัวอื่นๆในชุดข้อมูล วิธีการนำไปประยุกต์ใช้และการเปรียบเทียบจะนำเสนอในส่วนถัดไป

3.2.5 Kolmogorov-Smirnov and Efron Outlier Detection (KSE-test)

Kolmogorov-Smirnov and Efron Outlier Detection (KSE-Test) [12] เป็นการประยุกต์ใช้ Kolmogorov-Smirnov two sample test มาใช้ในการตรวจสอบข้อมูลแปลกแยก ซึ่งจะประยุกต์ใช้ ค่าเฉลี่ยของ KS-score มาอธิบายความน่าจะเป็นที่ข้อมูลแต่ละตัวมีโอกาสจะเป็นข้อมูลแปลกแยก

ให้ชุดข้อมูล S มีจำนวนข้อมูล n ตัว และ ในแต่ละแถวข้อมูลมีจำนวน m คุณลักษณะ (attributes) เพื่อที่จะหา Outlier score ของข้อมูลตัว ที่เราทำการสังเกต โดยคำนวณได้จากค่าของ KSE-score ของข้อมูลแต่ละตัวใน dataset เพื่อนำคะแนนที่ได้มาชี้วัดระดับความแปลกแยกของข้อมูลที่สนใจ โดย ค่าคะแนนที่มีค่ามากแสดงถึงความมีโอกาสที่ข้อมูลนั้นจะเป็นข้อมูลแปลกแยกมากขึ้นตามไปด้วย

สำหรับแต่ละจุดข้อมูล j ใน S จะมี ECDF $F_{pj}(x)$ คือ empirical cumulative distribution function ที่อธิบายการกระจายตัวของ Euclidean distance จากจุด j ไปยังจุดอื่นๆในชุดข้อมูล S ที่เราจะใช้การคิดคำนวณ KS-score จากวิธี Two Sample Kolmogorov Smirnov test ดังแสดงสมการที่ (4)

$$KS(p_j, p_i) = MAX|Fp_j(x) - Fp_i(x)| \quad (4)$$

เราใช้ค่าเฉลี่ยของ KS-score เพื่อแทนค่า KSE-score ดังแสดงในสมการที่ (5)

$$KSE(p_j) = \frac{1}{n-1}\sum_{\substack{i=1 \\ i\neq j}}^{n} KS(p_j - p_i) \quad (5)$$

เราจะจำแนกข้อมูลแปลกแยก ได้โดยใช้ค่าเฉลี่ยของ KS-score เทียบกับจุดอื่นๆ ทั้งหมดในชุดข้อมูล S เมื่อค่า KS-score ได้ออกมาจะอยู่ระหว่าง 0-1 ดังนั้น ค่า KSE-score ซึ่งเป็นค่าเฉลี่ยของ KS-score ก็จะอยู่ในช่วง 0-1 ด้วยเช่นกัน

แต่การคำนวณค่า KSE-score จากจุดหนึ่งไปยังทุกจุดบนข้อมูล เพื่อเปรียบเทียบกับจุดอื่นๆ ทั้งหมดในชุดข้อมูล S นั้นเป็นการสิ้นเปลืองเกินไป ซึ่งจะทำให้ประสิทธิภาพเชิงเวลาโตเป็น Exponential คือ $O(n^3)$ จึงใช้วิธีการสุ่มตัวแทนของข้อมูลแทนการใช้ข้อมูลทั้งหมด เพื่อลดจำนวนการประมวลผล โดยสุ่มข้อมูลกลุ่มตัวอย่าง sample1(S_1) และ กลุ่มตัวอย่าง sample2(S_2) ซึ่งมีขนาดเป็นค่าคงที่ C_1 และ C_2 ตามลำดับ โดยค่า C_1 และ C_2 เป็นค่าคงที่คือ ขนาดของกลุ่มตัวอย่าง ที่ ซึ่งเป็นค่าคงที่ใดก็ได้โดยที่ $C_1<n-1$ และ $C_2<n-1$ ทำให้ประสิทธิภาพเชิงเวลาเป็น $O(C_1C_2n)$ ทำงานได้ในเวลาเชิงเส้น ดังนั้นจึงได้ข้อสรุปการใช้งานว่าถ้าหากคำนวณบนชุดข้อมูลเล็กๆ สามารถใช้ทั้งชุดข้อมูลได้เลยถึงแม้ว่ามีประสิทธิภาพเป็น $O(n^3)$ แต่หากต้องการใช้ประมวลผลชุดข้อมูลใหญ่ๆจะใช้การสุ่มข้อมูลตัวอย่างมาใช้แทนเพื่อลดภาระการประมวลผลลง โดยที่ไม่ส่งผลกระทบต่อความแม่นยำของผลลัพธ์

3.2.6 A MapReduce Programming Model

MapReduce[13, 14] เป็นโมเดลการพัฒนาโปรแกรมสำหรับการประมวลผลแบบกระจาย เพื่อสามารถประมวลผลไฟล์งานขนาดใหญ่ ให้สามารถทำการประมวลผลได้แบบขนาน รูปแบบการพัฒนาโปรแกรมด้วยโมเดลนี้เป็นรูปแบบหลักที่ใช้งานกับระบบแบบกระจายที่มีขนาดใหญ่ ที่สามารถทำการประมวลผลแบบขนาน และ ทนต่อความผิดพร่อง เช่น Apache Hadoop[15]

โครงสร้างการทำงาน ประกอบด้วยการทำงานสองส่วนหลักๆ คือ ส่วนของการ Map คือ การประมวลผลข้อมูล และ ส่วนของการ Reduce คือ การรวมผลลัพธ์

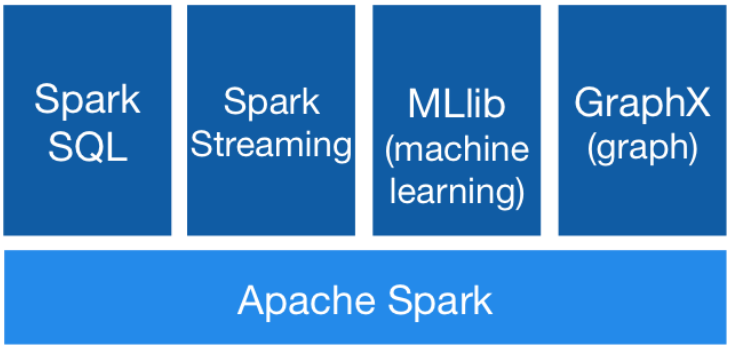
ซึ่งในปัจจุบัน มีการพัฒนา Library และ ส่วนต่อขยายต่างๆมากมาย ทำให้โปรแกรมมีการทำงานอยู่ในรูปแบบของ Map Reduce เพื่อรองรับภาระงานต่างๆบนระบบการประมวลผลแบบขนานกับข้อมูลขนาดใหญ่ได้ เช่น Library สำหรับการทำการเรียนรู้ของเครื่อง (Machine Learning)

3.2.7 Apache Spark

Apache Spark [16-18] เป็น platform สำหรับการประมวลผลข้อมูลบนหน่วยความจำแบบกระจาย ที่มีเป้าหมายสำหรับการเพิ่มความเร็วสำหรับการประมวลผลภาระงานแบบ batch ที่มีลักษณะของการทำซ้ำ ทำให้สามารถประมวลผลงานในรูปแบบ Map Reduce ได้เร็วขึ้น

Apache Spark ใช้ประโยชน์จากโครงสร้างข้อมูลแบบ Resilient Distributed Dataset (RDDs) ที่กระจายข้อมูลไปยังแต่ละเครื่อง(node) โดยที่ผู้ใช้สามารถเลือกเก็บส่วนสำคัญที่ใช้ในการประมวลผลไว้ในหน่วยความจำ เหมาะกับการคำนวณที่มีการวนรอบทำซ้ำ ซึ่งทำให้ Spark มีความสามารถในการประมวลผลแบบขนานโดยใช้ข้อมูลบนหน่วยความจำ และ ยังมีความสามารถในการทนต่อความผิดพร่อง(fault tolerant) ได้อีกด้วย การทำงานแบ่งออกเป็นสองส่วน คือ Transformation และ Action ซึ่งหลักการคล้ายคลึงกับการทำงานของ MapReduce Programing Model

ทั้งนี้ Apache Spark ยังสามารถรองรับภาษาระดับสูงได้ อาทิ Java, Scala, Python และ R ทั้งยังมีเครื่องมือเป็น Library สำหรับรองรับ ประเภทของงานที่หลากหลาย เช่น Spark SQL(สำหรับการทำฐานข้อมูล), MLlib Machine Learning Library(สำหรับงานด้านการเรียนรู้ของเครื่อง) และ GraphX(สำหรับการประมวลผลกราฟ) เป็นต้น นอกจากนี้ ยังมี Spark Streaming สำหรับรองรับการประมวลผลสายข้อมูล(streaming data)ในแบบขนานได้อีกด้วย ดังภาพที่ 4



ภาพที่ 4 ส่วนขยายของ Apache Spark[16]

3.2.8 A Confusion Matrix

วิธีการประเมินความถูกต้องแม่นยำของผลลัพธ์จากการทำการจัดกลุ่มข้อมูล เราจะนำ Confusion Matrix[19, 20] มาใช้ในการประเมินประสิทธิภาพของผลลัพธ์ของการจัดกลุ่มหรือการทำนายค่าที่มาจากกระบวนการ Machine Learning โดยการแสดงจำนวนของข้อมูลตาม Class ที่ได้ทำการทำนาย เปรียบเทียบกับ Class ที่แท้จริงของข้อมูล ลงในตาราง โดยแต่แถวของข้อมูลแบ่งข้อมูลตาม Class ที่ได้จากการทำนาย ส่วนสดมภ์(Column) ของตารางแบ่งข้อมูลตาม Class ที่แท้จริงของข้อมูล ค่าในแต่ละช่องของตารางแทนจำนวนของข้อมูลซึ่งใช้อธิบายความถูกต้องของผลลัพธ์การทำนายว่าสามารถทำนาย Class ของข้อมูลได้ถูกหรือผิดเป็นจำนวนเท่าใด

		Condition (as determined by "Gold standard")			
	Total population	Condition positive	Condition negative	Prevalence = Σ Condition positive / Σ Total population	
Test outcome	Test outcome positive	True positive	False positive (Type I error)	Positive predictive value (PPV, Precision) = Σ True positive / Σ Test outcome positive	False discovery rate (FDR) = Σ False positive / Σ Test outcome positive
	Test outcome negative	False negative (Type II error)	True negative	False omission rate (FOR) = Σ False negative / Σ Test outcome negative	Negative predictive value (NPV) = Σ True negative / Σ Test outcome negative
	Positive likelihood ratio (LR+) = TPR/FPR	True positive rate (TPR, Sensitivity, Recall) = Σ True positive / Σ Condition positive	False positive rate (FPR, Fall-out) = Σ False positive / Σ Condition negative	Accuracy (ACC) = Σ True positive + Σ True negative / Σ Total population	
	Negative likelihood ratio (LR−) = FNR/TNR	False negative rate (FNR) = Σ False negative / Σ Condition positive	True negative rate (TNR, Specificity, SPC) = Σ True negative / Σ Condition negative		
	Diagnostic odds ratio (DOR) = LR+/LR−				

ภาพที่ 5 Table of confusion and relationships among terms[19]

โดยตารางนี้สามารถย่นย่อให้เหลือขนาด 2x2 ได้ เรียกว่า Table of Confusion โดยในแต่ละช่องของตารางจะแสดงข้อมูลดังนี้ True Positive(TP), True Negative(TN), False Positive(FP) และ False Negative(FN) ดังภาพที่1 ซึ่งค่า True Positive และ True Negative จะแสดงจำนวนของข้อมูลที่สามารถทำนาย class ได้ถูกต้องตรงตามความเป็นจริง ในกรณีของการทำ Intrusion detection จะหมายความดังนี้ TP คือ จำนวนของข้อมูลที่เราจำแนกได้ถูกต้องว่าเป็นการจู่โจม และ TN คือ จำนวนของข้อมูลที่เป็นข้อมูลปกติ ซึ่งเราจำแนกได้ถูกต้องว่าเป็นข้อมูลปกติ ส่วนค่า False Positive และ False Negative เป็นจำนวนของการทำนายผลลัพธ์ที่ผิดไปจากความเป็นจริง ซึ่งก็คือกรณี False Positive คือ การจำแนกข้อมูลปกติว่าเป็นการบุกรุก และ False Negative คือ การจำแนกข้อมูลของการบุกรุกผิดว่าเป็นข้อมูลปกติ ซึ่งค่าทั้งสี่นี้สามารถนำมาคำนวณมาตรวัดความแม่นยำของผลลัพธ์ได้ตามสูตรคำนวณที่จะกล่าวดังต่อไปนี้

True positive rate (TPR หรือ sensitivity) หาค่าได้จากจำนวนของข้อมูลที่เป็นการบุกรุกระบบที่ถูกจำแนกได้อย่างถูกต้องหารด้วยจำนวนของการบุกรุกทั้งหมดในชุดข้อมูล สามารถคำนวณได้จากสูตรในสมการที่(6)

$$TPR = \frac{TP}{TP + FN} \tag{6}$$

True negative rate (TNR or specificity) หาค่าได้จากจำนวนของข้อมูลที่เป็นการใช้งานปกติที่ถูกจำแนกได้อย่างถูกต้องหารด้วยจำนวนของข้อมูลการใช้งานปกติทั้งหมดในชุดข้อมูล สามารถคำนวณได้จากสูตรในสมการที่(7)

$$TNR = \frac{TN}{TN + FP} \tag{7}$$

False positive rate (FPR หรือ false alarm rate) หาค่าได้จากจำนวนของข้อมูลที่เป็นการใช้งานปกติแต่ถูกจำแนกเป็นการบุกรุกหารด้วยจำนวนของข้อมูลการใช้งานปกติทั้งหมดในชุดข้อมูล สามารถคำนวณได้จากสูตรในสมการที่(8)

$$FPR = \frac{FP}{TN + FP} \tag{8}$$

False negative rate (FNR) หาค่าได้จากจำนวนของข้อมูลที่เป็นการบุกรุกแต่ถูกจำแนกเป็นการใช้งานปกติหารด้วยจำนวนของข้อมูลที่เป็นการบุกรุกระบบทั้งหมดในชุดข้อมูล สามารถคำนวณได้จากสูตรในสมการที่(9)

$$FNR = \frac{FN}{TP + FN} \tag{9}$$

Positive predictive value (PPV หรือ detection rate) จำนวนของข้อมูลการบุกรุกที่ตรวจจำแนกได้ถูกต้องจากจำนวนข้อมูลผลลัพธ์ทั้งหมดที่ถูกทำนายว่าเป็นการบุกรุกสามารถคำนวณได้จากสูตรในสมการที่(10)

$$PPV = \frac{TP}{TP + FP} \tag{10}$$

Accuracy หาได้จาก จำนวนของผลลัพธ์ที่ทำนายได้ถูกต้องทั้งหมด หารด้วยจำนวนข้อมูลทั้งหมดในชุดข้อมูลสามารถคำนวณได้จากสูตรในสมการที่(11)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

3.3 งานวิจัยที่เกี่ยวข้อง

3.3.1 Robust, Scalable Anomaly Detection for Large Collections of Images

งานวิจัยนี้ [12] ได้นำเสนอวิธีการ Kolmogorov-Smirnov and Efron Outlier Detection algorithm หรือ เรียกว่า KSE-test สำหรับใช้ทำการทำ Outlier detection ในรูปภาพ ซึ่งเป็นการนำเอาความรู้ด้านสถิติมาใช้ในการทดสอบความแปลกแยกของข้อมูล โดยใช้ Kolmogorov-Smirnov Two Sample Test ในการเปรียบเทียบ ซึ่งคะแนนที่ได้บ่งบอกถึงความน่าจะเป็นของข้อมูลที่จะเป็นข้อมูลแปลกแยก ยิ่งค่ามากยิ่งบ่งชี้ว่าข้อมูลนั้นมีความเบี่ยงเบนจากข้อมูลอื่นๆในชุดข้อมูลมาก ทำให้มีโอกาสสูงที่จะเป็นข้อมูลแปลกแยก หรือ สิ่งแปลกปลอม นอกเหนือจากนั้น

ข้อดีของวิธีการนี้คือ เป็นวิธีการแบบไม่มีการชี้นำ ไม่จำเป็นต้องใช้ชุดข้อมูลตัวอย่างในการสอน และ สามารถคำนวณคะแนนความแปลกแยกให้กับข้อมูลได้ในเวลาเชิงเส้น ทำให้มีการทำงานที่รวดเร็ว ผลการทดลองจากงานวิจัยนี้ สามารถใช้จำแนกภาพที่ไม่เหมือนภาพอื่นออกมาได้อย่างถูกต้อง

เราจึงเลือกนำวิธีการนี้มาประยุกต์เพื่อใช้กับการตรวจจำแนกข้อมูลการบุกรุกระบบจาก Log File ขนาดใหญ่ เพราะ มีประสิทธิภาพที่ดี และ สามารถต่อยอดโดยทำให้สามารถประมวลผลแบบขนานได้

3.3.2 Pruning Based Method for Outlier Detection

งานวิจัยนี้ [21] นำเสนอวิธีการทำ Clustering pruning ที่มีการค่อยๆตัดข้อมูลที่ไม่มีผลต่อการจำแนกข้อมูลแปลกแยกออก ในที่นี้คือ คัดแยกส่วนที่เป็นข้อมูลปกติที่ไม่เกี่ยวข้องกับการคำนวณออกไป เพื่อลดภาระในการประมวลผลลง และ เพิ่มความรวดเร็วในเชิงเวลา ซึ่งการทำงานประกอบด้วยสองขั้นตอนหลัก ได้แก่

ขั้นตอนแรก คือ การทำ Cluster pruning คือการจัดกลุ่มข้อมูล แล้วการตัดกลุ่มข้อมูลที่ไม่น่าจะเป็นข้อมูลแปลกแยกออก โดยการจัดกลุ่มเลือกใช้ K-Means algorithm ซึ่งมีความง่ายและรวดเร็วในการประมวลผล เมื่อได้กลุ่มข้อมูลออกมาแล้วทำการตัดกลุ่มขนาดใหญ่ที่มีขนาด(จำนวนสมาชิก) มากกว่าค่าเฉลี่ยของขนาดกลุ่มทั้งหมด หาได้จาก การใช้จำนวนข้อมูลทั้งหมด นำมาหารด้วยจำนวนกลุ่ม(ค่า K ที่เลือกใช้) ด้วยสมมติฐานที่ว่า ธรรมชาติของข้อมูล กลุ่มข้อมูลที่มีขนาดใหญ่ มักเป็นกลุ่มข้อมูลหลักในชุดข้อมูล(คือ การใช้งานปกติ) จึงทำการตัดออก ส่วนข้อมูลที่อยู่ในกลุ่มข้อมูลขนาดเล็ก มีโอกาสที่จะเป็นข้อมูลแปลกแยก ซึ่งจะนำไปคำนวณต่อในขั้นตอนต่อไป

ส่วนขั้นตอนที่สองจะทำการคำนวณ Outlier factor โดยการนำส่วนที่ไม่โดนตัดทิ้งจากขั้นตอนแรก มาคำนวณคะแนนความแปลกแยก(Outlier score) โดยใช้วิธีการ LDOF (Local Distance-based Outlier Factor) จากนั้นเรียงลำดับคะแนน แล้วเลือกค่าที่มากสุด n อันดับ แล้วจำแนกเป็นข้อมูลแปลกแยก ซึ่งทำให้ผลการจำแนกมีคุณภาพความถูกต้องแม่นยำคงเดิมหรือดีขึ้นกว่าเดิม และ ใช้เวลาในการประมวลผลน้อยลง สามารถสรุปขั้นตอนการทำงานได้ดังแสดงในภาพที่ 6

Algorithm 1: Outlier Detection Algorithm

Input: N : Number of data points,DS : Dataset k : required number of clusters, i : number of iterations, n number of outliers, .

Begin
- Set $Y \leftarrow kmeans(k, i, DS)$
- Set $Avgval \leftarrow \frac{N}{k}$
- **For** each cluster $C_i \in Y$ **do**
  - **If** $|C_i| > Avgval$ **then**
    - prune(C_i)
    - **Else**
      - Add C_i to UC
- **For** each point $p_i \in UC$ **do**
  - calculate $ldof(p_i)$
- Sort the points according to their $ldof(p_i)$ values.
- First n points with highest $ldof(p_i)$ values are the desired outliers.

End

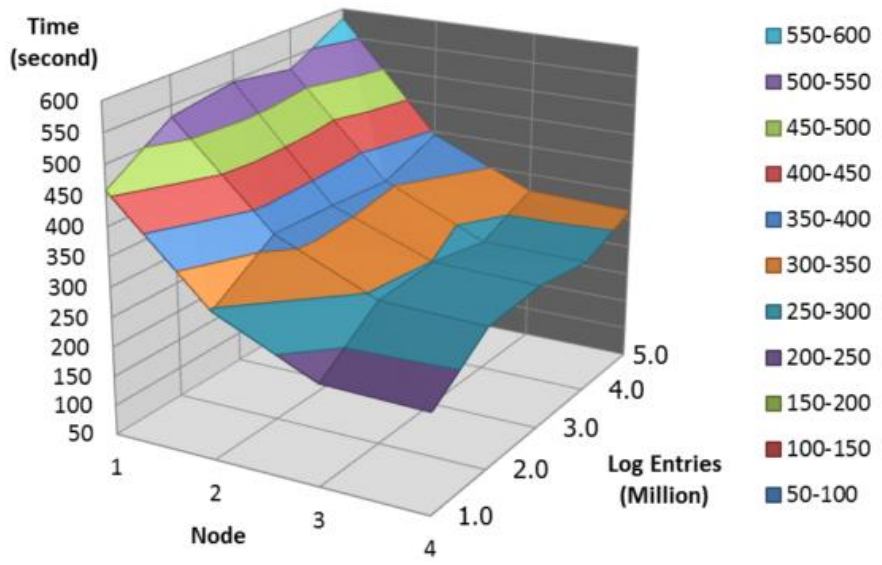
ภาพที่ 6 ขั้นตอนการทำงานของวิธีการ Cluster pruning[21]

แนวคิดที่น่าสนใจจากงานวิจัยนี้คือ การตัดข้อมูลที่ไม่น่าจะเป็นข้อมูลแปลกแยกออก เพื่อเพิ่มประสิทธิภาพและความแม่นยำของผลลัพธ์ มีการประยุกต์ใช้ วิธีการที่ง่ายและเร็วเพื่อการคัดกรองหรือตัดข้อมูลบางส่วนที่ไม่จำเป็นออก ทำให้จำนวนงานน้อยลง ในขณะที่ยังคงคุณภาพใกล้เคียงหรือดีขึ้นกว่าวิธีการเดิม โดยจากการทดลองแสดงให้เห็นว่า วิธีการนี้สามารถตัดทอนจำนวนจุดในขั้นตอนแรกได้เป็นสัดส่วนถึง 70-80 เปอร์เซ็นต์ของข้อมูลทั้งหมด โดยการทำงานได้คุณภาพผลลัพธ์คงเดิม หรือ ดีขึ้นอีกด้วย

แต่ข้อด้อยของวิธีการนี้ คือ ถึงแม้จะมีการตัดทอนข้อมูลบางส่วนออกเพื่อให้มีการคำนวณที่เร็วขึ้น แต่วิธีการที่ใช้ในการคำนวณคะแนน LDOF ยังมีความซับซ้อนทำให้ประสิทธิภาพเชิงเวลาที่ใช้ยังคงเป็น $O(n^2)$ ซึ่งอีกเป็นความยากและท้าทายในการหาวิธีการ Outlier detection ที่มีประสิทธิภาพสูงและใช้เวลาในการประมวลผลน้อย

3.3.3 Applying Hadoop for log analysis toward distributed IDS

จากงานงานวิจัย [22] เสนอการนำ Hadoop Cluster เข้ามาช่วยในการแก้ปัญหาด้านประสิทธิภาพ และ ข้อจำกัดด้านทรัพยากรของระบบในการประมวลผลงานจัดกลุ่มข้อมูล ซึ่งผลการทดลองได้แสดงในส่วนของประสิทธิภาพที่ได้จากการใช้ Hadoop Cluster ในการประมวลผลข้อมูล log ขนาดใหญ่ ให้ผลดีกว่าการทำการจัดกลุ่มข้อมูล บนคอมพิวเตอร์เครื่องเดียวทั้งในด้านประสิทธิภาพ และ ขนาดของภาระงานที่รองรับได้ ซึ่งการทำการจัดกลุ่มข้อมูลนี้สามารถทำได้โดยไม่จำเป็นต้องมีความรู้เกี่ยวกับตัวข้อมูลมาก่อน โดยการทดลองนี้ ได้นำ Hadoop มาใช้ในการประมวลผล Log file ขนาดใหญ่ โดยใช้ K-Means algorithm แบบขนาน ผ่านทาง Apache Mahout[23] ซึ่งเป็น Library ในการทำการประมวลผลงานในด้าน Machine Learning ที่เป็น MapReduce Model



ภาพที่ 7 แสดงความสัมพันธ์ของเวลาการทำงาน กับ ขนาดภาระงาน และ จำนวนเครื่องที่ใช้

การทดสอบประสิทธิภาพในด้านเวลาที่ใช้ในการประมวลผล โดยการแปรผันจำนวนเครื่อง (Node) ที่ใช้ ในการประมวลผล จากผลการทดลองพบว่า เมื่อทำการประมวลผลแบบขนาน โดยการเพิ่มจำนวนของ Node ที่ใช้ในการประมวลผล จะทำให้เวลาที่ใช้ในการประมวลผลลดลง ตามจำนวนของ Node ที่เพิ่มเข้าไป แสดงให้เห็นว่า การประมวลผลแบบขนาน เป็นทางเลือกที่เหมาะสมในการรองรับภาระงานขนาดใหญ่ และสามารถก้าวข้ามข้อจำกัดของการประมวลผลงานขนาดใหญ่โดยใช้เครื่องคอมพิวเตอร์เพียงเครื่องเดียว

3.3.4 An analysis of suitable parameters for efficiently applying K-means clustering to large TCPdump dataset using Hadoop framework

จากงานวิจัย [24] ได้เสนอการใช้ Hadoop Cluster ในการทำ K-mean Clustering กับข้อมูล TCPdump KDD'99 เพื่อหาค่า K ที่เหมาะสมกับชุดข้อมูล โดยอาศัยจุดเด่นคือพลังของ Hadoop Cluster ทำให้สามารถประมวลผลในแบบขนาน แต่ยังมีข้อจำกัดอยู่ที่ตัววิธีการที่เลือกใช้ คือ K-mean algorithm ที่เป็นการจัดกลุ่มแบบแบ่งส่วน (partitional clustering technique) โดยอาศัยการคำนวณระยะห่างจากจุดที่สนใจไปยัง จุดศูนย์กลางของแต่ละกลุ่ม ทำให้กลุ่มที่จัดได้เป็นลักษณะทรงกลมจากจุดศูนย์กลางที่เป็นค่าเฉลี่ยของกลุ่ม และ ทุกสมาชิกของเซตข้อมูลที่นำมาพิจารณาจะต้องถูกจัดเข้าไปยังกลุ่มที่ใกล้ที่สุด กล่าวคือ ข้อมูลทุกตัวต้องมีกลุ่ม ทำให้ความแม่นยำถูกจำกัดลงด้วยข้อจำกัดของวิธีการจัดกลุ่มข้อมูลที่ใช้ ผลลัพธ์ที่ได้จึงยังมีค่า False Positive Rate ค่อนข้างมาก ดังผลการทดลองที่แสดงในภาพที่ 8

Detection rate and False alarm rate at K25 of all data set

Data set (K25)	Detection Rate (PPV)	False Alarm Rate (FPR)
10 percentage	0.65055	0.34075
10per.Samp.01	0.76358	0.19336
10per.Samp.02	0.86278	0.09993
10per.Samp.03	0.87390	0.09014
10per.Samp.04	0.86476	0.09601
10per.Samp.05	0.87439	0.09174
Full Data Set	0.73099	0.00023
Full.Samp.01	0.68902	0.00034
Full.Samp.02	0.52715	0.28298
Full.Samp.03	0.52602	0.28021
Full.Samp.04	0.52602	0.28021
Full.Samp.05	0.51017	0.29592

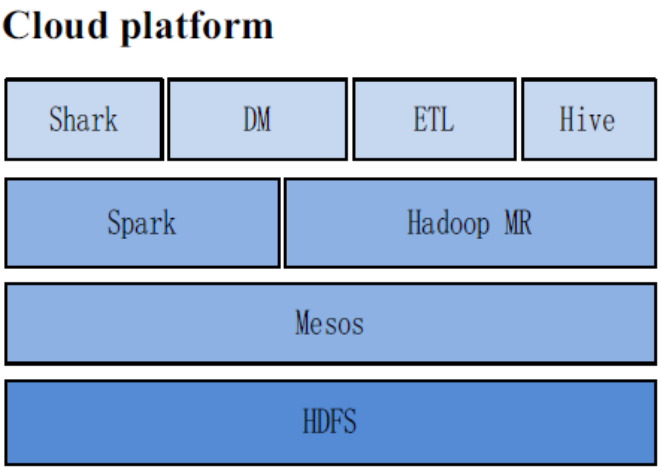
ภาพที่ 8 ผลการทดลองการจัดจำแนกการบุกรุกด้วย K-Means algorithm ที่ K=25

3.3.5 Log Analysis in Cloud Computing Environment with Hadoop and Spark

เนื่องจากในปัจจุบันการประมวลผลภาระงานวิเคราะห์ข้อมูลการใช้งานในปูม (log file) ขนาดใหญ่ เพื่อวิเคราะห์พฤติกรรมผู้ใช้งาน มีความสำคัญมากขึ้น โดยข้อมูลที่เก็บบันทึกมีขนาดใหญ่ขึ้นเรื่อยๆ จนเกินขีดจำกัดของการประมวลผลข้อมูลบนเครื่องเดียวได้ ข้อจำกัดในด้านขนาดของข้อมูลจึงกลายเป็นปัญหาที่มีความท้าทาย การนำ platform ในการประมวลผลแบบกระจาย คือ Hadoop จึงถูกหยิบยกมาใช้ในการประมวลผล และ จัดการกับข้อมูลขนาดใหญ่ในด้านความจุและการขยายขนาด แต่ก็ยังไม่สามารถตอบโจทย์สำหรับการทำงานวิเคราะห์ข้อมูลที่เน้นการประมวลผลได้ดีเท่าที่ควร ซึ่งโดยปกติภาระงานประเภทนี้มักจะมีการทำงานประมวลผลซ้ำเป็นรอบๆ จึงมีการนำ Apache Spark ที่เป็น platform การประมวลผลในหน่วยความจำแบบกระจายที่ถูกพัฒนาโดย UC

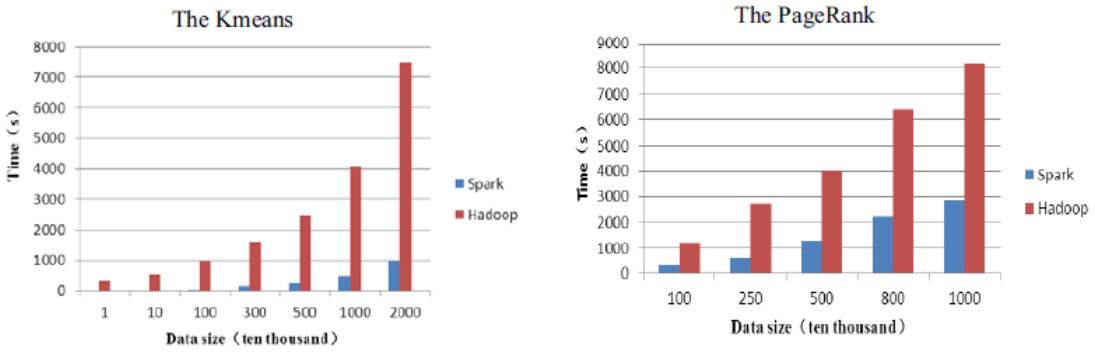
Berkeley AMP Lab เข้ามาใช้เพิ่มความเร็วในการประมวลผลงานวิเคราะห์ข้อมูลขนาดใหญ่ ที่เหมาะการทำงานที่มีการวนซ้ำมากกว่าการใช้ Hadoop MapReduce ปกติ

งานวิจัยนี้ [25] นำเสนอการทดสอบเปรียบเทียบระหว่าง Hadoop(MapReduce) กับ Spark ซึ่งเป็น platform ในการประมวลผลข้อมูลขนาดใหญ่แบบขนาน โดยใช้การประมวลผลการวิเคราะห์ปูมขนาดใหญ่ เพื่อเปรียบเทียบข้อแตกต่างระหว่างสอง platform โดย ทำการประมวลผลภาระงาน คือ K-Means algorithm และ PageRank algorithm เพื่อเปรียบเทียบความแตกต่างในด้านเวลาที่ใช้ในการประมวลผล โดย platform ที่ใช้ในการทดสอบดังแสดงในภาพที่ 9



ภาพที่ 9 platform ที่ใช้ในการทดสอบ[25]

โดยการประมวลผล K-Means algorithm และ PageRank algorithm ดำเนินการทดลองบน cluster ขนาด 6 node โดยที่แต่ละ node ในการประมวลผล ประกอบด้วย หน่วยประมวลผล 2 cores และ หน่วยความจำขนาด 4GB โดย มีการแปรผันขนาดของชุดข้อมูลที่ใช้ในการประมวลผล K-Means ที่ขนาด 10,000-20,000,000 แถว และ ขนาดของชุดข้อมูลที่ใช้ในการประมวลผล PageRank algorithm ที่ขนาด 10,000-10,000,000 แถวข้อมูล แล้ววัดเวลาที่ใช้ในแต่ละชุดการทดลอง ได้ผลการทดลองดังแสดงในภาพที่ 10



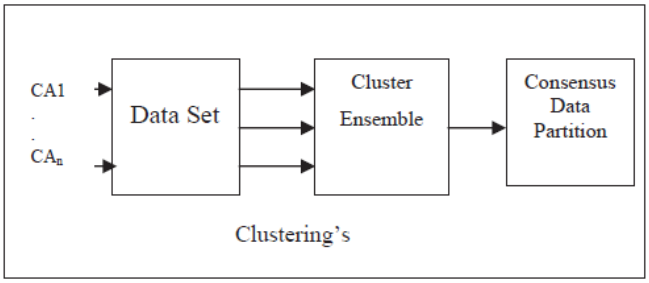
ภาพที่ 10 เวลาที่ใช้ประมวลผลระหว่าง Hadoop เปรียบเทียบ Spark ด้วย K-Means และ PageRank algorithm

จากผลการทดลอง จะพบว่าการประมวลผลบน platform Apache Spark ใช้เวลาในการประมวลผลน้อยกว่า Hadoop อย่างเห็นได้ชัด ในการประมวลผลงานวิเคราะห์ข้อมูลที่มีการวนซ้ำจากภาระงานที่ทดสอบทั้งสองชนิด เนื่องจากข้อได้เปรียบของ Spark ที่สามารถทำการประมวลผลข้อมูลบนหน่วยความจำ ที่เหนือกว่าการทำงานแบบ MapReduce ปกติ ทำให้ Spark มีความเหมาะสมสำหรับการนำมาใช้ในงานประมวลผลวิเคราะห์ข้อมูลที่มีลักษณะของการวนซ้ำ อย่างเช่นงานในด้าน Machine Learning เป็นต้น

3.3.6 Improving the Quality of Clustering Using Cluster Ensembles

งานวิจัยนี้[26]นำเสนอ แนวคิดการใช้ Clustering ensemble ในการปรับปรุงเสถียรภาพ ความแม่นยำ และ ความสมบูรณ์ของผลลัพธ์จากการทำ Unsupervised clustering โดยการรวมผลลัพธ์ที่ได้จากการทำ algorithm หลายๆชุดบน dataset เดียวกันซึ่งแต่ละชุดจะได้ผลลัพธ์ที่แตกต่างกันมาตีความเพื่อให้ได้บทสรุปสุดท้ายเป็นชุดผลลัพธ์เดียว เนื่องจากการใช้หลายๆ Clustering algorithm ที่แตกต่างกัน หรือ แม้แต่การใช้ Clustering algorithm เดียวกัน แต่ใช้ค่าของตัวแปรที่แตกต่างกันจะทำให้ผลลัพธ์ที่ได้ในแต่ชุดมีคุณสมบัติเฉพาะตัวที่แตกต่างกัน

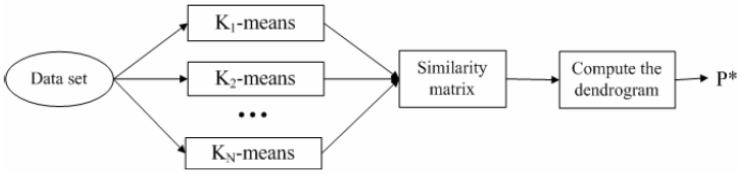
ในงานนี้ได้กล่าวถึง โครงสร้างการออกแบบ Clustering ensemble ประกอบด้วยสองส่วนหลักๆ ในส่วนแรกคือ ขั้นตอนการเก็บรวบรวมผลลัพธ์ที่ได้จาก Clustering algorithm แต่ละชุด และ ขั้นตอนที่สอง คือการใช้ Consensus function ในการรวมผลลัพธ์เพื่อหา Final partitions ดังแสดงในภาพที่ 11



ภาพที่ 11 Cluster ensemble framework[26]

3.3.7 An Intrusion Detection System Based on the Clustering Ensemble

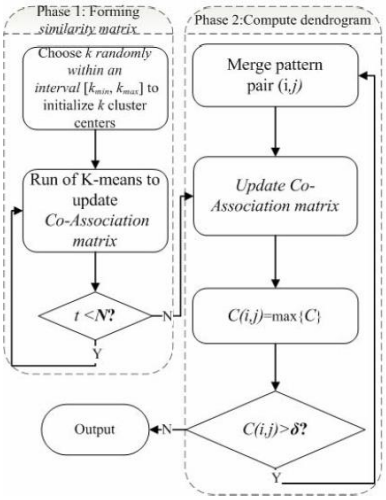
งานวิจัยนี้[7]นำเสนอแนวคิดของการทำ Clustering Ensemble คือ การนำผลลัพธ์ลัพธ์จากการจัดกลุ่มๆจากการประมวลผลหลายๆชุดเพื่อลดความผิดพลาดคลาดเคลื่อนจากการดำเนินการประมวลผลเพียงชุดเดียว โดยใช้วิธีการ K-means Clustering Ensemble คือการนำ K-means algorithm มาทำการจัดกลุ่มข้อมูลโดยการแปรผันค่า Parameter โดยแต่ละชุดการประมวลผลจะมีการสุ่มค่าจำนวนกลุ่มผลลัพธ์(K) ที่แตกต่างกันไปในแต่ละชุด นำผลลัพธ์มาสร้างเป็น Co-Association Matrix แล้วใช้ Single-link hierarchical clustering ในการสรุปผลลัพธ์จากทุกชุดเพื่อให้ได้ผลลัพธ์สุดท้ายเป็นกลุ่มข้อมูลที่จำแนกประเภทตามลักษณะข้อมูลที่มีความคล้ายคลึงกัน โดยนำเสนอวิธีการนี้ในชื่อ Evidence Accumulation (EA) ดังแสดงในภาพที่ 12



ภาพที่ 12 The Diagram of EA algorithm[7]

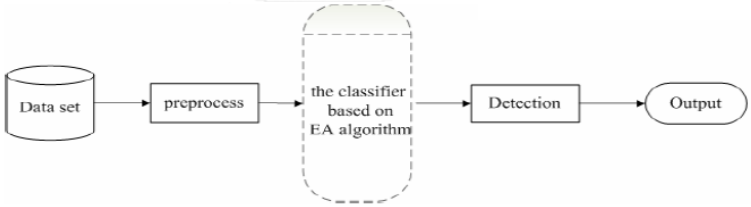
ขั้นตอนหลักๆแบ่งออกเป็นสองส่วน คือ ส่วนแรก Compute Algorithm (การประมวลผล Clustering algorithm) เป็นการสร้าง และ Update similarity matrix จากการประมวลผล K-means แต่ละชุด ส่วนที่สอง Consensus data partitions (ส่วนของการรวมผลลัพธ์) เป็นการรวม

ผลลัพธ์โดยการใช้ Single-link technique เพื่อสร้าง Dendogram สำหรับหาผลลัพธ์สุดท้าย แผนผังแสดงขั้นตอนการทำงานของ EA algorithm โดยคร่าวเป็นดังภาพที่ 13



ภาพที่ 13 The flowchart of EA algorithm[7]

ซึ่งได้ผลลัพธ์จากการทำ Clustering Ensemble ด้วย EA algorithm เป็นกลุ่มข้อมูลที่สามารถเอาไปใช้ใน IDS ที่เรียกว่า EAIDS ดังภาพที่ 14



ภาพที่ 14 The Diagram of EAIDS[7]

วิธีการทำ Ensemble เป็นอีกหนึ่งแนวทางที่ดีในการนำมาใช้ในการลดความผิดพลาดคลาดเคลื่อน แต่เนื่องจาก การนำมาใช้จำเป็นต้องมีวิธีการรวมผลลัพธ์ที่ดีด้วย จึงจะไม่เกิดปัญหาในแง่ของเวลาในการประมวลผลที่จะตามมา และ อีกทั้งการประมวลผลข้อมูลหลายๆชุด หลายๆครั้ง จะเป็นต้องคำนึงถึงความคุ้มค่าในการทำงานที่เพิ่มขึ้น ขั้นตอนมากขึ้น เวลาที่ใช้มากขึ้น และ ใช้ทรัพยากรของระบบมากขึ้น เหล่านี้มีความคุ้มค่าต่อคุณภาพงานที่ทำและสามารถตอบสนองความต้องการได้อย่างเหมาะสมหรือไม่

3.3.8 A Novel Approach for Outlier Detection and Clustering Improvement

งานวิจัยนี้[4]นำเสนอแนวคิดการทำ Outlier Detection โดยการนำ Clustering algorithm มาปรับปรุงเพื่อให้มีความสามารถในการนำมาใช้เป็น Anomaly detector โดยได้นำเสนอ ODC algorithm (Modified K-means algorithm) ที่เกิดจากการดัดแปลง K-Means algorithm เพื่อให้มีความสามารถในการจำแนก External Outliers (ลักษณะของ Outlier ที่มีการรวมตัวกันกลุ่มเล็กๆ อยู่ห่างไกลจากกลุ่มอื่นๆมากๆ) และ Internal Outlier (ลักษณะของ Outlier ที่มีความแปลกแยกห่างออกมาจากข้อมูลอื่นๆในกลุ่มเดียวกันมากๆ) ซึ่ง ODC Algorithm มีหลักการทำงานดังภาพที่ 15

Algorithm ODC

Input: D(A_1,A_2,....,A_n),k,p // The dataset, no. of cluster and threshold
Output: Clustered Data, Outliers and SSE/SST.
Begin
1. Choose a value of k.
2. Select k objects randomly and use them as initial set of centroids
3. Calculate the distances between k centroids and all the objects in dataset **D**
4. Calculate the mean distances (***Md***) between k centroids and all the objects in dataset **D**
5. Assign each object to the cluster for which it is nearest centroid and calculate SSE/SST.
6. ***for*** each object x in dataset **D**
7. If distance (x,c_k)> p* (***Md***)
8. Consider x as an outlier and remove from dataset **D** and calculate SSE/SST.
9. ***end***
10. Recalculate the centroids.
11. Repeat steps 3-10 until objects stop changing clusters.
End

ภาพที่ 15 Algorithm ODC[4]

ซึ่งวิธีการนี้สามารถให้ผลลัพธ์ในส่วนของประสิทธิภาพเชิงเวลาได้ดี เป็น O(n) เนื่องจากพัฒนาต่อยอดมาจากวิธีการที่ง่าย แต่ยังมีส่วนของการคำนวณและจำแนก Outlier ในการทำงานทุกรอบของการจัดกลุ่มใน Modified K-means ซึ่งการพยายามจำแนกข้อมูลแปลกแยก ออกมาจากชุดข้อมูล โดยการเปรียบเทียบระยะระหว่างตำแหน่งของจุดข้อมูลที่สนใจ กับ จุดศูนย์กลางของกลุ่ม ตั้งแต่ขั้นตอนการจัดกลุ่มยังไม่เสร็จสมบูรณ์ ซึ่งในขั้นตอนจะมีการคำนวณค่าของตำแหน่งจุดศูนย์กลางของกลุ่มค่าใหม่ที่ทำให้มีการเคลื่อนตำแหน่งศูนย์กลางของกลุ่มอยู่เรื่อยๆ และในขณะเดียวกันสมาชิกยังมีการถ่ายเทระหว่างแต่ละกลุ่มในช่วงแรกๆของการจัดกลุ่ม ทำให้มีโอกาสในการจำแนกผิดพลาดสูง เพราะ ทั้งตำแหน่งของกลุ่ม สมาชิกในกลุ่ม และ ระยะห่างของสมาชิกกับจุดศูนย์กลางมีการเปลี่ยนแปลงตลอดเวลา ยังไม่คงที่ในขณะรอบแรกๆของการขยับตำแหน่งจุดศูนย์กลางของกลุ่ม ซึ่งการดำเนินการในขั้นตอนนี้อาจทำให้มีโอกาสความผิดพลาดในการจำแนกสูง และ การดำเนินการส่วนนี้ยังเป็นการประมวลผลที่เกินความจำเป็น และ ด้วยข้อจำกัด ของ K-Means algorithm ทำให้ผลลัพธ์ของการจำแนกความถูกต้องแม่นยำยังไม่ดีเท่าที่ควร

3.3.9 An Intrusion Detection Method Based on Outlier Ensemble Detection

งานวิจัยชิ้นนี้[27]ได้นำเสนอ VoteOut algorithm ซึ่งนำเอาวิธีการ ensemble มาใช้กับการทำ Outlier mining สำหรับการทำ Outlier detection การใช้ Clustering algorithm เพื่อแก้ปัญหาจากการที่ Clustering algorithm ซึ่งไม่ได้ถูกออกแบบมาเพื่อการทำ Anomaly detection โดยตรง ทำให้การใช้ Clustering algorithm ในตรวจหา Outlier ทำได้ไม่ดีเท่าที่ควร เพราะทำให้เกิด false alarm มาก โดยในขั้นตอนการทำ ensemble จะใช้ Incidence matrix และการคิดคำนวณค่า Similarity coefficient ของแต่ละคู่ instance บันทึกลงใน matrix แล้วนำมาคิดค่าผลรวม Similar coefficient sum ของ instance แต่ละตัวจากผลรวมของแต่ละแถว(row)ใน matrix น่าค่าที่ได้มาจำแนกโดยเปรียบเทียบกับ Threshold ที่กำหนด เพื่อแบ่งว่าแต่ละคู่มีความสัมพันธ์กันมากพอที่จะจัดอยู่ในกลุ่มเดียวกันหรือไม่

แต่จุดที่ทำให้ประสิทธิภาพในการทำงานของ VoteOut algorithm ยังไม่ดีเท่าที่ควรคือ ส่วนที่มีการใช้ Incidence matrix(Co-association Matrix) ขนาด NxN ทำให้ต้องใช้พลังในการประมวลผลเยอะ และ ยังต้องใช้ memory มากในการรองรับการประมวลผล ประสิทธิภาพเชิงเวลาเป็น $O(n^2)$ ทำให้ไม่เหมาะกับการนำไปใช้ประมวลผลชุดข้อมูลที่มีขนาดใหญ่มีจำนวนสมาชิกในข้อมูลเป็นจำนวนมากๆ จะทำให้เวลาและทรัพยากรที่ต้องใช้ในการประมวลผลเติบโตขึ้นอย่างรวดเร็ว

3.3.10 A New Semi-unsupervised Intrusion Detection Method Based on Improved DBSCAN

งานวิจัยนี้[28]ได้นำเสนอแง่คิดว่าในปัจจุบัน วิธี unsupervised ที่มีอยู่โดยส่วนมากนั้นมีจุดอ่อนหลักๆ ส่วนแรกคือเป็นการยากในการจัดการกับข้อมูลที่เป็น categorical หรือ บางวิธีต้องใช้เทคนิคและขั้นตอนที่ซับซ้อน ส่วนที่สองกรรมวิธีเหล่านั้น sensitive ต่อการกำหนดค่า parameter ที่เหมาะสม ซึ่งเป็นการยากในทางปฏิบัติจริงที่จะกำหนดค่าที่เหมาะสมกับชุดข้อมูล และ ส่วนสุดท้ายคือ การที่สรุปว่า cluster ที่มีขนาดเล็กทั้งหมดเป็น anomaly เป็นวิธีการที่ไม่ค่อยสมเหตุสมผลนัก

DBSCAN (Density-Based Spatial Clustering of Application with Noise) มีข้อดีที่เด่นชัดอยู่สองอย่างคือ ไม่ sensitive ต่อลำดับของข้อมูลที่เป็น input และ สามารถจัดกลุ่มข้อมูลเป็นรูปร่างใดๆก็ได้ตามความหนาแน่น และ สามารถจำแนกข้อมูลที่เป็น noise จากการจัดกลุ่มออกมาได้ จึงเลือกวิธีการนี้มาดัดแปลงให้มีความเหมาะสมมากขึ้น

งานวิจัยนี้นำเสนอวิธีการ IIDBG ซึ่งเป็น semi-unsupervised เพื่อการทำ Intrusion detection โดยการ ปรับปรุงประสิทธิภาพของ DBSCAN algorithm ที่เป็น density based

clustering ที่มีความสามารถในการจำแนกข้อมูล Outlier ออกมาโดยใช้ noise ที่ได้จากการจัดกลุ่ม โดยได้นำมาปรับปรุงเพิ่มเติมในส่วนของการ merge small cluster ในกรณีที่อยู่ใกล้กับ normal cluster พิจารณาจาก uniform คือ สมาชิกในกลุ่มการกระจายตัวสม่ำเสมอ หรือ not uniform สมาชิกกระจายตัวไปทางฝั่งใดฝั่งหนึ่งมากกว่าอีกด้าน ถ้าสมาชิกส่วนใหญ่ในกลุ่มเอนเอียงอยู่ใกล้กับกลุ่มข้างเคียงใดที่สุด ให้รวมเข้ากับกลุ่มนั้น เพื่อสร้างผลลัพธ์ที่ให้มีความสามารถในการจำแนกมากขึ้น และ มีความผิดพลาดน้อยลง แต่เนื่องจากเป็นวิธีการที่ดัดแปลงมาจากการจัดกลุ่ม แบบ DBSCAN ซึ่งมีประสิทธิภาพเชิงเวลาเป็น $O(n^2)$ ทำให้ประสิทธิภาพเชิงเวลาของวิธีการ IIDBG มีการเติบโตเป็น $O(n^2)$ ตามไปด้วย เมื่อขนาดของข้อมูลที่ประมวลผลโตขึ้น จะทำให้เวลาและทรัพยากรที่ใช้ เติบโตขึ้นอย่างรวดเร็ว เมื่อถึงระดับหนึ่งที่ข้อมูลมีขนาดใหญ่เกินไป จะทำให้แม้แต่การขยายระบบและทำการประมวลผลแบบขนานไม่สามารถเติบโตได้ทันขนาดของภาระงานที่เพิ่มขึ้น

3.3.11 สรุปงานวิจัยที่เกี่ยวข้อง

จากงานวิจัยที่ได้สำรวจมาข้างต้นจะพบว่า การประมวลผลจำแนกข้อมูลการบุกรุกระบบใน Log file ขนาดใหญ่นั้นมีวิธีการที่นำมาประยุกต์ใช้หลากหลายวิธี ซึ่งวิธีการที่ให้ความแม่นยำสูงมักมีความซับซ้อนในขั้นตอนการประมวลผล และ มีประสิทธิภาพเชิงเวลาที่เติบโตอย่างรวดเร็วเมื่อขนาดของข้อมูลเพิ่มขึ้นเวลาที่ใช้กลับเพิ่มขึ้นในอัตราที่เป็น Exponential กับขนาดข้อมูล แต่เมื่อแก้ปัญหาลดความซับซ้อนในการประมวลผล โดยการเลือกใช้วิธีการที่ง่าย ใช้เวลาในการประมวลผลน้อย (ประสิทธิภาพเชิงเวลาเป็นเชิงเส้น) และ ใช้ทรัพยากรน้อย มาปรับปรุงดัดแปลง ให้สามารถใช้สำหรับการตรวจจับการบุกรุกระบบได้นั้น กลับให้ความถูกต้องแม่นยำของผลลัพธ์ต่ำ คือ จำแนกการบุกรุกได้น้อย และ อัตราการผิดพลาดสูง อันเนื่องมาจากข้อจำกัดของวิธีการที่เลือกใช้

การประมวลผลแบบกระจาย เป็นอีกวิธีหนึ่ง ที่ถูกนำมาใช้ในการแก้ปัญหาข้อจำกัดในด้านของเวลา และ ทรัพยากรที่ใช้ในการประมวลผล ข้อดีคือการใช้วิธีประมวลผลแบบขนานช่วยให้สามารถลดเวลาที่ใช้ในการประมวลผลภาระงานลงได้จากการเพิ่มหน่วยประมวลผล และ ทำให้สามารถขยายระบบเพื่อรองรับการเติบโตของข้อมูลได้ แต่ถึงกระนั้น ถ้าขนาดของภาระงาน โตขึ้นเร็วกว่าความสามารถในการขยายระบบ ดังนั้นวิธีการที่เลือกใช้ในการประมวลผลจึงยังมีความสำคัญ ในแง่ของการจำกัดประสิทธิภาพเชิงเวลาให้อยู่ในขอบเขตที่ระบบสามารถรองรับได้

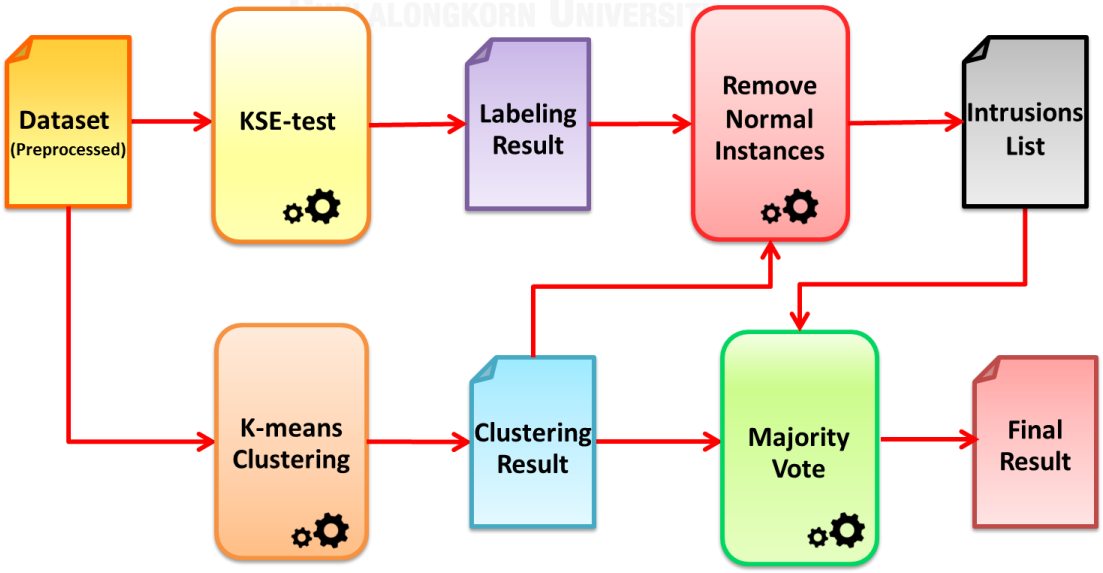
บทที่ 4
ขั้นตอนการดำเนินงาน

4.1 การออกแบบงานวิจัย

ในส่วนนี้เราจะนำเสนอแนวคิดและวิธีการดำเนินงาน แบ่งออกเป็น 7 ขั้นตอนหลักๆ ดังนี้

1. Data Collection
2. Data preprocessing
3. Calculating KSE-score and Labeling Outlier Instances
4. K-means Clustering and Creating Normal Profile
5. Enhancing Accuracy
6. Labeling Clusters
7. Calculate DR and FPR

โดยส่วนของวิธีการที่นำเสนอ ประกอบด้วยขั้นตอนหลัก 4 ขั้นตอน คือ ส่วนของขั้นตอนที่ 3 ถึง 6 ตามที่ได้นำเสนอไว้ด้านบน เนื่องจากการนำ KSE-test มาใช้ในการทำการตรวจจำแนกข้อมูลการบุกรุกระบบนั้นยังให้ผลลัพธ์ของการจำแนกไม่แม่นยำเท่าที่ควร เราจึงจำเป็นต้องมีวิธีการเพิ่มเติมเพื่อเพิ่มประสิทธิภาพของผลลัพธ์การจัดจำแนกที่ได้ ให้เหมาะสมกับประเภทของงาน โดยวิธีการที่นำเสนอ ในการปรับปรุงประสิทธิภาพผลลัพธ์มีลำดับการทำงานดังภาพที่ 16



ภาพที่ 16 แผนผังการเดินทางของข้อมูล และ ผลลัพธ์ที่ได้จากแต่ละขั้นตอน

4.2 วิธีการดำเนินงาน

ในส่วนนี้เราจะนำเสนอแนวคิดและวิธีการดำเนินงาน แบ่งออกเป็น 7 ขั้นตอนหลักๆตามที่ได้กล่าวไว้แล้วในหัวข้อที่ 4.1 โดยมีรายละเอียดของวิธีการในแต่ละลำดับขั้นตอนดังต่อนี้

4.2.1 การจัดเก็บข้อมูล (Data Collection)

ในการทดลองนี้ เราจะใช้ข้อมูล TCPdump log คือ KDD'99 [29] ซึ่งเป็นข้อมูล Network intrusion ที่ใช้ในการแข่งขัน KDD Cup 1999: Computer Network Intrusion Detection Competition ที่ทำการจัดเก็บข้อมูลโดย MIT Lincoln Labs มาใช้ในการทดลอง ซึ่งข้อมูลดังกล่าวประกอบด้วยจำนวนข้อมูลประมาณ 5 ล้านชุด (records) แต่ละชุดมีตัวแปรคุณลักษณะ 41 attributes ประกอบไปด้วยข้อมูลที่เป็นข้อมูลพฤติกรรมการใช้งานปกติ และ ข้อมูลพฤติกรรมการบุกรุกระบบที่แบ่งได้ออกเป็น 4 ประเภทหลักๆ คือ DoS, U2L, L2R และ Probing แต่ในการทดลองนี้จะใช้ข้อมูลตั้งต้นเป็นชุดข้อมูล 10% ของ Dataset ที่ทางผู้จัดเตรียมข้อมูลได้มีไว้ให้

เนื่องจากด้วยสัดส่วนของข้อมูล KDD'99 ในข้อมูลต้นฉบับนั้นมีจำนวนข้อมูลการบุกรุกระบบมากกว่า จำนวนข้อมูลการใช้งานปกติ ซึ่งขัดกับสมมติฐานของการทำ Anomaly Detection ที่ใช้คือจำนวนของข้อมูลปกติจะต้องมีจำนวนมากกว่าจำนวนข้อมูลการโจมตีเป็นจำนวนมากๆ จึงต้องทำการสร้างชุดข้อมูลขึ้นมาใหม่จากชุดข้อมูลเดิมด้วยวิธีการสุ่ม โดยชุดข้อมูลใหม่ที่มีสัดส่วนข้อมูลตามที่กำหนด ในที่นี้เราจะใช้สัดส่วน ข้อมูลปกติมากกว่า 90 เปอร์เซ็นต์ และ ข้อมูลที่เป็นการโจมตีไม่เกิน 10 เปอร์เซ็นต์ เพื่อให้สอดคล้องกับลักษณะของข้อมูลที่พบในการใช้งานแอพพลิเคชั่นจริง ทำการสุ่มจนกว่าจะได้ข้อมูลครบตามจำนวนที่กำหนด

4.2.2 การเตรียมข้อมูล (Data Preprocessing)

ในขั้นตอนนี้เราจะทำการเตรียมข้อมูลจากชุดข้อมูลที่ได้จากขั้นตอนแรก(data collection) ให้มีลักษณะที่เหมาะสมต่อการนำไปประมวลผลการตรวจจับข้อมูลแปลกแยก และ การจัดกลุ่มข้อมูล โดยขั้นตอนแรก เราจะทำการแปลงค่าของข้อมูลในแต่ละ attribute ของแต่ละ record ให้อยู่ในรูปตัวเลข เพื่อให้เหมาะกับการนำไปหาค่าระยะห่างด้วยวิธี Euclidean distance ในขั้นตอนของการจัดกลุ่ม หลังจากนั้น แล้วนำไปเข้าสู่กระบวนการ Normalize ให้ค่าของข้อมูลในแต่ละ attribute ให้อยู่ในช่วงค่าที่กำหนด คือ เราจะใช้ค่าระหว่าง 0-1000 เริ่มต้นจากการแปลงให้เป็นค่า 0-1 แล้วคูณด้วย 1000 เพื่อให้เหมาะสำหรับการนำไปคิดระยะห่างในขั้นตอนต่อไป และทำให้ทุก attribute จะมีผลต่อระยะห่างที่คำนวณในขั้นตอนของการจัดกลุ่มเท่าๆกัน

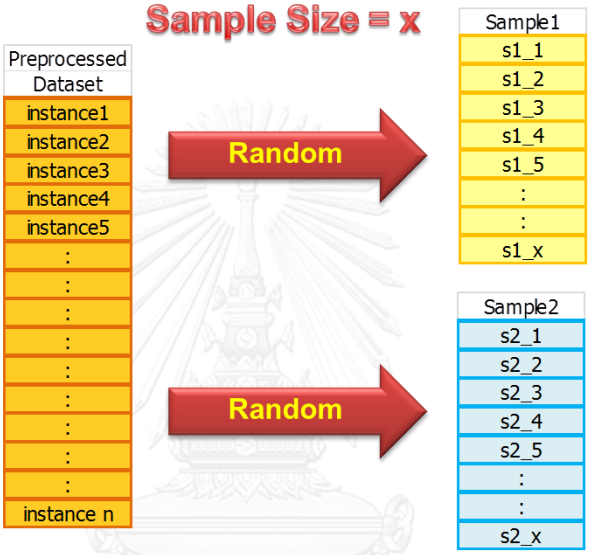
4.2.3 Calculating KSE-score and Labeling Outlier Instances

ในขั้นตอนแรกเราจะนำชุดข้อมูลที่ผ่านขั้นตอนการ preprocess เรียบร้อยแล้ว มาทำการคำนวณหาคะแนนความแปลกแยก(Outlier score) ให้กับข้อมูลแต่ละตัวในชุดข้อมูล S โดยวิธีการ Kolmogorov-Smirnov and Efron Outlier Detection[12] เพื่อให้ได้ KSE-score ของทุกข้อมูลในชุดข้อมูล โดยได้มีการปรับปรุงแก้ไขลำดับการทำงาน และ การสุ่มข้อมูล ของ KSE-test algorithm ที่มีอยู่เดิม เพื่อเพิ่มประสิทธิภาพให้ทำงานได้รวดเร็วขึ้น และ ลดการทำงานซ้ำซ้อนซึ่งเป็นการประมวลผลส่วนที่เกินความจำเป็นลง โดยได้ทำการตัด ขั้นตอนของการสุ่มข้อมูลใหม่ และ คำนวณระยะทาง Euclidean distance ที่ต้องทำใหม่ทั้งหมดในทุกรอบการการประมวลผล ออกไปโดยเปลี่ยนมาใช้วิธีการคำนวณ Euclidean distance เก็บไว้ใน Distance Matrix ครั้งเดียวสำหรับเรียกข้อมูลไปใช้ได้เลยทันทีโดยไม่ต้องคำนวณใหม่ เป็นการลดต้นทุนของการคำนวณคะแนนของข้อมูลแต่ละตัวในชุดข้อมูลได้ สรุปขั้นตอนการทำงานหลังจากปรับปรุง KSE-test ได้แสดงไว้เป็นรหัสเทียม (pseudo code) ในภาพที่ 17

```
function KSE_Test (dataset[],sample_size){

    sample1[] = create1DArray(sample_size)
    sample2[] = create1DArray(sample_size)
    distance_matrix[][] = create2DArray(sample_size,sample_size)
    kse_score[] = create1DArray(dataset.length)

    sample1 = randomData(dataset[],sample_size) //random sample1
    sample2 = randomData(dataset[],sample_size) //random sample2

    //Calculate distance from all points in sample1 to all points in sample2
    for(x=0 -> sample1.length){
        for(y=0 ->sample2.length){
            distance_matrix[x][y] = EuclideanDistance(instance_s1[x],instance_s2[y])
        }
    }

    //Calculate average KS-score for each instance
    for(x=0 -> dataset.length){
        observed_instance = dataset[x]
        distance_array[] = create1DArray(sample_size)
        for(y=0 ->sample2.length){
            distance_array[y] = EuclideanDistance(observed_instance,instance_s2[y])
        }
        sum = 0
        for(z=0 ->sample1.length){
            ks_score = KStest(distance_array,distance_matrix[z])
            sum += ks_score
        }
        average = sum/sample1.length
        kse_score[x] = average
    }
    return kse_score
}
```

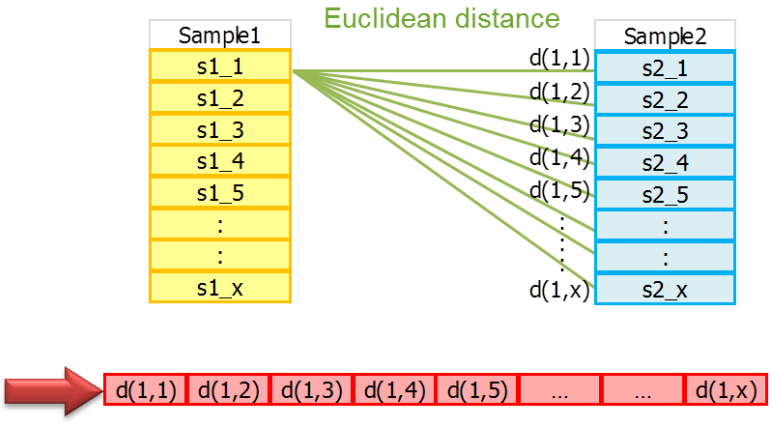
ภาพที่ 17 ลำดับขั้นตอนการทำงานของ KSE-test ที่มีการปรับปรุงแก้ไขให้เหมาะกับงานวิจัยนี้

จากขั้นตอนการทำงานใน pseudo code สามารถอธิบายขั้นตอนการทำงานดังนี้ ขั้นตอนเตรียมข้อมูลก่อนประมวลผลสุ่มชุดข้อมูล Sample1 และ Sample2 เพื่อเป็นชุดข้อมูลตัวแทนของชุดข้อมูลปกติ (เนื่องจากสมมติฐานหลักของการจำแนกข้อมูลแปลกแยก คุณลักษณะสำคัญของชุดข้อมูลที่นำมาประมวลผลด้วยวิธีการแบบไม่มีการชี้นำ คือ มีจำนวนข้อมูลปกติมากกว่าข้อมูลแปลกแยกมากๆ ดังนั้นการสุ่มข้อมูลจากชุดข้อมูลที่มีข้อมูลปกติเป็นข้อมูลส่วนใหญ่ ย่อมได้ชุดข้อมูลทีมีจำนวนของข้อมูลปกติเป็นกลุ่มหลักในชุดข้อมูล sample ทั้งสองชุดเช่นกัน) ทำการสุ่มข้อมูลจาก dataset ให้ได้ครบตามจำนวน Sample Size ที่กำหนด ดังแสดงในภาพที่ 18



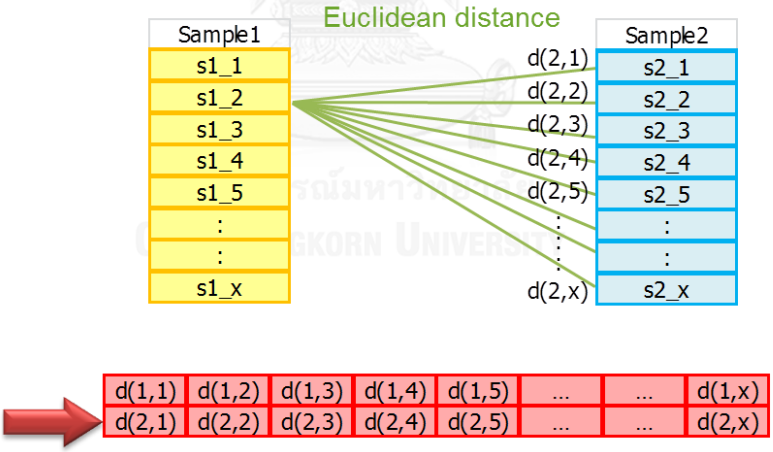
ภาพที่ 18 ทำการสุ่มข้อมูล Sample1 และ Sample2 จากชุดข้อมูล

จากนั้นเข้าสู่ขั้นตอนการประมวลผล ซึ่งแบ่งได้เป็นเป็นสองส่วนหลักๆ การทำงานส่วนแรก คือ การคำนวณ Distance Matrix เก็บระยะห่าง Euclidean distance ระหว่างข้อมูลแต่ละตัวใน Sample1 ไปยัง ข้อมูลทุกตัวใน Sample2 ซึ่งขนาดเท่ากับ sample1_size x และ sample2_size ตามลำดับ (เบื้องต้นในการทดลองนี้จะใช้ขนาด sample1_size = sample2_size แล้วสุ่มเลือกข้อมูลจนครบ) ค่าของระยะห่างที่คำนวณเก็บไว้ใช้ในการอ้างอิง และ เปรียบเทียบในขั้นตอนถัดไป ซึ่งสามารถแสดงขั้นตอนได้ดังภาพที่ 19 ถึง ภาพที่ 21



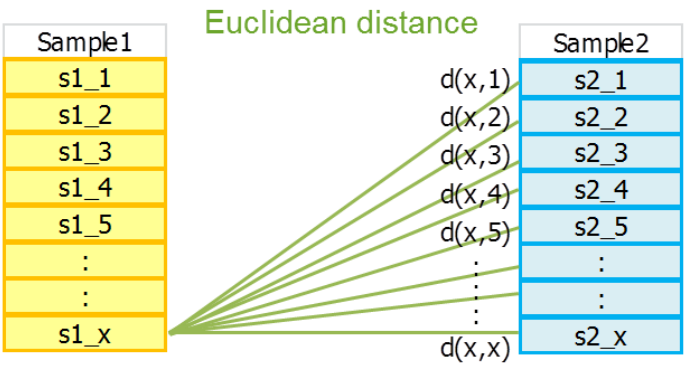
ภาพที่ 19 การคำนวณ Euclidean distance เพื่อสร้าง Distance Matrix

การคำนวณ Distance Matrix เก็บ Euclidean distance ระหว่าง sample1 และ sample2 เก็บลงในแถวของข้อมูล ดังแสดงในภาพที่ 19 เป็นการคำนวณชุดของระยะห่างจากข้อมูลตัวแรกของ Sample1 ไปยัง ข้อมูลทุกจุดใน Sample2 จะทำให้ได้ค่าของชุดระยะห่างเก็บลงใน แถวแรกของ Distance Matrix



ภาพที่ 20 คำนวณ Euclidean distance เก็บลงในแถวถัดไปของ Distance Matrix

จากภาพที่ 20 เป็นการคำนวณชุดของระยะห่างจากข้อมูลตัวถัดมาของ Sample1 ไปยังข้อมูลทุกจุดใน Sample2 จะทำให้ได้ค่าของชุดระยะห่างเก็บลงใน แถวต่อไปใน Distance Matrix ทำการคำนวณและบันทึกค่าลงในแต่ละแถวของ Distance Matrix ทำซ้ำไปทีละค่าจนครบทุกคู่ของข้อมูลระหว่าง Sample1 และ Sample2

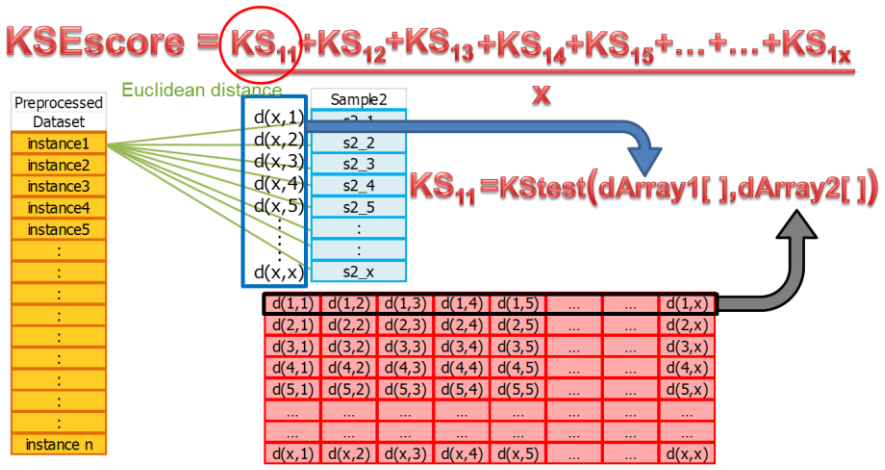


d(1,1)	d(1,2)	d(1,3)	d(1,4)	d(1,5)	...	...	d(1,x)
d(2,1)	d(2,2)	d(2,3)	d(2,4)	d(2,5)	...	...	d(2,x)
d(3,1)	d(3,2)	d(3,3)	d(3,4)	d(3,5)	...	...	d(3,x)
d(4,1)	d(4,2)	d(4,3)	d(4,4)	d(4,5)	...	...	d(4,x)
d(5,1)	d(5,2)	d(5,3)	d(5,4)	d(5,5)	...	...	d(5,x)
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
d(x,1)	d(x,2)	d(x,3)	d(x,4)	d(x,5)	...	...	d(x,x)

ภาพที่ 21 Distance Matrix หลังจากการคำนวณเสร็จสิ้น

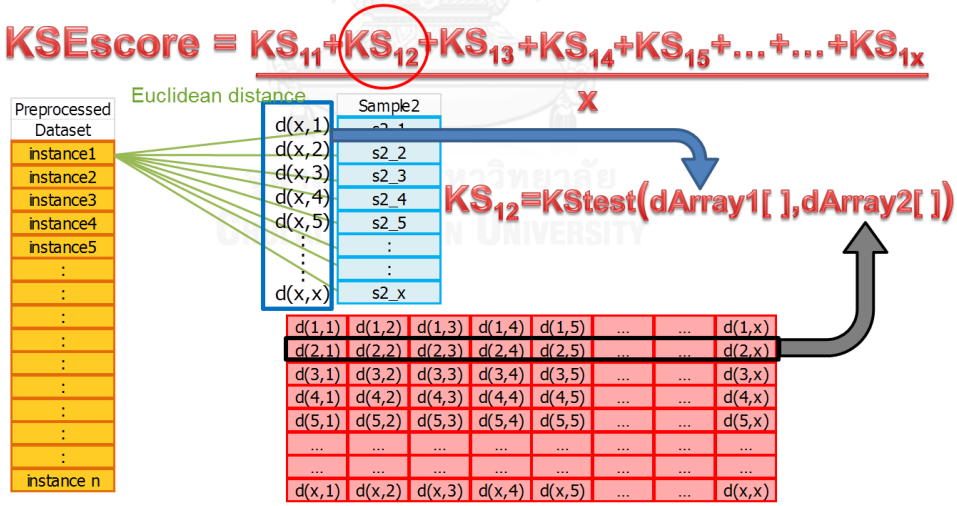
เมื่อการคำนวณเสร็จสิ้นจะได้ค่าของ Euclidean distance เก็บอยู่ใน Distance Matrix ขนาดเท่ากับ sample1_size x และ sample2_size จนเต็ม ดังภาพที่ 21

ขั้นตอนการประมวลผลหลักส่วนที่สองคือส่วนของการคำนวณ KSE-score ให้กับข้อมูลแต่ละตัวในชุดข้อมูล โดย เปรียบเทียบกับชุดของ Euclidean distance ของ Sample1 ที่ได้คำนวณเก็บไว้ใน Distance Matrix เพื่อคิดค่าเฉลี่ยของผลคะแนน KS-score ซึ่งเป็นค่าที่ได้จากการทดสอบ Kolmogorov-Smirnov test (KS-test) เปรียบเทียบการกระจายตัวระหว่าง ชุด Euclidean distance ของจุดที่เราสนใจ กับ ชุด Euclidean distance ของข้อมูลใน Sample1 สามารถอธิบายวิธีการสำหรับการคำนวณ KSE-score โดยละเอียดได้ดังภาพที่ 22 ถึง ภาพที่ 24



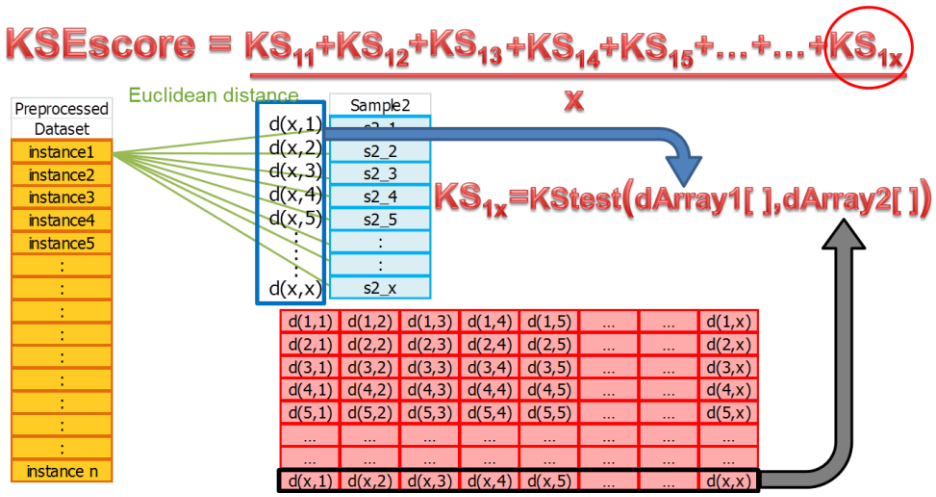
ภาพที่ 22 คำนวณ KS-score ระหว่าง จุดที่เราสนใจ เทียบกับข้อมูลตัวแรกของ Sample1

จากภาพที่ 22 การคำนวณ KS-score ระหว่าง โดยใช้การกระจายตัวของระยะทาง จุดที่เราสนใจ(ชุดของ Euclidean distance ระหว่างจุดข้อมูลที่สนใจกับทุกจุดบน Sample2) เทียบกับการกระจายตัวของชุดระยะทางจากข้อมูลตัวแรกของ Sample1 (ชุดของ Euclidean distance ระหว่างข้อมูลตัวแรกกับทุกจุดบน Sample2 ซึ่งเราได้ทำการคำนวณล่วงหน้าเก็บไว้ในแนวแรกของ Distance Matrix ไว้แล้ว) นำไปคิดคะแนนความแตกต่างโดยใช้การทดสอบด้วย KS-test



ภาพที่ 23 คำนวณ KS-score ระหว่าง จุดที่เราสนใจ เทียบกับข้อมูลของตัวที่สองใน Sample1

คำนวณ KS-score ระหว่าง จุดที่เราสนใจ เทียบกับข้อมูลตัวถัดไปของ Sample1 โดยใช้วิธีการเดียวกับครั้งแรก ทำการเปรียบเทียบทีละคู่ต่อไปเรื่อยๆ จนกว่าจะเปรียบเทียบกับ ข้อมูลใน Sample1 จนครบทุกตัว



ภาพที่ 24 การคิด KSE-score หลังจากคำนวณ KS-score ครบทุกคู่

จากภาพที่ 24 เมื่อคำนวณครบทุกคู่แล้วแล้ว จะสามารถคิด KSE-score ได้จากการหาค่าเฉลี่ยของ KS-score ระหว่างจุดที่เราสนใจเทียบกับทุกจุดใน Sample1 (ผลรวมของ KS-score หารด้วยขนาดของ Sample ที่ใช้) เนื่องจากค่าที่ได้จาก KS-test จะมีค่าอยู่ในช่วงระหว่าง 0-1 ดังนั้นค่าของ KSE-test ย่อมมีค่าอยู่ในระหว่าง 0-1 เช่นกัน

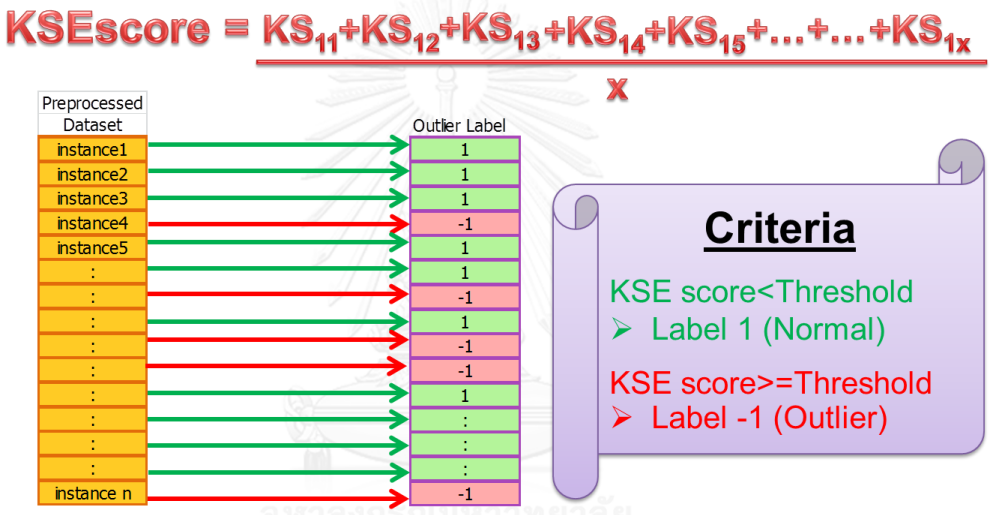
ทำการคำนวณ KSE-score ให้กับข้อมูลแต่ละตัวในชุดข้อมูล จนครบทั้งหมดทุกตัว ค่า KSE-score นี้ แสดงถึงความน่าจะเป็นที่ข้อมูลนั้นจะเป็นข้อมูลแปลกแยก หรือ เป็นข้อมูลการบุกรุกระบบนั่นเอง คะแนนยิ่งมาก ยิ่งมีโอกาสมากตามไปด้วย

เนื่องจาก KSE-test ดั้งเดิมที่ได้ถูกในเสนอไว้ใน [12] ถูกนำมาใช้ในงานเป็นการใช้สำหรับการตรวจจับข้อมูลที่แปลกแยกของรูปภาพ และ ยังไม่เคยถูกนำมาประยุกต์ใช้ในการใช้ตรวจสอบข้อมูลการบุกรุกระบบบนข้อมูลขนาดใหญ่ทำให้มีประเด็นที่แตกต่างกันออกมาพอสมควรทั้งในเรื่องของหลักเกณฑ์เงื่อนไขในการระบุว่าข้อมูลใดบ้างที่เป็น ข้อมูลปกติ และ ข้อมูลใดเป็นการบุกรุกระบบ เพราะเนื่องจากวิธีเดิมใช้การเลือก top-n outliers จากคะแนนมากที่สุด n อันดับ แต่เนื่องจากในการตรวจสอบข้อมูลการบุกรุกระบบแบบไม่มีการชี้นำ (Unsupervised Intrusion Detection) ซึ่งเราไม่สามารถระบุจำนวนของข้อมูลการบุกรุกที่แอบแฝงอยู่ในข้อมูลล่วงหน้าได้ จึงทำการเปลี่ยนมาใช้ค่า Threshold ในการแบ่งเพื่อให้ได้หลักเกณฑ์ในการจัดจำแนกที่ชัดเจนขึ้น กว่าการใช้วิธีการของ Top-n outliers ที่เราไม่สามารถระบุจำนวนที่ชัดเจนได้ การเลือกใช้วิธีกำหนดค่า Threshold ในการจำแนกทำให้สามารถทำงานได้โดยไม่จำเป็นต้องทราบจำนวนของข้อมูลบุกรุกมาก่อน

จากผลลัพธ์คะแนนที่ได้จากการคำนวณ KSE-score เราจะจำแนกข้อมูล โดยใช้การกำหนด Threshold เป็นค่าคงที่ค่าหนึ่งเพื่อใช้ในการแบ่งแยก ข้อมูลปกติ กับ การบุกรุกระบบ โดยมีหลักเกณฑ์การจำแนกข้อมูลดังนี้

- ข้อมูลที่ได้คะแนนมากกว่าหรือเท่ากับระดับ Threshold ที่กำหนดให้จำแนกเป็นการบุกรุกระบบ แทนด้วยค่า -1
- ข้อมูลที่มีคะแนนต่ำกว่าค่า Threshold ให้จัดจำแนกเป็นข้อมูลปกติ แทนด้วยค่า 1

หลังจากเสร็จสิ้นขั้นตอนนี้ จะได้รายการของข้อมูลที่มีความน่าจะเป็นสูงที่จะเป็นการบุกรุกระบบ ผลการจำแนกแสดงภาพที่ 25



ภาพที่ 25 ภาพแสดงวิธีการใช้ Threshold สร้างรายการจำแนกชนิดข้อมูลจากคะแนน KSE-score

เมื่อวิเคราะห์ผลจากการกำหนดระดับค่า Threshold การเลือกใช้ค่า Threshold ที่มาก จะทำให้มีความถูกต้องสูง ผิดพลาดน้อย แต่การใช้ค่า Threshold ที่มากเกินไป จะทำให้จำนวนข้อมูลการบุกรุกถูกจำแนกได้น้อยลงตามไปด้วย และ อาจจะยังไม่ครอบคลุมถึงจำนวนข้อมูลการบุกรุกทั้งหมดที่ยังซ่อนอยู่ในข้อมูล

แต่การเลือกใช้ค่า Threshold ที่ต่ำลงนั้น ทำให้สามารถตรวจจำแนกได้กว้างขวางและครอบคลุมมากขึ้น แต่ก็อาจทำให้ความแม่นยำในการจำแนกนั้นลดลงได้ เนื่องจากการเลือกข้อมูลที่มีความน่าจะเป็นการบุกรุกต่ำลง ทำให้ความแม่นยำถูกต้องนั้นต่ำลงด้วย

จากการทดลองนำ KSE-test ไปใช้จริงจะพบว่าการจำแนกผลลัพธ์ด้วย Threshold ไม่สามารถให้ประสิทธิภาพในการจำแนกข้อมูลการบุกรุกระบบ ได้ถูกต้องแม่นยำได้ตามที่ต้องการ เมื่อ

พยายามลดค่า Threshold ต่ำลงเพื่อให้ครอบคลุมจำนวนข้อมูลการบุกรุกในชุดข้อมูลได้มากขึ้น ความถูกต้องแม่นยำจะลดลงมาก และ การที่จะทำให้สามารถเลือกใช้ค่า Threshold ที่ต่ำลงจึงจำเป็นต้องใช้เทคนิควิธีอื่นเข้ามาช่วย และ วิธีการนั้นต้องไม่ทำให้ ระดับความแม่นยำในการจำแนกแย่ลงมากจนเกินไป จึงต้องมีการเพิ่มเติมขั้นตอนเพื่อปรับปรุงประสิทธิภาพของผลลัพธ์ ที่จะกล่าวรายละเอียดของวิธีการที่เพิ่มเติมขึ้นในขั้นตอนถัดไป

4.2.4 K-means Clustering and Creating Normal Profile

จากขั้นตอนแรก เนื่องจากผลลัพธ์ที่ได้นั้น จะได้รายการของข้อมูลที่มีความน่าจะเป็นสูงที่จะเป็นการบุกรุกระบบ แต่เนื่องจากการนำข้อมูลที่ได้นั้นมาใช้จำแนกชนิดของผลลัพธ์แล้วนำผลลัพธ์ไปใช้โดยทันทีนั้น จะให้ผลลัพธ์ที่ไม่แม่นยำเท่าที่ควร เพราะ ผลลัพธ์ที่ได้จากการเลือกใช้ระดับของ Threshold ที่ต่ำ จะทำให้มีการตรวจจับข้อมูลปกติปะปนมากับข้อมูลการบุกรุกในผลลัพธ์เป็นจำนวนมาก ทำให้ความถูกต้องแม่นยำของผลลัพธ์ลดลง ดังนั้นเราจึงมีความจำเป็นที่จะต้องกำจัดข้อมูลปกติที่อาจปะปนมากับผลลัพธ์การที่ได้มาจากขั้นตอนแรก (จาก KSE-test ข้อมูลการบุกรุก คือ ที่มี KSE-score มากกว่า Threshold ซึ่งเป็นกลุ่มมีความน่าจะเป็นสูงที่จะเป็นข้อมูลการแปลกแยกนั่นเอง) ซึ่งวิธีการที่เลือกมาใช้ในการกำจัดข้อมูลปกติที่ปะปนมาออกไป โดยการสร้าง Normal Profile จากผลลัพธ์ที่ได้จากกลุ่มข้อมูล จากการใช้ K-Means clustering algorithm ซึ่งมีความง่ายและทำงานได้รวดเร็วเข้ามาช่วยในการตรวจหาและจำแนกข้อมูลปกติ

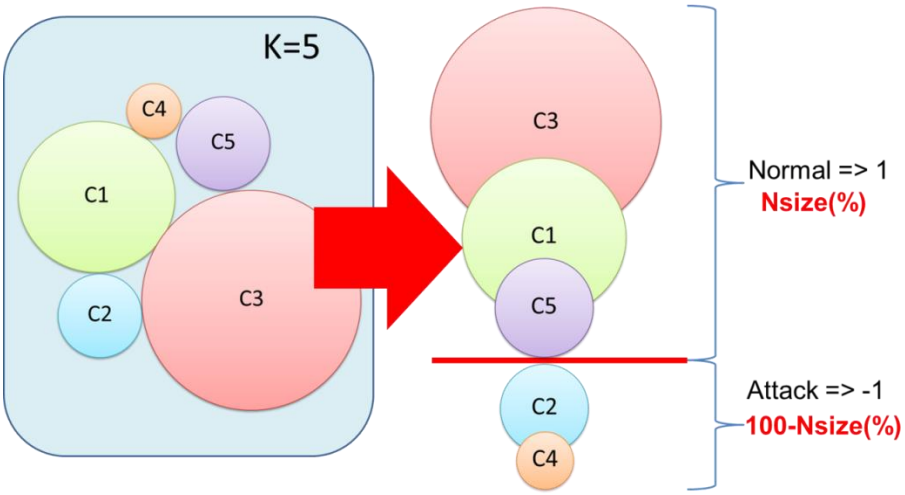
ถึงแม้การนำ clustering algorithm มาดัดแปลงเพื่อหาข้อมูลแปลกแยกโดยตรง จะมีการทดลองวิจัยมาบ้างแล้ว แต่การใช้งานยังไม่ให้ผลที่ดีเท่าที่ควร เนื่องจากขั้นตอนวิธีต่างๆที่นำมาใช้ในการจัดกลุ่มข้อมูล โดยส่วนมากถูกออกแบบมาให้ใช้ในงานหากลุ่มก้อนของข้อมูลมากกว่า(ใช้เพื่อหาความเหมือนกันของข้อมูลกลุ่มใหญ่มากกว่าใช้เพื่อหาความแตกต่าง) ดังนั้นเราจึงนำมาใช้ในการหากลุ่มก้อนของข้อมูลปกติซึ่งมีขนาดใหญ่ แทนที่จะนำมาดัดแปลงเพื่อใช้ในการตรวจจับหาข้อมูลแปลกแยกซึ่งเป็นข้อมูลส่วนน้อยโดยตรง

เนื่องจาก K-means clustering algorithm เป็นวิธีการจัดกลุ่มข้อมูลที่เหมือนกันเข้าไว้ด้วยกัน จึงไม่เหมาะกับการนำมาใช้ตรวจจับการบุกรุกโดยตรง แต่สามารถนำมาใช้ระบุกลุ่มก้อนของข้อมูลที่มีลักษณะคล้ายคลึงกันและทำการจำแนกเอาไว้ในกลุ่มเดียวกันได้ ซึ่งเป็นจุดแข็งของการใช้ clustering algorithm ทำให้กลุ่มของข้อมูลที่ได้มีความใกล้เคียงกันสูงและมักจะเป็นข้อมูลประเภทเดียวกัน เราสามารถนำข้อดีในจุดนี้มาใช้ในการระบุกลุ่มของข้อมูลที่มีโอกาสสูงที่จะเป็นข้อมูลปกติ

เพื่อนำมาสร้างเป็น Normal profile โดยใช้การจัดกลุ่มผลลัพธ์เรียงตามขนาดของกลุ่มที่ได้ ข้อมูลการใช้งานปกติซึ่งเป็นข้อมูลส่วนใหญ่ในชุดข้อมูล จะถูกจัดอยู่กลุ่มก้อนที่มีขนาดใหญ่

ซึ่งวิธีการนี้ยังสอดคล้องกับสมมติฐานขั้นต้นของการทำ Anomaly Detection ที่ถูกระบุไว้ว่า ข้อมูลที่จะนำมาใช้ในกระบวนการทำ Unsupervised Anomaly Detection คือ ข้อมูลการบุกรุกจะต้องมีจำนวนน้อยกว่าข้อมูลปกติมากๆ และ มีความแตกต่างกับข้อมูลปกติอย่างเห็นได้ชัด เราจึงแยกกลุ่มของข้อมูลปกติที่มีขนาดใหญ่ได้จากการสังเกตจำนวนของสมาชิกในกลุ่มผลลัพธ์ที่ได้ออกมาจากการจัดกลุ่มข้อมูล

โดยการ Label ชนิดของข้อมูลจากผลลัพธ์ที่ได้จากการจัดกลุ่ม จะใช้การพิจารณาผลลัพธ์จากการเรียงตามขนาดของ Cluster มีวิธีการ คือ นำข้อมูลที่ได้จากขั้นตอนการเตรียมข้อมูล มาทำการจัดกลุ่มด้วยวิธีการ K-means กลุ่มข้อมูลที่มีขนาดใหญ่ และมีจำนวนสมาชิกมากๆ (มีโอกาสที่จะเป็นกลุ่มของข้อมูลปกติมาก) เราจะนำมาใช้สร้าง Normal Profile หลังจากการเรียงตามขนาดกลุ่มแล้ว เราจะกำหนดค่าสัดส่วนของ จำนวน Normal ที่คาดหวัง แทนด้วยตัวแปร Nsize ว่ามีสัดส่วนเป็นเปอร์เซ็นต์เท่าใดของข้อมูลทั้งหมด (เบื้องต้นใช้การประมาณการสัดส่วน) เพื่อใช้ในการกำหนดจุดแบ่งระหว่างกลุ่มของข้อมูลปกติ กับ กลุ่มของข้อมูลการบุกรุก ดังภาพที่ภาพที่ 26 แล้วเลือกกลุ่มของข้อมูลที่ได้ตามขนาด โดยเรียงลำดับจากกลุ่มขนาดใหญ่ไปหาเล็ก (เลือกข้อมูลในกลุ่มที่มีขนาดใหญ่ที่สุดใส่ลงใน Normal Profile ก่อนตามด้วยกลุ่มขนาดเล็กกว่าในลำดับถัดลงมา) จนได้ขนาดของ Normal profile ครบตามขนาด Nsize ที่กำหนด (ประมาณเป็นจำนวนข้อมูลได้โดย Nsize*datasetSize/100 ถ้าจำนวนรวมเกินค่านี้แล้วให้เสร็จสิ้นการทำงาน) แล้วไปทำการจำแนกชนิดด้วยวิธีการดังภาพที่ 26



ภาพที่ 26 การแบ่งขอบเขตข้อมูลปกติและข้อมูลการบุกรุก โดยการกำหนดสัดส่วน Nsize

- ถ้าหากข้อมูลเป็นกลุ่มที่ถูกจัดอยู่ในกลุ่มข้อมูลขนาดใหญ่ที่เลือกมา หมายความว่ามีโอกาสเป็นข้อมูลปกติสูง ก็ให้ทำการ Label สมาชิกตัวนั้นเป็นชนิดข้อมูลปกติ แทนด้วยค่า 1
- กลุ่มที่เหลือที่ไม่ได้คัดเลือก จัดเป็นข้อมูลส่วนน้อยที่ยังไม่แน่นอนว่าเป็นการบุกรุกระบบหรืออาจเป็นข้อมูลปกติปะปนอยู่ก็ได้ ให้แทนด้วยค่า -1 ไว้

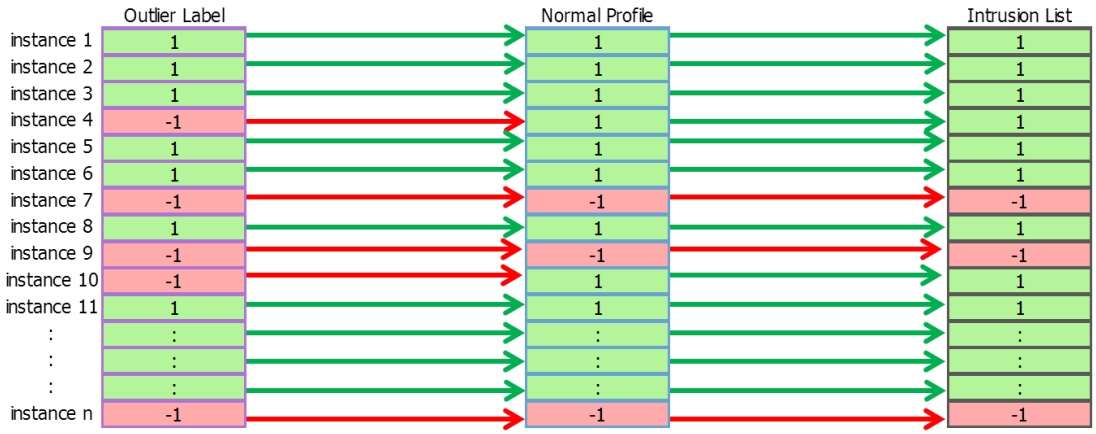
เมื่อทำการจำแนกข้อมูลครบตามจำนวนแล้วจึงสิ้นสุดขั้นตอนการสร้าง Normal Profile (โดยที่สัดส่วนที่ใช้ในการสร้าง Normal Profile นั้น ไม่จำเป็นต้องเท่ากับจำนวน สัดส่วนที่แท้จริง ในสัดส่วนที่มั่นใจได้ว่าครอบคลุมส่วนที่เป็นข้อมูลปกติโดยมีการประมาณส่วนเผื่อเอาไว้เพื่อไม่ให้ครอบคลุมไปถึงข้อมูลการบุกรุก เรามักจะเลือกใช้ค่าประมาณที่น้อยกว่าจำนวนของสัดส่วนข้อมูลปกติที่มีอยู่จริงในชุดข้อมูล เพื่อให้ Normal profile ที่ได้ มีความถูกต้องแม่นยำสูง)

ค่าสัดส่วนที่เลือกใช้จะอยู่ในช่วง 50 เปอร์เซ็นต์ขึ้นไป จนถึง จำนวนข้อมูลปกติที่มีอยู่จริงในชุดข้อมูล แต่ด้วยความที่ ข้อมูลที่เรานำมาใช้กับการประมวลผลแบบไมมีการชี้นำ เราจะไม่ทราบสัดส่วนที่แท้จริงของข้อมูลล่วงหน้า ก็สามารถเลือกใช้ค่าน้อยที่สุดเท่าที่เป็นไปได้ คือ 50 เปอร์เซ็นต์ (เนื่องจากตามสมมติฐานการตรวจจับข้อมูลแปลกแยกแบบไม่มีการชี้นำระบุไว้ว่า ชุดข้อมูลที่นำมาใช้จะต้องมีจำนวนของข้อมูลปกติ มากกว่าข้อมูลการบุกรุกมากๆ ดังนั้นอ้างอิงจากสมมติฐานที่ใช้ ในชุดข้อมูลหนึ่งที่นำมาใช้ประมวลผล จะต้องมีจำนวนของข้อมูลปกติเกินกึ่งหนึ่งของข้อมูลทั้งหมดในชุดข้อมูล) หรือ ประมาณการตามความเหมาะสม

นำข้อมูล Normal Profile ที่ได้ ไปใช้คัดกรองผลลัพธ์จากขั้นตอนแรก เพื่อเพิ่มความแม่นยำด้วยวิธีการในขั้นตอนถัดไป

4.2.5 Enhancing Accuracy

เมื่อได้ผลลัพธ์จากทั้งสองขั้นตอน คือ outlier list และ normal profile เราจะนำ normal profile ที่ได้ มาทำการกำจัดข้อมูลปกติ ที่เจือปนมาใน outlier list ที่ได้จากขั้นตอนแรก เพื่อปรับปรุงคุณภาพความแม่นยำของผลลัพธ์ เพื่อลดจำนวนของการจำแนกผิดพลาดให้น้อยลง ซึ่งทำให้คุณภาพของผลลัพธ์โดยรวมดีขึ้น โดยสามารถสรุปขั้นตอนการทำงานดังแสดงในภาพที่ 27



ภาพที่ 27 การใช้ Normal Profile เพิ่มความถูกต้องแม่นยำของผลลัพธ์

จากภาพ การสร้าง intrusion list เราจะกำหนด Label ของผลลัพธ์โดยเราจะแบ่งเป็นสองประเภท คือ ชนิดข้อมูลพฤติกรรมปกติ(Normal) และ ข้อมูลพฤติกรรมการจู่โจมระบบ(Attack) การทำนายชนิดของข้อมูลโดยมีวิธีดังนี้

- ข้อมูลที่เป็น Outliers จากขั้นตอนแรก และ ไม่ถูกจัดอยู่ในกลุ่ม normal ในขั้นตอนที่สองให้ทำนายเป็นชนิด Attack และแทนชนิดด้วยค่า -1
- ข้อมูลที่เป็น Outliers จากขั้นตอนแรก แต่ ถูกจัดอยู่ในกลุ่มเป็น normal ในขั้นตอนที่สองให้ทำนายเป็นชนิด Normal แทนชนิดด้วยค่า 1
- ข้อมูลในกรณีอื่นๆ ที่ไม่ถูกจัดอยู่ในกลุ่ม normal ในขั้นตอนที่สองให้จำแนกชนิดยึดตามผลลัพธ์จากขั้นตอนแรกเป็นหลัก

เมื่อเสร็จสิ้นกระบวนการ จะได้รายการทำนายชนิดของข้อมูลทุกตัวในชุดข้อมูล ว่าเป็น Normal หรือ Attack ซึ่งขอเรียกว่า intrusion list เป็นผลลัพธ์ที่มีการทำนายชนิดของข้อมูลรายตัว ว่าข้อมูลใดบ้างที่มีโอกาสเป็นการบุกรุกระบบ

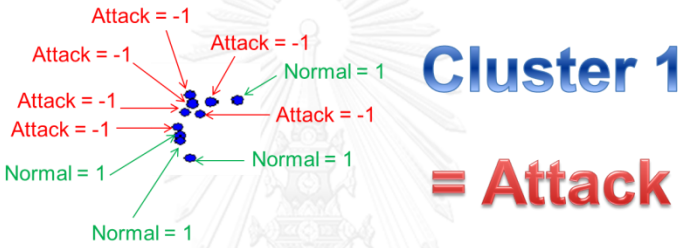
4.2.6 Labeling Clusters

ในขั้นตอนนี้ เราจะนำรายการผลลัพธ์ที่มีโอกาสสูงที่จะเป็นการบุกรุกระบบที่ได้จากขั้นตอนก่อนหน้า มาเพิ่มมิติของการจำแนกผลลัพธ์ จากการวิเคราะห์ข้อมูลเป็นรายตัว ขยายผลมาทำการวิเคราะห์ข้อมูลรายกลุ่ม โดยใช้ผลลัพธ์ intrusion list ทำการวิเคราะห์ร่วมกับ กลุ่มข้อมูล(cluster) ที่ได้จากขั้นตอนการจัดกลุ่มข้อมูล โดยการทำ Majority vote ผลลัพธ์ของสมาชิกแต่ละตัวในกลุ่ม

ข้อมูล ถ้าหากกลุ่มของข้อมูลมีจำนวนของ ข้อมูลการบุกรุก มากกว่าครึ่งหนึ่งของจำนวนข้อมูลทั้งหมดในกลุ่ม กลุ่มนั้นคือกลุ่มข้อมูลที่มีโอกาสสูงที่จะเป็นกลุ่มของข้อมูลการบุกรุก

วิธีการในขั้นตอนนี้ใช้ผลลัพธ์จากการแบ่งกลุ่มจากขั้นตอนการจัดกลุ่มข้อมูลด้วย K-Means algorithm แล้วนำสมาชิกในกลุ่มที่ได้มาพิจารณาชนิดของข้อมูลสำหรับแต่ละกลุ่มก้อน โดยการใช้ intrusion list โดยใช้การนับผลรวมของสมาชิกในกลุ่มนั้นๆว่า มีจำนวนสมาชิกที่ถูกทำนายว่าเป็นข้อมูลการบุกรุก มากกว่า จำนวนสมาชิกที่ถูกทำนายข้อมูลปกติ หรือไม่ ตามขั้นตอนดังนี้

- หากจำนวนการบุกรุกมีค่ามากกว่ากึ่งหนึ่ง ให้จำแนกสมาชิกทั้งกลุ่มนั้นเป็นชนิดการบุกรุก
- หากจำนวนการบุกรุกหากผลรวมมีค่าน้อยกว่ากึ่งหนึ่ง ให้จำแนกสมาชิกในกลุ่มนั้นตามประเภทที่ถูกจำแนกมาในขั้นตอนก่อนหน้า (ผลลัพธ์คงเดิม)



ภาพที่ 28 ตัวอย่างการทำ Majority vote ในกลุ่มข้อมูลผลลัพธ์

จากการทำ Majority vote จะทำให้เรารู้ว่า กลุ่มข้อมูลขนาดเล็กกลุ่มใด ที่มีโอกาสจะเป็นกลุ่มข้อมูลที่มีลักษณะเป็นการบุกรุก (เนื่องผลลัพธ์ของการจัดกลุ่มข้อมูลคือการนำกลุ่มข้อมูลที่ได้มีลักษณะคล้ายคลึงกันสูง ดังนั้นข้อมูลภายในกลุ่มเดียวกัน จึงควรเป็นข้อมูลชนิดเดียวกันนั่นเอง) เพราะ ผลลัพธ์ intrusion list เพียงอย่างเดียวในขั้นตอนก่อนหน้า บอกผลการทำนายชนิดของข้อมูลรายตัว แต่ไม่สามารถบอกกลุ่มก้อนของข้อมูลได้ว่า กลุ่มใดบ้างที่จะมีความน่าจะเป็นที่อยู่ใน category เดียวกัน มีลักษณะคล้ายกัน ซึ่งข้อมูลนี้อาจมีประโยชน์ในการนำไปใช้ต่อไป (อาจทำให้การสร้างกฎในการตรวจจับข้อมูลที่เป็นลักษณะของการบุกรุกระบบได้ง่ายขึ้น จากการพิจารณาลักษณะที่มีเหมือนๆกันของข้อมูลในกลุ่ม แทนที่จะพิจารณาสร้างกฎจากข้อมูลเดี่ยวๆทีละตัว)

ซึ่งผลลัพธ์ที่ได้จากขั้นตอนนี้ จะทำให้ทราบว่าข้อมูลแต่ละตัวใดๆมีลักษณะใกล้เคียงกับข้อมูลตัวอื่นๆใดบ้าง ซึ่งโดยมากมักจะเป็นข้อมูลตัวที่ถูกจัดรวมไว้ในกลุ่มก้อนเดียวกัน ข้อมูลที่ได้จากขั้นตอนนี้จะสามารถนำไปใช้เป็นประโยชน์คือ สามารถนำคุณลักษณะร่วมที่สมาชิกในกลุ่มมีคล้ายกันไปสร้างเป็นกฎในการคัดกรองข้อมูลที่มีลักษณะเดียวกัน เพื่อให้สามารถจำแนกข้อมูลที่เป็นประเภทเดียวกันได้ง่าย

4.2.7 ประเมินความแม่นยำโดยการคำนวณ Detection Rate(DR) และ False Positive Rate(FPR)

ในขั้นตอนสุดท้าย ทำการคำนวณวัดค่า Detection rate และ False positive rate ของผลลัพธ์ แล้วทำการสรุปผลความแม่นยำของผลลัพธ์จากวิธีการที่นำเสนอ และ เปรียบเทียบความแม่นยำกับวิธีการอื่นๆ และ สรุปผลการทดลอง

บทที่ 5
การวัดและประเมินผล

5.1 หลักเกณฑ์การวัดและประเมินผล

ในการทดลองความถูกต้องแม่นยำของผลลัพธ์นี้จะใช้การสร้าง Confusion Matrix เพื่อการคำนวณมาตรวัดสำคัญ คือ Detection Rate และ False Alarm Rate (False Positive Rate) ที่ได้บรรยายรายละเอียดไว้ในบทที่ 2 เป็นหลักเกณฑ์การวัดและประเมินผลการทดลองในส่วนของประสิทธิภาพการจำแนกข้อมูลการบุกรุกระบบจากข้อมูล Log file ขนาดใหญ่

ในส่วนของประสิทธิภาพเชิงเวลาของระบบ ได้ทำการวิเคราะห์ประสิทธิภาพเชิงเวลา และทำการทดลองประมวลผลเพื่อวัดเวลาที่ใช้จริง เพื่อเปรียบเทียบความสัมพันธ์ระหว่างค่าของตัวแปรที่ใช้กับเวลาในการประมวลผล และ ผลของเวลาเปรียบเทียบกับวิธีการอื่นๆ

5.2 การเตรียมข้อมูลทดสอบ

ทำการเตรียมข้อมูลจากชุดข้อมูลต้นฉบับ(KDD'99) ด้วยการ preprocess และ normalize ค่าให้พร้อมสำหรับนำไปประมวลผล โดยรายละเอียดในการเตรียมข้อมูลสำหรับแต่ละการทดลองจะมีการสุ่มจำนวนข้อมูล และ สัดส่วนของข้อมูลในแบบต่างๆ ดังจะกล่าวถึงรายละเอียดอีกครั้งในการทดลองนั้นๆ

5.3 รายละเอียดของระบบที่ใช้ในการทดสอบ

การทดลองผลลัพธ์ดำเนินการสร้างโปรแกรม KSE-test และ ส่วนของการทำงานทั้งหมดโดยใช้ภาษา Java และ การจัดกลุ่มข้อมูลด้วย K-means algorithm ใช้ WEKA Library[30] ซึ่งเป็น JAVA Library การทดลองในส่วนของการประมวลผลโปรแกรมแบบลำดับ (Sequential) ทำบนระบบปฏิบัติการ Windows 7 64-bit บนเครื่องคอมพิวเตอร์ที่มีหน่วยประมวลผล core i5-2500K, 3.3 GHz และ หน่วยความจำ 4 GB DDR3 รายละเอียดตามภาพที่ 29 เพื่อทดสอบความแม่นยำของผลลัพธ์ และ วัดเวลาที่ใช้ในการประมวลผลโปรแกรมแบบลำดับ ในขั้นตอนที่ 5.4 ถึง 5.6

ภาพที่ 29 รายละเอียดของเครื่องที่ใช้ในการทดสอบ

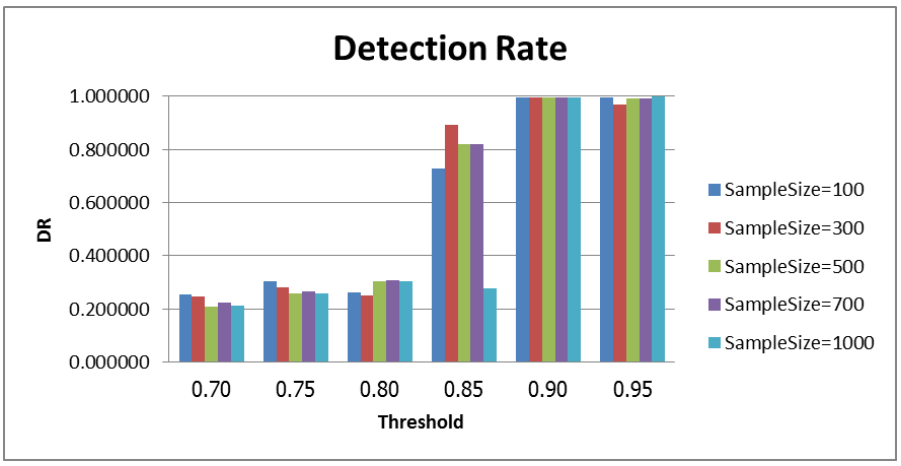
5.4 ทดสอบความสัมพันธ์ของขนาด Sample และ Threshold ที่มีผลต่อความแม่นยำในการจำแนกการบุกรุกระบบ

การทดสอบประสิทธิภาพของ KSE-test ในการทดลองนี้ จะใช้ข้อมูลที่ทำการสุ่มชุดข้อมูลจาก Log file KDD'99 ให้มีขนาด 400,000 บรรทัด มีสัดส่วนของข้อมูลปกติเป็นสัดส่วน 97% และข้อมูลการบุกรุกระบบเป็นสัดส่วน 3% ของชุดข้อมูล สำหรับการทดสอบความสัมพันธ์ของการกำหนดค่า Threshold และ ขนาดของ Sample ที่มีผลต่อความแม่นยำของผลลัพธ์ โดยการแปรผันค่า Threshold โดยใช้ค่า 0.7, 0.75, 0.8, 0.85, 0.9 และ 0.95 เพื่อแสดงความสัมพันธ์ต่อความถูกต้องแม่นยำของผลลัพธ์ที่ได้ และ แปรผันขนาด Sample ที่ใช้สำหรับ KSE-test โดยผลลัพธ์ที่ได้ แสดงในรูปของ Detection Rate(DR), False Positive Rate(FPR) และ Accuracy(ACC) ไว้ใน ตารางที่ 1

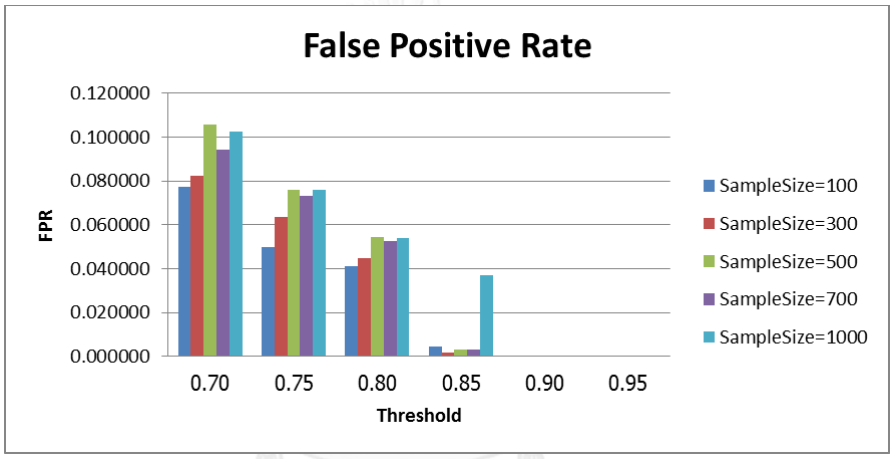
ตารางที่ 1 เปรียบเทียบความสัมพันธ์ของความแม่นยำ กับ ขนาด Sample และ Threshold

Threshold	Sample Size														
	100			300			500			700			1000		
	DR	FPR	ACC	DR	FPR	ACC	DR	FPR	ACC	DR	FPR	ACC	DR	FPR	ACC
0.70	0.256318	0.077209	0.920920	0.247189	0.082479	0.916265	0.207921	0.105714	0.894375	0.225206	0.094353	0.905080	0.212701	0.102438	0.897480
0.75	0.305735	0.049923	0.942900	0.283268	0.063647	0.932663	0.257669	0.076085	0.921815	0.264797	0.073183	0.924580	0.259069	0.076070	0.922013
0.80	0.262049	0.041080	0.944303	0.252499	0.044910	0.941153	0.305731	0.054454	0.940440	0.308344	0.052361	0.941853	0.304649	0.053732	0.940715
0.85	0.726610	0.004541	0.977303	0.891253	0.001541	0.980758	0.819550	0.002997	0.980298	0.819550	0.002997	0.980298	0.278650	0.037183	0.947865
0.90	0.994650	0.000036	0.976473	0.995289	0.000028	0.975783	0.995325	0.000028	0.975828	0.995325	0.000028	0.975828	0.993542	0.000072	0.980700
0.95	0.995640	0.000008	0.971705	0.967742	0.000008	0.970218	0.990762	0.000010	0.971063	0.990762	0.000010	0.971063	0.996896	0.000013	0.974003

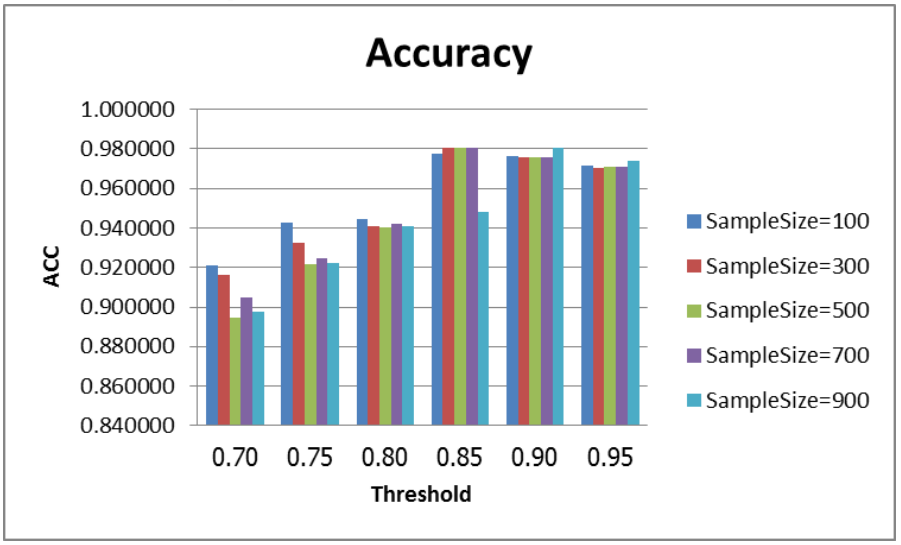
จากผลลัพธ์ที่ได้จากตารางที่ 1 พบว่าที่ระดับ Threshold เดียวกัน ผลลัพธ์ที่ได้จากการใช้ขนาดของ Sample ที่แตกต่างกันนั้น ให้ผลลัพธ์ใกล้เคียงกัน จึงสามารถเลือกใช้ Sample ขนาดใดก็ได้ โดยไม่กระทบต่อความแม่นยำของผลลัพธ์ ซึ่งจากผลการทดลองพบว่า การประมวลผลด้วย KSE-test กับชุดข้อมูลใดๆ มีความสัมพันธ์กับระดับของค่า Threshold ที่ใช้ในการจำแนกเป็นอย่างมาก ดังแสดงในภาพที่ 30 ถึง ภาพที่ 32



ภาพที่ 30 ความสัมพันธ์ ระหว่าง Detection Rate กับ Threshold



ภาพที่ 31 ความสัมพันธ์ ระหว่าง False Positive Rate กับ Threshold



ภาพที่ 32 ความสัมพันธ์ ระหว่าง Accuracy กับ Threshold

โดยการเลือกค่า Threshold ที่สูง ทำให้การเลือกข้อมูลที่ได้มีความถูกต้องแม่นยำสูง Detection Rate สูง และมี False Positive Rate ต่ำ แต่จะพบว่า การเลือกจำแนกข้อมูลที่น้อยเกินไป ทำให้ Accuracy ต่ำกว่าการใช้ค่า Threshold ที่เหมาะสม (การเลือก Threshold ที่สูงเกินไปอาจทำให้ไม่มีการจำแนกข้อมูลใดๆเป็นการบุกรุกระบบเลยก็ได้ ทำให้ Detection Rate = 0 และ False Positive Rate = 1) และ จากผลการทดลอง พบว่า การเลือกใช้ Threshold ที่ต่ำเกินไป อาจทำให้มีการจำแนกผิดพลาดเยอะขึ้น เนื่องจาก มีโอกาสที่จะจำแนกข้อมูลปกติปะปนมาด้วยได้มากกว่าการใช้ Threshold ที่สูง ทำให้ Detection Rate ลดลง และ False Positive Rate สูงขึ้น

การประมาณค่า Threshold ที่เหมาะสมจากการทดลอง จะใช้วิธีการอ้างอิงจากคะแนน KSE-score ที่สูงสุดในชุดข้อมูล ทำการลบด้วย 0.1 จะทำให้ได้ค่า Threshold ใกล้เคียงกับจุดแบ่งที่ให้คุณภาพของผลลัพธ์ดีที่สุด

5.5 ความแม่นยำในการจำแนกการบุกรุกระบบ

ในการทดลองนี้จะใช้ Confusion Matrix และ มาตรวัดสำคัญ คือ Detection Rate และ False Alarm Rate (False Positive Rate) ที่ได้บรรยายไว้ในหัวข้อ 3.2.8 เป็นหลักเกณฑ์การวัดและประเมินผลการทดลองในส่วนของประสิทธิภาพการจำแนกข้อมูลการบุกรุกระบบจากข้อมูล Log file ขนาดใหญ่ ในการทดลองนี้จะทำการสุ่มชุดข้อมูลจาก Log file KDD'99 ให้แต่ละชุดมีขนาด 400,000 บรรทัด จำนวน 5 ชุด สำหรับการวิเคราะห์ความแม่นยำในการจัดจำแนก โดยข้อมูลแต่ละชุด จะมีสัดส่วนของข้อมูลปกติ และ ข้อมูลการบุกรุกที่แตกต่างกัน ดังนี้

- ข้อมูลชุดที่ 1 ประกอบด้วย ข้อมูลปกติ 90% และ ข้อมูลการบุกรุก 10% ของชุดข้อมูล
- ข้อมูลชุดที่ 2 ประกอบด้วย ข้อมูลปกติ 92% และ ข้อมูลการบุกรุก 8% ของชุดข้อมูล
- ข้อมูลชุดที่ 3 ประกอบด้วย ข้อมูลปกติ 94% และ ข้อมูลการบุกรุก 6% ของชุดข้อมูล
- ข้อมูลชุดที่ 4 ประกอบด้วย ข้อมูลปกติ 96% และ ข้อมูลการบุกรุก 4% ของชุดข้อมูล
- ข้อมูลชุดที่ 5 ประกอบด้วย ข้อมูลปกติ 98% และ ข้อมูลการบุกรุก 2% ของชุดข้อมูล

ทำการประมวลผลข้อมูลทั้ง 5 ชุดโดยใช้วิธีการที่นำเสนอ กำหนดค่า Threshold แปรผันในช่วง 0.7-0.95 (เพิ่มค่าทีละ 0.05) และ ใช้ค่า K แปรผันในช่วง 5-55 (เพิ่มค่าขึ้นทีละ 5) และใช้ Nsize = 70% ของชุดข้อมูล ได้ผลลัพธ์ของทั้ง 5 ชุดดังแสดงในตารางที่ 2 ถึง ตารางที่ 6

ตารางที่ 2 Confusion Matrix แสดงผลการทดลองของข้อมูลชุดที่ 1 (90/10)

Threshold	Experiment	TP	FP	TN	FN	TPR	FPR	TNR	FNR	DR	FPR	ACC
T = 0.7	KSE-test	38935	8442	351558	1065	0.973375	0.023450	0.976550	0.026625	0.821812	0.023450	0.976233
	K5	38899	5272	354728	1101	0.972475	0.014644	0.985356	0.027525	0.880646	0.014644	0.984068
	K10	38870	1789	358211	1130	0.971750	0.004969	0.995031	0.028250	0.956000	0.004969	0.992703
	K15	38933	3550	356450	1067	0.973325	0.009861	0.990139	0.026675	0.916437	0.009861	0.988458
	K20	38935	7852	352148	1065	0.973375	0.021811	0.978189	0.026625	0.832176	0.021811	0.977708
	K25	38935	8123	351877	1065	0.973375	0.022564	0.977436	0.026625	0.827383	0.022564	0.977030
	K30	38935	8231	351769	1065	0.973375	0.022864	0.977136	0.026625	0.825489	0.022864	0.976760
	K35	38935	8128	351872	1065	0.973375	0.022578	0.977422	0.026625	0.827295	0.022578	0.977018
	K40	38935	8126	351874	1065	0.973375	0.022572	0.977428	0.026625	0.827330	0.022572	0.977023
	K45	27538	8108	351892	12462	0.688450	0.022522	0.977478	0.311550	0.772541	0.022522	0.948575
	K50	27538	8214	351786	12462	0.688450	0.022817	0.977183	0.311550	0.770251	0.022817	0.948310
	K55	27555	8215	351785	12445	0.688875	0.022819	0.977181	0.311125	0.770338	0.022819	0.948350
T = 0.75	KSE-test	38858	5817	354183	1142	0.971450	0.016158	0.983842	0.028550	0.869793	0.016158	0.982603
	K5	38822	4015	355985	1178	0.970550	0.011153	0.988847	0.029450	0.906273	0.011153	0.987018
	K10	38801	1643	358357	1199	0.970025	0.004564	0.995436	0.029975	0.959376	0.004564	0.992895
	K15	38858	3400	356600	1142	0.971450	0.009444	0.990556	0.028550	0.919542	0.009444	0.988645
	K20	38858	5229	354771	1142	0.971450	0.014525	0.985475	0.028550	0.881394	0.014525	0.984073
	K25	38858	5500	354500	1142	0.971450	0.015278	0.984722	0.028550	0.876009	0.015278	0.983395
	K30	38858	5608	354392	1142	0.971450	0.015578	0.984422	0.028550	0.873881	0.015578	0.983125
	K35	38858	5505	354495	1142	0.971450	0.015292	0.984708	0.028550	0.875910	0.015292	0.983383
	K40	38858	5503	354497	1142	0.971450	0.015286	0.984714	0.028550	0.875950	0.015286	0.983388
	K45	27461	5485	354515	12539	0.686525	0.015236	0.984764	0.313475	0.833515	0.015236	0.954940
	K50	27461	5591	354409	12539	0.686525	0.015531	0.984469	0.313475	0.830842	0.015531	0.954675
	K55	27478	5592	354408	12522	0.686950	0.015533	0.984467	0.313050	0.830904	0.015533	0.954715
T = 0.8	KSE-test	30207	2915	357085	9793	0.755175	0.008097	0.991903	0.244825	0.911992	0.008097	0.968230
	K5	30171	2685	357315	9829	0.754275	0.007458	0.992542	0.245725	0.918280	0.007458	0.968715
	K10	30151	1418	358582	9849	0.753775	0.003939	0.996061	0.246225	0.955083	0.003939	0.971833
	K15	30207	2144	357856	9793	0.755175	0.005956	0.994044	0.244825	0.933727	0.005956	0.970158
	K20	30207	2618	357382	9793	0.755175	0.007272	0.992728	0.244825	0.920244	0.007272	0.968973
	K25	30207	2915	357085	9793	0.755175	0.008097	0.991903	0.244825	0.911992	0.008097	0.968230
	K30	30207	2914	357086	9793	0.755175	0.008094	0.991906	0.244825	0.912020	0.008094	0.968233
	K35	30207	2915	357085	9793	0.755175	0.008097	0.991903	0.244825	0.911992	0.008097	0.968230
	K40	30207	2915	357085	9793	0.755175	0.008097	0.991903	0.244825	0.911992	0.008097	0.968230
	K45	20412	2896	357104	19588	0.510300	0.008044	0.991956	0.489700	0.875751	0.008044	0.943790
	K50	20412	2895	357105	19588	0.510300	0.008042	0.991958	0.489700	0.875788	0.008042	0.943793
	K55	20429	2896	357104	19571	0.510725	0.008044	0.991956	0.489275	0.875841	0.008044	0.943833
T = 0.85	KSE-test	3077	3	359997	36923	0.076925	0.000008	0.999992	0.923075	0.999026	0.000008	0.907685
	K5	3077	3	359997	36923	0.076925	0.000008	0.999992	0.923075	0.999026	0.000008	0.907685
	K10	3077	3	359997	36923	0.076925	0.000008	0.999992	0.923075	0.999026	0.000008	0.907685
	K15	3077	3	359997	36923	0.076925	0.000008	0.999992	0.923075	0.999026	0.000008	0.907685
	K20	3077	3	359997	36923	0.076925	0.000008	0.999992	0.923075	0.999026	0.000008	0.907685
	K25	3077	3	359997	36923	0.076925	0.000008	0.999992	0.923075	0.999026	0.000008	0.907685
	K30	3077	3	359997	36923	0.076925	0.000008	0.999992	0.923075	0.999026	0.000008	0.907685
	K35	3077	3	359997	36923	0.076925	0.000008	0.999992	0.923075	0.999026	0.000008	0.907685
	K40	3077	3	359997	36923	0.076925	0.000008	0.999992	0.923075	0.999026	0.000008	0.907685
	K45	2	3	359997	39998	0.000050	0.000008	0.999992	0.999950	0.400000	0.000008	0.899998
	K50	2	3	359997	39998	0.000050	0.000008	0.999992	0.999950	0.400000	0.000008	0.899998
	K55	2	3	359997	39998	0.000050	0.000008	0.999992	0.999950	0.400000	0.000008	0.899998
T = 0.9	KSE-test	19	0	360000	39981	0.000475	0.000000	1.000000	0.999525	1.000000	0.000000	0.900048
	K5	19	0	360000	39981	0.000475	0.000000	1.000000	0.999525	1.000000	0.000000	0.900048
	K10	19	0	360000	39981	0.000475	0.000000	1.000000	0.999525	1.000000	0.000000	0.900048
	K15	19	0	360000	39981	0.000475	0.000000	1.000000	0.999525	1.000000	0.000000	0.900048
	K20	19	0	360000	39981	0.000475	0.000000	1.000000	0.999525	1.000000	0.000000	0.900048
	K25	19	0	360000	39981	0.000475	0.000000	1.000000	0.999525	1.000000	0.000000	0.900048
	K30	19	0	360000	39981	0.000475	0.000000	1.000000	0.999525	1.000000	0.000000	0.900048
	K35	19	0	360000	39981	0.000475	0.000000	1.000000	0.999525	1.000000	0.000000	0.900048
	K40	19	0	360000	39981	0.000475	0.000000	1.000000	0.999525	1.000000	0.000000	0.900048
	K45	0	0	360000	40000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.900000
	K50	0	0	360000	40000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.900000
	K55	0	0	360000	40000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.900000
T = 0.95	KSE-test	0	0	360000	40000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.900000
	K5	0	0	360000	40000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.900000
	K10	0	0	360000	40000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.900000
	K15	0	0	360000	40000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.900000
	K20	0	0	360000	40000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.900000
	K25	0	0	360000	40000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.900000
	K30	0	0	360000	40000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.900000
	K35	0	0	360000	40000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.900000
	K40	0	0	360000	40000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.900000
	K45	0	0	360000	40000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.900000
	K50	0	0	360000	40000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.900000
	K55	0	0	360000	40000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.900000

ตารางที่ 3 Confusion Matrix แสดงผลการทดลองของข้อมูลชุดที่ 2 (92/8)

Threshold	Experiment	TP	FP	TN	FN	TPR	FPR	TNR	FNR	DR	FPR	ACC
T = 0.7	KSE-test	31646	8166	359834	354	0.988938	0.022190	0.977810	0.011063	0.794886	0.022190	0.978700
	K5	31011	4373	363627	989	0.969094	0.011883	0.988117	0.030906	0.876413	0.011883	0.986595
	K10	31583	2636	365364	417	0.986969	0.007163	0.992837	0.013031	0.922967	0.007163	0.992368
	K15	31645	4127	363873	355	0.988906	0.011215	0.988785	0.011094	0.884630	0.011215	0.988795
	K20	31645	6833	361167	355	0.988906	0.018568	0.981432	0.011094	0.822418	0.018568	0.982030
	K25	31646	8048	359952	354	0.988938	0.021870	0.978130	0.011063	0.797249	0.021870	0.978995
	K30	31646	7792	360208	354	0.988938	0.021174	0.978826	0.011063	0.802424	0.021174	0.979635
	K35	31646	7793	360207	354	0.988938	0.021177	0.978823	0.011063	0.802404	0.021177	0.979633
	K40	31646	7788	360212	354	0.988938	0.021163	0.978837	0.011063	0.802505	0.021163	0.979645
	K45	31646	7796	360204	354	0.988938	0.021185	0.978815	0.011063	0.802343	0.021185	0.979625
	K50	22528	7896	360104	9472	0.704000	0.021457	0.978543	0.296000	0.740468	0.021457	0.956580
	K55	22530	8160	359840	9470	0.704063	0.022174	0.977826	0.295938	0.734115	0.022174	0.955925
T = 0.75	KSE-test	31613	8010	359990	387	0.987906	0.021766	0.978234	0.012094	0.797845	0.021766	0.979008
	K5	30989	4342	363658	1011	0.968406	0.011799	0.988201	0.031594	0.877105	0.011799	0.986618
	K10	31550	2482	365518	450	0.985938	0.006745	0.993255	0.014063	0.927069	0.006745	0.992670
	K15	31612	3971	364029	388	0.987875	0.010791	0.989209	0.012125	0.888402	0.010791	0.989103
	K20	31612	6677	361323	388	0.987875	0.018144	0.981856	0.012125	0.825616	0.018144	0.982338
	K25	31613	7892	360108	387	0.987906	0.021446	0.978554	0.012094	0.800228	0.021446	0.979303
	K30	31613	7636	360364	387	0.987906	0.020750	0.979250	0.012094	0.805447	0.020750	0.979943
	K35	31613	7637	360363	387	0.987906	0.020753	0.979247	0.012094	0.805427	0.020753	0.979940
	K40	31613	7632	360368	387	0.987906	0.020739	0.979261	0.012094	0.805529	0.020739	0.979953
	K45	31613	7640	360360	387	0.987906	0.020761	0.979239	0.012094	0.805365	0.020761	0.979933
	K50	22495	7740	360260	9505	0.702969	0.021033	0.978967	0.297031	0.744005	0.021033	0.956888
	K55	22497	8004	359996	9503	0.703031	0.021750	0.978250	0.296969	0.737582	0.021750	0.956233
T = 0.8	KSE-test	31320	5667	362333	680	0.978750	0.015399	0.984601	0.021250	0.846784	0.015399	0.984133
	K5	30967	3329	364671	1033	0.967719	0.009046	0.990954	0.032281	0.902933	0.009046	0.989095
	K10	31265	2334	365666	735	0.977031	0.006342	0.993658	0.022969	0.930534	0.006342	0.992328
	K15	31320	3823	364177	680	0.978750	0.010389	0.989611	0.021250	0.891216	0.010389	0.988743
	K20	31320	5326	362674	680	0.978750	0.014473	0.985527	0.021250	0.854664	0.014473	0.984985
	K25	31320	5552	362448	680	0.978750	0.015087	0.984913	0.021250	0.849425	0.015087	0.984420
	K30	31320	5298	362702	680	0.978750	0.014397	0.985603	0.021250	0.855317	0.014397	0.985055
	K35	31320	5298	362702	680	0.978750	0.014397	0.985603	0.021250	0.855317	0.014397	0.985055
	K40	31320	5294	362706	680	0.978750	0.014386	0.985614	0.021250	0.855410	0.014386	0.985065
	K45	31320	5301	362699	680	0.978750	0.014405	0.985595	0.021250	0.855247	0.014405	0.985048
	K50	22202	5401	362599	9798	0.693813	0.014677	0.985323	0.306188	0.804333	0.014677	0.962003
	K55	22204	5661	362339	9796	0.693875	0.015383	0.984617	0.306125	0.796842	0.015383	0.961358
T = 0.85	KSE-test	30977	2910	365090	1023	0.968031	0.007908	0.992092	0.031969	0.914126	0.007908	0.990168
	K5	30792	2113	365887	1208	0.962250	0.005742	0.994258	0.037750	0.935785	0.005742	0.991698
	K10	30923	2135	365865	1077	0.966344	0.005802	0.994198	0.033656	0.935417	0.005802	0.991970
	K15	30977	2660	365340	1023	0.968031	0.007228	0.992772	0.031969	0.920920	0.007228	0.990793
	K20	30977	2910	365090	1023	0.968031	0.007908	0.992092	0.031969	0.914126	0.007908	0.990168
	K25	30977	2910	365090	1023	0.968031	0.007908	0.992092	0.031969	0.914126	0.007908	0.990168
	K30	30977	2910	365090	1023	0.968031	0.007908	0.992092	0.031969	0.914126	0.007908	0.990168
	K35	30977	2910	365090	1023	0.968031	0.007908	0.992092	0.031969	0.914126	0.007908	0.990168
	K40	30977	2910	365090	1023	0.968031	0.007908	0.992092	0.031969	0.914126	0.007908	0.990168
	K45	30977	2910	365090	1023	0.968031	0.007908	0.992092	0.031969	0.914126	0.007908	0.990168
	K50	21859	2904	365096	10141	0.683094	0.007891	0.992109	0.316906	0.882728	0.007891	0.967388
	K55	21861	2904	365096	10139	0.683156	0.007891	0.992109	0.316844	0.882738	0.007891	0.967393
T = 0.9	KSE-test	6258	5	367995	25742	0.195563	0.000014	0.999986	0.804438	0.999202	0.000014	0.935633
	K5	6258	5	367995	25742	0.195563	0.000014	0.999986	0.804438	0.999202	0.000014	0.935633
	K10	6258	5	367995	25742	0.195563	0.000014	0.999986	0.804438	0.999202	0.000014	0.935633
	K15	6258	5	367995	25742	0.195563	0.000014	0.999986	0.804438	0.999202	0.000014	0.935633
	K20	6258	5	367995	25742	0.195563	0.000014	0.999986	0.804438	0.999202	0.000014	0.935633
	K25	6258	5	367995	25742	0.195563	0.000014	0.999986	0.804438	0.999202	0.000014	0.935633
	K30	6258	5	367995	25742	0.195563	0.000014	0.999986	0.804438	0.999202	0.000014	0.935633
	K35	6258	5	367995	25742	0.195563	0.000014	0.999986	0.804438	0.999202	0.000014	0.935633
	K40	6258	5	367995	25742	0.195563	0.000014	0.999986	0.804438	0.999202	0.000014	0.935633
	K45	6258	5	367995	25742	0.195563	0.000014	0.999986	0.804438	0.999202	0.000014	0.935633
	K50	3397	5	367995	28603	0.106156	0.000014	0.999986	0.893844	0.998530	0.000014	0.928480
	K55	3397	5	367995	28603	0.106156	0.000014	0.999986	0.893844	0.998530	0.000014	0.928480
T = 0.95	KSE-test	0	0	368000	32000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.920000
	K5	0	0	368000	32000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.920000
	K10	0	0	368000	32000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.920000
	K15	0	0	368000	32000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.920000
	K20	0	0	368000	32000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.920000
	K25	0	0	368000	32000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.920000
	K30	0	0	368000	32000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.920000
	K35	0	0	368000	32000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.920000
	K40	0	0	368000	32000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.920000
	K45	0	0	368000	32000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.920000
	K50	0	0	368000	32000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.920000
	K55	0	0	368000	32000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.920000

ตารางที่ 4 Confusion Matrix แสดงผลการทดลองของข้อมูลชุดที่ 3 (94/6)

Threshold	Experiment	TP	FP	TN	FN	TPR	FPR	TNR	FNR	DR	FPR	ACC
T = 0.7	KSE-test	23706	20315	355685	294	0.987750	0.054029	0.945971	0.012250	0.538516	0.054029	0.948478
	K5	23108	10625	365375	892	0.962833	0.028258	0.971742	0.037167	0.685027	0.028258	0.971208
	K10	23661	9616	366384	339	0.985875	0.025574	0.974426	0.014125	0.711032	0.025574	0.975113
	K15	23697	13676	362324	303	0.987375	0.036372	0.963628	0.012625	0.634067	0.036372	0.965053
	K20	23698	18149	357851	302	0.987417	0.048269	0.951731	0.012583	0.566301	0.048269	0.953873
	K25	23706	20049	355951	294	0.987750	0.053322	0.946678	0.012250	0.541790	0.053322	0.949143
	K30	23596	18682	357318	404	0.983167	0.049686	0.950314	0.016833	0.558115	0.049686	0.952285
	K35	23704	19991	356009	296	0.987667	0.053168	0.946832	0.012333	0.542488	0.053168	0.949283
	K40	23705	19987	356013	295	0.987708	0.053157	0.946843	0.012292	0.542548	0.053157	0.949295
	K45	23705	19987	356013	295	0.987708	0.053157	0.946843	0.012292	0.542548	0.053157	0.949295
	K50	23704	19988	356012	296	0.987667	0.053160	0.946840	0.012333	0.542525	0.053160	0.949290
	K55	23705	19995	356005	295	0.987708	0.053178	0.946822	0.012292	0.542449	0.053178	0.949275
T = 0.75	KSE-test	23690	15119	360881	310	0.987083	0.040210	0.959790	0.012917	0.610425	0.040210	0.961428
	K5	23103	6839	369161	897	0.962625	0.018189	0.981811	0.037375	0.771592	0.018189	0.980660
	K10	23648	4495	371505	352	0.985333	0.011955	0.988045	0.014667	0.840280	0.011955	0.987883
	K15	23684	8561	367439	316	0.986833	0.022769	0.977231	0.013167	0.734501	0.022769	0.977808
	K20	23684	12955	363045	316	0.986833	0.034455	0.965545	0.013167	0.646415	0.034455	0.966823
	K25	23690	14855	361145	310	0.987083	0.039508	0.960492	0.012917	0.614606	0.039508	0.962088
	K30	23591	14575	361425	409	0.982958	0.038763	0.961237	0.017042	0.618116	0.038763	0.962540
	K35	23689	14797	361203	311	0.987042	0.039354	0.960646	0.012958	0.615523	0.039354	0.962230
	K40	23690	14793	361207	310	0.987083	0.039343	0.960657	0.012917	0.615596	0.039343	0.962243
	K45	23690	14793	361207	310	0.987083	0.039343	0.960657	0.012917	0.615596	0.039343	0.962243
	K50	23690	14794	361206	310	0.987083	0.039346	0.960654	0.012917	0.615581	0.039346	0.962240
	K55	23690	14800	361200	310	0.987083	0.039362	0.960638	0.012917	0.615485	0.039362	0.962225
T = 0.8	KSE-test	23673	8234	367766	327	0.986375	0.021899	0.978101	0.013625	0.741938	0.021899	0.978598
	K5	23097	3979	372021	903	0.962375	0.010582	0.989418	0.037625	0.853043	0.010582	0.987795
	K10	23631	1179	374821	369	0.984625	0.003136	0.996864	0.015375	0.952479	0.003136	0.996130
	K15	23667	2704	373296	333	0.986125	0.007191	0.992809	0.013875	0.897463	0.007191	0.992408
	K20	23667	6070	369930	333	0.986125	0.016144	0.983856	0.013875	0.795877	0.016144	0.983993
	K25	23673	7970	368030	327	0.986375	0.021197	0.978803	0.013625	0.748128	0.021197	0.979258
	K30	23587	8135	367865	413	0.982792	0.021636	0.978364	0.017208	0.743553	0.021636	0.978630
	K35	23673	7912	368088	327	0.986375	0.021043	0.978957	0.013625	0.749501	0.021043	0.979403
	K40	23673	7908	368092	327	0.986375	0.021032	0.978968	0.013625	0.749596	0.021032	0.979413
	K45	23673	7908	368092	327	0.986375	0.021032	0.978968	0.013625	0.749596	0.021032	0.979413
	K50	23673	7909	368091	327	0.986375	0.021035	0.978965	0.013625	0.749573	0.021035	0.979410
	K55	23673	7915	368085	327	0.986375	0.021051	0.978949	0.013625	0.749430	0.021051	0.979395
T = 0.85	KSE-test	23631	6012	369988	369	0.984625	0.015989	0.984011	0.015375	0.797187	0.015989	0.984048
	K5	23073	2879	373121	927	0.961375	0.007657	0.992343	0.038625	0.889064	0.007657	0.990485
	K10	23595	1085	374915	405	0.983125	0.002886	0.997114	0.016875	0.956037	0.002886	0.996275
	K15	23630	1584	374416	370	0.984583	0.004213	0.995787	0.015417	0.937178	0.004213	0.995115
	K20	23630	3851	372149	370	0.984583	0.010242	0.989758	0.015417	0.859867	0.010242	0.989448
	K25	23631	5751	370249	369	0.984625	0.015295	0.984705	0.015375	0.804268	0.015295	0.984700
	K30	23565	5929	370071	435	0.981875	0.015769	0.984231	0.018125	0.798976	0.015769	0.984090
	K35	23631	5693	370307	369	0.984625	0.015141	0.984859	0.015375	0.805859	0.015141	0.984845
	K40	23631	5688	370312	369	0.984625	0.015128	0.984872	0.015375	0.805996	0.015128	0.984858
	K45	23631	5688	370312	369	0.984625	0.015128	0.984872	0.015375	0.805996	0.015128	0.984858
	K50	23631	5689	370311	369	0.984625	0.015130	0.984870	0.015375	0.805969	0.015130	0.984855
	K55	23631	5695	370305	369	0.984625	0.015146	0.984854	0.015375	0.805804	0.015146	0.984840
T = 0.9	KSE-test	23377	4037	371963	623	0.974042	0.010737	0.989263	0.025958	0.852739	0.010737	0.988350
	K5	23046	1981	374019	954	0.960250	0.005269	0.994731	0.039750	0.920845	0.005269	0.992663
	K10	23342	1007	374993	658	0.972583	0.002678	0.997322	0.027417	0.958643	0.002678	0.995838
	K15	23377	1507	374493	623	0.974042	0.004008	0.995992	0.025958	0.939439	0.004008	0.994675
	K20	23377	3000	373000	623	0.974042	0.007979	0.992021	0.025958	0.886265	0.007979	0.990943
	K25	23377	3814	372186	623	0.974042	0.010144	0.989856	0.025958	0.859733	0.010144	0.988908
	K30	23345	4006	371994	655	0.972708	0.010654	0.989346	0.027292	0.853534	0.010654	0.988348
	K35	23377	4034	371966	623	0.974042	0.010729	0.989271	0.025958	0.852833	0.010729	0.988358
	K40	23377	4032	371968	623	0.974042	0.010723	0.989277	0.025958	0.852895	0.010723	0.988363
	K45	23377	4032	371968	623	0.974042	0.010723	0.989277	0.025958	0.852895	0.010723	0.988363
	K50	23377	4033	371967	623	0.974042	0.010726	0.989274	0.025958	0.852864	0.010726	0.988360
	K55	23377	4034	371966	623	0.974042	0.010729	0.989271	0.025958	0.852833	0.010729	0.988358
T = 0.95	KSE-test	3116	20	375980	20884	0.129833	0.000053	0.999947	0.870167	0.993622	0.000053	0.947740
	K5	3116	20	375980	20884	0.129833	0.000053	0.999947	0.870167	0.993622	0.000053	0.947740
	K10	3116	20	375980	20884	0.129833	0.000053	0.999947	0.870167	0.993622	0.000053	0.947740
	K15	3116	20	375980	20884	0.129833	0.000053	0.999947	0.870167	0.993622	0.000053	0.947740
	K20	3116	20	375980	20884	0.129833	0.000053	0.999947	0.870167	0.993622	0.000053	0.947740
	K25	3116	20	375980	20884	0.129833	0.000053	0.999947	0.870167	0.993622	0.000053	0.947740
	K30	3116	20	375980	20884	0.129833	0.000053	0.999947	0.870167	0.993622	0.000053	0.947740
	K35	3116	20	375980	20884	0.129833	0.000053	0.999947	0.870167	0.993622	0.000053	0.947740
	K40	3116	20	375980	20884	0.129833	0.000053	0.999947	0.870167	0.993622	0.000053	0.947740
	K45	3116	20	375980	20884	0.129833	0.000053	0.999947	0.870167	0.993622	0.000053	0.947740
	K50	3116	20	375980	20884	0.129833	0.000053	0.999947	0.870167	0.993622	0.000053	0.947740
	K55	3116	20	375980	20884	0.129833	0.000053	0.999947	0.870167	0.993622	0.000053	0.947740

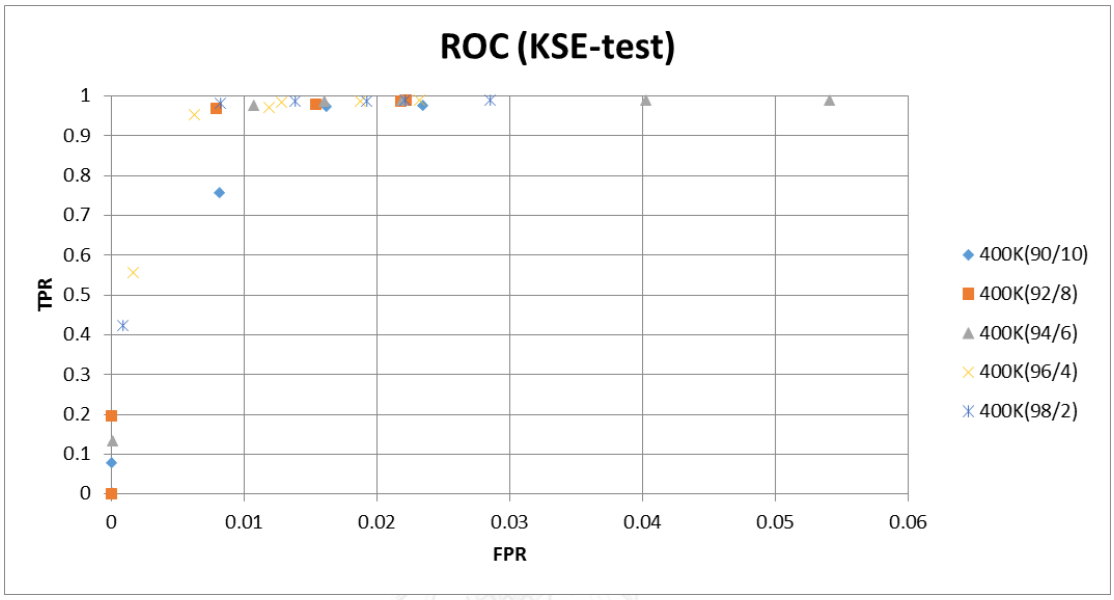
ตารางที่ 5 Confusion Matrix แสดงผลการทดลองของข้อมูลชุดที่ 4 (96/4)

Threshold	Experiment	TP	FP	TN	FN	TPR	FPR	TNR	FNR	DR	FPR	ACC
T = 0.7	KSE-test	15781	8920	375080	219	0.986313	0.023229	0.976771	0.013688	0.638881	0.023229	0.977153
	K5	15781	4482	379518	219	0.986313	0.011672	0.988328	0.013688	0.778809	0.011672	0.988248
	K10	15769	1226	382774	231	0.985563	0.003193	0.996807	0.014438	0.927861	0.003193	0.996358
	K15	15779	7340	376660	221	0.986188	0.019115	0.980885	0.013813	0.682512	0.019115	0.981098
	K20	15780	8362	375638	220	0.986250	0.021776	0.978224	0.013750	0.653633	0.021776	0.978545
	K25	15781	8345	375655	219	0.986313	0.021732	0.978268	0.013688	0.654108	0.021732	0.978590
	K30	15781	8345	375655	219	0.986313	0.021732	0.978268	0.013688	0.654108	0.021732	0.978590
	K35	15781	8347	375653	219	0.986313	0.021737	0.978263	0.013688	0.654053	0.021737	0.978585
	K40	15781	8636	375364	219	0.986313	0.022490	0.977510	0.013688	0.646312	0.022490	0.977863
	K45	15781	8580	375420	219	0.986313	0.022344	0.977656	0.013688	0.647798	0.022344	0.978003
	K50	15781	8593	375407	219	0.986313	0.022378	0.977622	0.013688	0.647452	0.022378	0.977970
	K55	15781	8607	375393	219	0.986313	0.022414	0.977586	0.013688	0.647081	0.022414	0.977935
T = 0.75	KSE-test	15742	7202	376798	258	0.983875	0.018755	0.981245	0.016125	0.686105	0.018755	0.981350
	K5	15742	4278	379722	258	0.983875	0.011141	0.988859	0.016125	0.786314	0.011141	0.988660
	K10	15732	1020	382980	268	0.983250	0.002656	0.997344	0.016750	0.939112	0.002656	0.996780
	K15	15740	5622	378378	260	0.983750	0.014641	0.985359	0.016250	0.736822	0.014641	0.985295
	K20	15741	6644	377356	259	0.983813	0.017302	0.982698	0.016188	0.703194	0.017302	0.982743
	K25	15742	6627	377373	258	0.983875	0.017258	0.982742	0.016125	0.703742	0.017258	0.982788
	K30	15742	6627	377373	258	0.983875	0.017258	0.982742	0.016125	0.703742	0.017258	0.982788
	K35	15742	6629	377371	258	0.983875	0.017263	0.982737	0.016125	0.703679	0.017263	0.982783
	K40	15742	6918	377082	258	0.983875	0.018016	0.981984	0.016125	0.694704	0.018016	0.982060
	K45	15742	6862	377138	258	0.983875	0.017870	0.982130	0.016125	0.696425	0.017870	0.982200
	K50	15742	6875	377125	258	0.983875	0.017904	0.982096	0.016125	0.696025	0.017904	0.982168
	K55	15742	6889	377111	258	0.983875	0.017940	0.982060	0.016125	0.695595	0.017940	0.982133
T = 0.8	KSE-test	15715	4910	379090	285	0.982188	0.012786	0.987214	0.017813	0.761939	0.012786	0.987013
	K5	15715	3214	380786	285	0.982188	0.008370	0.991630	0.017813	0.830208	0.008370	0.991253
	K10	15710	978	383022	290	0.981875	0.002547	0.997453	0.018125	0.941395	0.002547	0.996830
	K15	15715	4352	379648	285	0.982188	0.011333	0.988667	0.017813	0.783127	0.011333	0.988408
	K20	15715	4352	379648	285	0.982188	0.011333	0.988667	0.017813	0.783127	0.011333	0.988408
	K25	15715	4335	379665	285	0.982188	0.011289	0.988711	0.017813	0.783791	0.011289	0.988450
	K30	15715	4335	379665	285	0.982188	0.011289	0.988711	0.017813	0.783791	0.011289	0.988450
	K35	15715	4337	379663	285	0.982188	0.011294	0.988706	0.017813	0.783712	0.011294	0.988445
	K40	15715	4626	379374	285	0.982188	0.012047	0.987953	0.017813	0.772578	0.012047	0.987723
	K45	15715	4570	379430	285	0.982188	0.011901	0.988099	0.017813	0.774710	0.011901	0.987863
	K50	15715	4583	379417	285	0.982188	0.011935	0.988065	0.017813	0.774214	0.011935	0.987830
	K55	15715	4597	379403	285	0.982188	0.011971	0.988029	0.017813	0.773681	0.011971	0.987795
T = 0.85	KSE-test	15502	4573	379427	498	0.968875	0.011909	0.988091	0.031125	0.772204	0.011909	0.987323
	K5	15502	3165	380835	498	0.968875	0.008242	0.991758	0.031125	0.830449	0.008242	0.990843
	K10	15497	929	383071	503	0.968563	0.002419	0.997581	0.031438	0.943443	0.002419	0.996420
	K15	15502	4303	379697	498	0.968875	0.011206	0.988794	0.031125	0.782732	0.011206	0.987998
	K20	15502	4303	379697	498	0.968875	0.011206	0.988794	0.031125	0.782732	0.011206	0.987998
	K25	15502	4286	379714	498	0.968875	0.011161	0.988839	0.031125	0.783404	0.011161	0.988040
	K30	15502	4286	379714	498	0.968875	0.011161	0.988839	0.031125	0.783404	0.011161	0.988040
	K35	15502	4288	379712	498	0.968875	0.011167	0.988833	0.031125	0.783325	0.011167	0.988035
	K40	15502	4289	379711	498	0.968875	0.011169	0.988831	0.031125	0.783285	0.011169	0.988033
	K45	15502	4521	379479	498	0.968875	0.011773	0.988227	0.031125	0.774210	0.011773	0.987453
	K50	15502	4534	379466	498	0.968875	0.011807	0.988193	0.031125	0.773707	0.011807	0.987420
	K55	15502	4548	379452	498	0.968875	0.011844	0.988156	0.031125	0.773167	0.011844	0.987385
T = 0.9	KSE-test	15213	2390	381610	787	0.950813	0.006224	0.993776	0.049188	0.864228	0.006224	0.992058
	K5	15213	2096	381904	787	0.950813	0.005458	0.994542	0.049188	0.878907	0.005458	0.992793
	K10	15211	863	383137	789	0.950688	0.002247	0.997753	0.049313	0.946311	0.002247	0.995870
	K15	15213	2154	381846	787	0.950813	0.005609	0.994391	0.049188	0.875972	0.005609	0.992648
	K20	15213	2154	381846	787	0.950813	0.005609	0.994391	0.049188	0.875972	0.005609	0.992648
	K25	15213	2154	381846	787	0.950813	0.005609	0.994391	0.049188	0.875972	0.005609	0.992648
	K30	15213	2154	381846	787	0.950813	0.005609	0.994391	0.049188	0.875972	0.005609	0.992648
	K35	15213	2156	381844	787	0.950813	0.005615	0.994385	0.049188	0.875871	0.005615	0.992643
	K40	15213	2156	381844	787	0.950813	0.005615	0.994385	0.049188	0.875871	0.005615	0.992643
	K45	15213	2390	381610	787	0.950813	0.006224	0.993776	0.049188	0.864228	0.006224	0.992058
	K50	15213	2390	381610	787	0.950813	0.006224	0.993776	0.049188	0.864228	0.006224	0.992058
	K55	15213	2390	381610	787	0.950813	0.006224	0.993776	0.049188	0.864228	0.006224	0.992058
T = 0.95	KSE-test	8873	628	383372	7127	0.554563	0.001635	0.998365	0.445438	0.933902	0.001635	0.980613
	K5	8873	628	383372	7127	0.554563	0.001635	0.998365	0.445438	0.933902	0.001635	0.980613
	K10	8873	628	383372	7127	0.554563	0.001635	0.998365	0.445438	0.933902	0.001635	0.980613
	K15	8873	628	383372	7127	0.554563	0.001635	0.998365	0.445438	0.933902	0.001635	0.980613
	K20	8873	628	383372	7127	0.554563	0.001635	0.998365	0.445438	0.933902	0.001635	0.980613
	K25	8873	628	383372	7127	0.554563	0.001635	0.998365	0.445438	0.933902	0.001635	0.980613
	K30	8873	628	383372	7127	0.554563	0.001635	0.998365	0.445438	0.933902	0.001635	0.980613
	K35	8873	628	383372	7127	0.554563	0.001635	0.998365	0.445438	0.933902	0.001635	0.980613
	K40	8873	628	383372	7127	0.554563	0.001635	0.998365	0.445438	0.933902	0.001635	0.980613
	K45	8873	628	383372	7127	0.554563	0.001635	0.998365	0.445438	0.933902	0.001635	0.980613
	K50	8873	628	383372	7127	0.554563	0.001635	0.998365	0.445438	0.933902	0.001635	0.980613
	K55	8873	628	383372	7127	0.554563	0.001635	0.998365	0.445438	0.933902	0.001635	0.980613

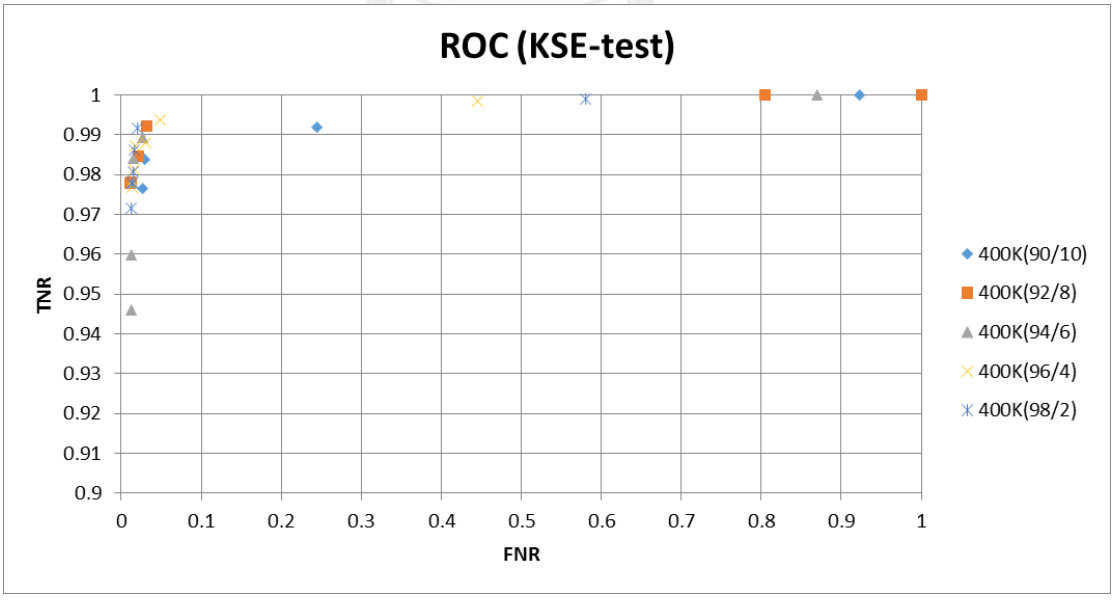
ตารางที่ 6 Confusion Matrix แสดงผลการทดลองของข้อมูลชุดที่ 5 (98/2)

Threshold	Experiment	TP	FP	TN	FN	TPR	FPR	TNR	FNR	DR	FPR	ACC
T = 0.7	KSE-test	7902	9287	382713	98	0.987750	0.023691	0.976309	0.012250	0.459713	0.023691	0.976538
	K5	7902	4617	387383	98	0.987750	0.011778	0.988222	0.012250	0.631201	0.011778	0.988213
	K10	7899	8066	383934	101	0.987375	0.020577	0.979423	0.012625	0.494770	0.020577	0.979583
	K15	7899	8068	383932	101	0.987375	0.020582	0.979418	0.012625	0.494708	0.020582	0.979578
	K20	7899	8864	383136	101	0.987375	0.022612	0.977388	0.012625	0.471216	0.022612	0.977588
	K25	7892	2272	389728	108	0.986500	0.005796	0.994204	0.013500	0.776466	0.005796	0.994050
	K30	7899	8864	383136	101	0.987375	0.022612	0.977388	0.012625	0.471216	0.022612	0.977588
	K35	7899	8864	383136	101	0.987375	0.022612	0.977388	0.012625	0.471216	0.022612	0.977588
	K40	7901	9037	382963	99	0.987625	0.023054	0.976946	0.012375	0.466466	0.023054	0.977160
	K45	7901	9037	382963	99	0.987625	0.023054	0.976946	0.012375	0.466466	0.023054	0.977160
	K50	7901	9036	382964	99	0.987625	0.023051	0.976949	0.012375	0.466493	0.023051	0.977163
	K55	7899	9036	382964	101	0.987375	0.023051	0.976949	0.012625	0.466430	0.023051	0.977158
T = 0.75	KSE-test	7896	2526	389474	104	0.987000	0.006444	0.993556	0.013000	0.757628	0.006444	0.993425
	K5	7896	1793	390207	104	0.987000	0.004574	0.995426	0.013000	0.814945	0.004574	0.995258
	K10	7893	1414	390586	107	0.986625	0.003607	0.996393	0.013375	0.848071	0.003607	0.996198
	K15	7893	1414	390586	107	0.986625	0.003607	0.996393	0.013375	0.848071	0.003607	0.996198
	K20	7893	2104	389896	107	0.986625	0.005367	0.994633	0.013375	0.789537	0.005367	0.994473
	K25	7886	2243	389757	114	0.985750	0.005722	0.994278	0.014250	0.778557	0.005722	0.994108
	K30	7893	2104	389896	107	0.986625	0.005367	0.994633	0.013375	0.789537	0.005367	0.994473
	K35	7893	2104	389896	107	0.986625	0.005367	0.994633	0.013375	0.789537	0.005367	0.994473
	K40	7895	2276	389724	105	0.986875	0.005806	0.994194	0.013125	0.776227	0.005806	0.994048
	K45	7895	2276	389724	105	0.986875	0.005806	0.994194	0.013125	0.776227	0.005806	0.994048
	K50	7895	2276	389724	105	0.986875	0.005806	0.994194	0.013125	0.776227	0.005806	0.994048
	K55	7893	2276	389724	107	0.986625	0.005806	0.994194	0.013375	0.776183	0.005806	0.994043
T = 0.8	KSE-test	7889	2168	389832	111	0.986125	0.005531	0.994469	0.013875	0.784429	0.005531	0.994303
	K5	7889	1731	390269	111	0.986125	0.004416	0.995584	0.013875	0.820062	0.004416	0.995395
	K10	7886	1351	390649	114	0.985750	0.003446	0.996554	0.014250	0.853740	0.003446	0.996338
	K15	7886	1351	390649	114	0.985750	0.003446	0.996554	0.014250	0.853740	0.003446	0.996338
	K20	7886	1746	390254	114	0.985750	0.004454	0.995546	0.014250	0.818729	0.004454	0.995350
	K25	7879	2109	389891	121	0.984875	0.005380	0.994620	0.015125	0.788847	0.005380	0.994425
	K30	7886	1746	390254	114	0.985750	0.004454	0.995546	0.014250	0.818729	0.004454	0.995350
	K35	7886	1746	390254	114	0.985750	0.004454	0.995546	0.014250	0.818729	0.004454	0.995350
	K40	7888	1918	390082	112	0.986000	0.004893	0.995107	0.014000	0.804405	0.004893	0.994925
	K45	7888	1918	390082	112	0.986000	0.004893	0.995107	0.014000	0.804405	0.004893	0.994925
	K50	7888	1918	390082	112	0.986000	0.004893	0.995107	0.014000	0.804405	0.004893	0.994925
	K55	7886	1918	390082	114	0.985750	0.004893	0.995107	0.014250	0.804366	0.004893	0.994920
T = 0.85	KSE-test	7878	1928	390072	122	0.984750	0.004918	0.995082	0.015250	0.803386	0.004918	0.994875
	K5	7878	1525	390475	122	0.984750	0.003890	0.996110	0.015250	0.837818	0.003890	0.995883
	K10	7877	1337	390663	123	0.984625	0.003411	0.996589	0.015375	0.854895	0.003411	0.996350
	K15	7877	1337	390663	123	0.984625	0.003411	0.996589	0.015375	0.854895	0.003411	0.996350
	K20	7877	1698	390302	123	0.984625	0.004332	0.995668	0.015375	0.822663	0.004332	0.995448
	K25	7873	1881	390119	127	0.984125	0.004798	0.995202	0.015875	0.807156	0.004798	0.994980
	K30	7877	1698	390302	123	0.984625	0.004332	0.995668	0.015375	0.822663	0.004332	0.995448
	K35	7877	1698	390302	123	0.984625	0.004332	0.995668	0.015375	0.822663	0.004332	0.995448
	K40	7877	1870	390130	123	0.984625	0.004770	0.995230	0.015375	0.808146	0.004770	0.995018
	K45	7877	1870	390130	123	0.984625	0.004770	0.995230	0.015375	0.808146	0.004770	0.995018
	K50	7877	1870	390130	123	0.984625	0.004770	0.995230	0.015375	0.808146	0.004770	0.995018
	K55	7877	1870	390130	123	0.984625	0.004770	0.995230	0.015375	0.808146	0.004770	0.995018
T = 0.9	KSE-test	7867	1697	390303	133	0.983375	0.004329	0.995671	0.016625	0.822564	0.004329	0.995425
	K5	7867	1508	390492	133	0.983375	0.003847	0.996153	0.016625	0.839147	0.003847	0.995898
	K10	7867	1320	390680	133	0.983375	0.003367	0.996633	0.016625	0.856319	0.003367	0.996368
	K15	7867	1320	390680	133	0.983375	0.003367	0.996633	0.016625	0.856319	0.003367	0.996368
	K20	7867	1508	390492	133	0.983375	0.003847	0.996153	0.016625	0.839147	0.003847	0.995898
	K25	7865	1659	390341	135	0.983125	0.004232	0.995768	0.016875	0.825808	0.004232	0.995515
	K30	7867	1508	390492	133	0.983375	0.003847	0.996153	0.016625	0.839147	0.003847	0.995898
	K35	7867	1508	390492	133	0.983375	0.003847	0.996153	0.016625	0.839147	0.003847	0.995898
	K40	7867	1680	390320	133	0.983375	0.004286	0.995714	0.016625	0.824028	0.004286	0.995468
	K45	7867	1680	390320	133	0.983375	0.004286	0.995714	0.016625	0.824028	0.004286	0.995468
	K50	7867	1680	390320	133	0.983375	0.004286	0.995714	0.016625	0.824028	0.004286	0.995468
	K55	7867	1680	390320	133	0.983375	0.004286	0.995714	0.016625	0.824028	0.004286	0.995468
T = 0.95	KSE-test	7779	1293	390707	221	0.972375	0.003298	0.996702	0.027625	0.857474	0.003298	0.996215
	K5	7779	1293	390707	221	0.972375	0.003298	0.996702	0.027625	0.857474	0.003298	0.996215
	K10	7779	1293	390707	221	0.972375	0.003298	0.996702	0.027625	0.857474	0.003298	0.996215
	K15	7779	1293	390707	221	0.972375	0.003298	0.996702	0.027625	0.857474	0.003298	0.996215
	K20	7779	1293	390707	221	0.972375	0.003298	0.996702	0.027625	0.857474	0.003298	0.996215
	K25	7779	1293	390707	221	0.972375	0.003298	0.996702	0.027625	0.857474	0.003298	0.996215
	K30	7779	1293	390707	221	0.972375	0.003298	0.996702	0.027625	0.857474	0.003298	0.996215
	K35	7779	1293	390707	221	0.972375	0.003298	0.996702	0.027625	0.857474	0.003298	0.996215
	K40	7779	1293	390707	221	0.972375	0.003298	0.996702	0.027625	0.857474	0.003298	0.996215
	K45	7779	1293	390707	221	0.972375	0.003298	0.996702	0.027625	0.857474	0.003298	0.996215
	K50	7779	1293	390707	221	0.972375	0.003298	0.996702	0.027625	0.857474	0.003298	0.996215
	K55	7779	1293	390707	221	0.972375	0.003298	0.996702	0.027625	0.857474	0.003298	0.996215

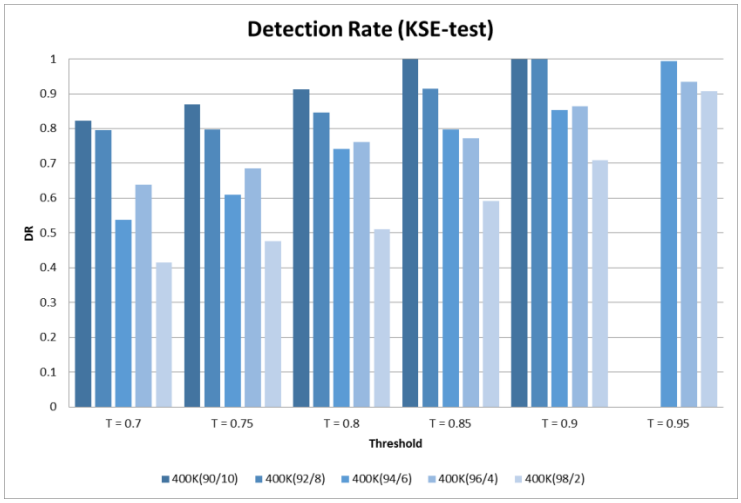
แสดงผลคุณภาพของการจำแนกโดยใช้แผนภูมิ ROC (Receiver operating characteristic) [31] และ แผนภูมิของ Detection Rate, False Positive Rate และ Accuracy เพื่อแสดงความสัมพันธ์และแนวโน้มของแต่ละการทดลองที่ผลลัพธ์จาก KSE-test และ การทดลองที่ค่า K ต่างๆได้ดังต่อไปนี้



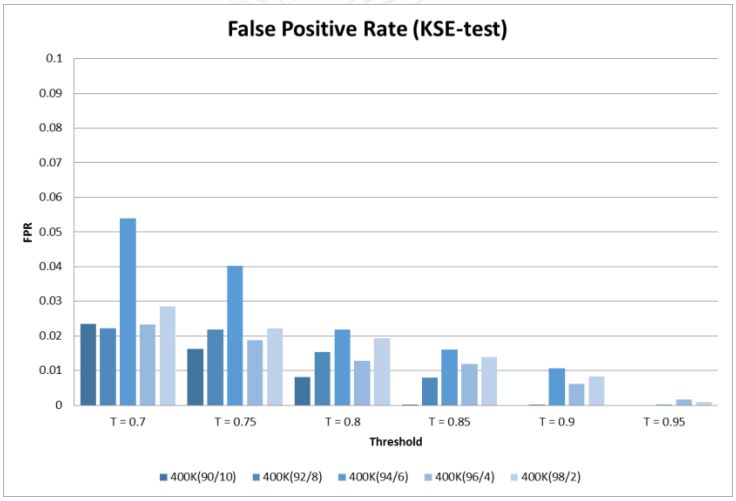
ภาพที่ 33 ROC ระหว่าง TPR และ FPR ของ KSE-test



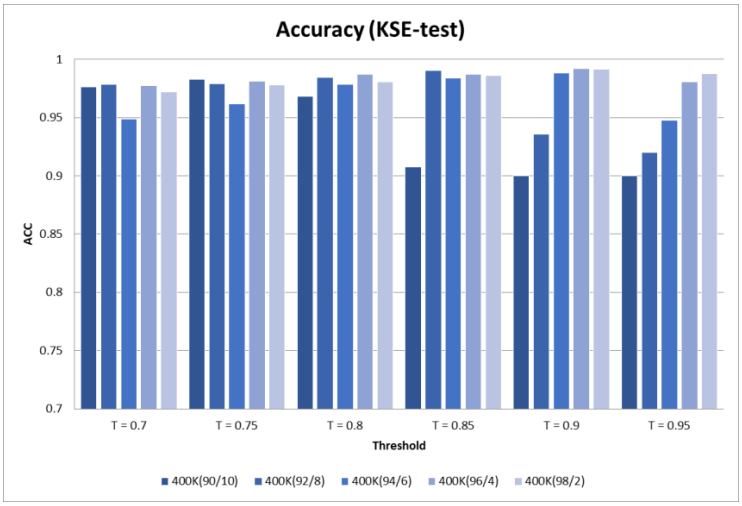
ภาพที่ 34 ROC ระหว่าง TNR และ FNR ของ KSE-test



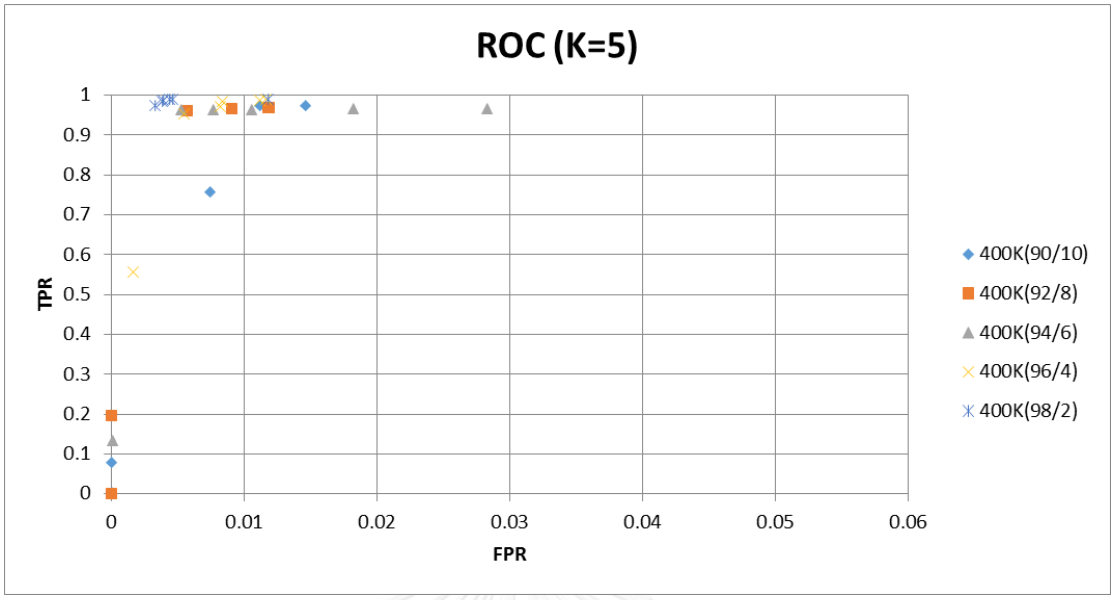
ภาพที่ 35 ผลการทดลอง Detection Rate ของ KSE-test



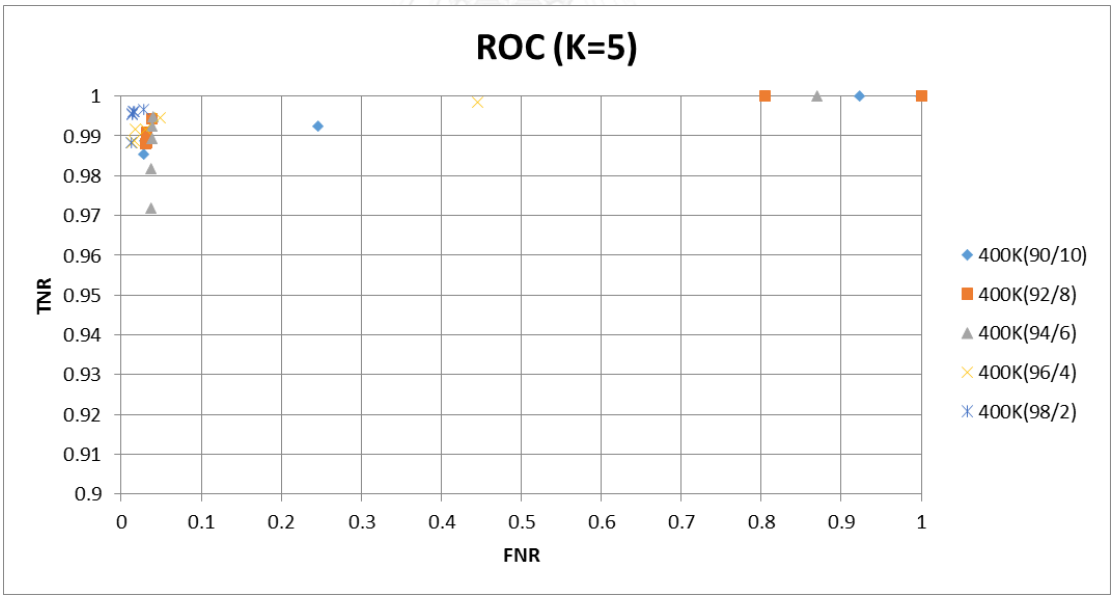
ภาพที่ 36 ผลการทดลอง False Positive Rate ของ KSE-test



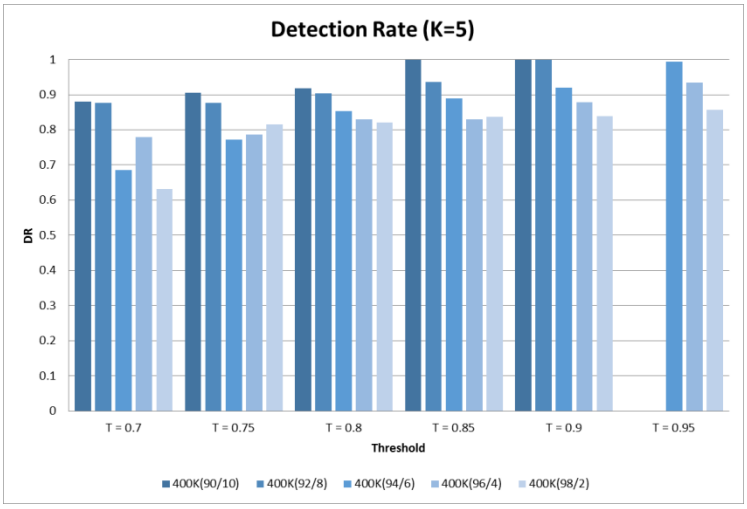
ภาพที่ 37 ผลการทดลอง Accuracy ของ KSE-test



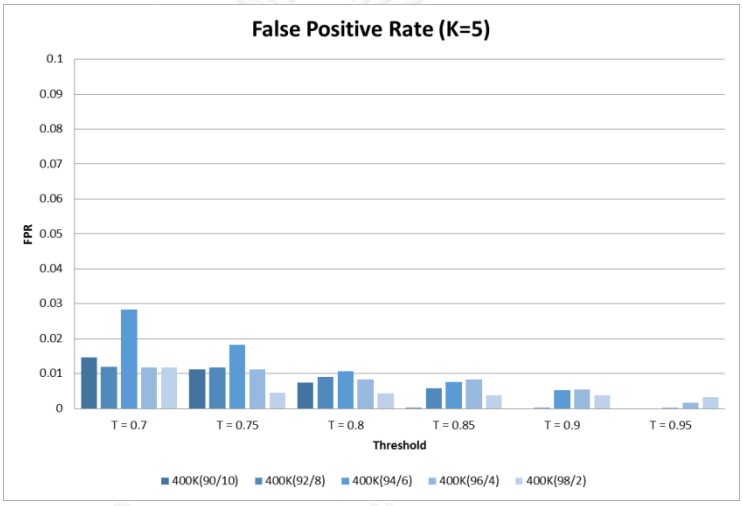
ภาพที่ 38 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=5)



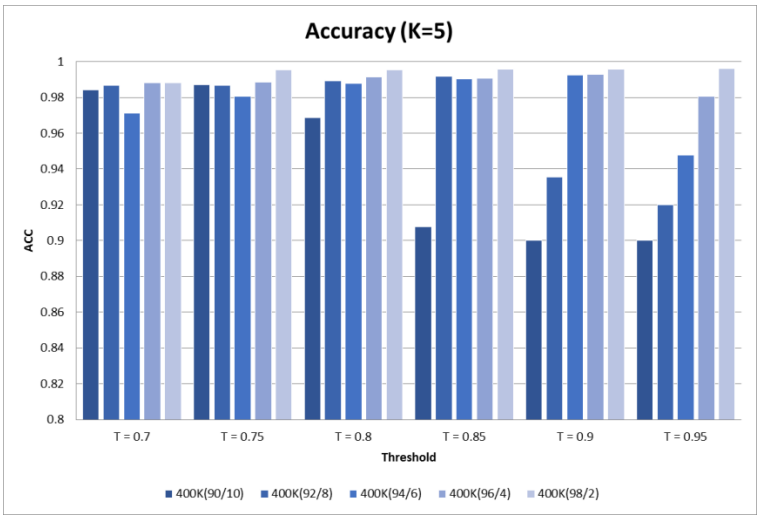
ภาพที่ 39 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=5)



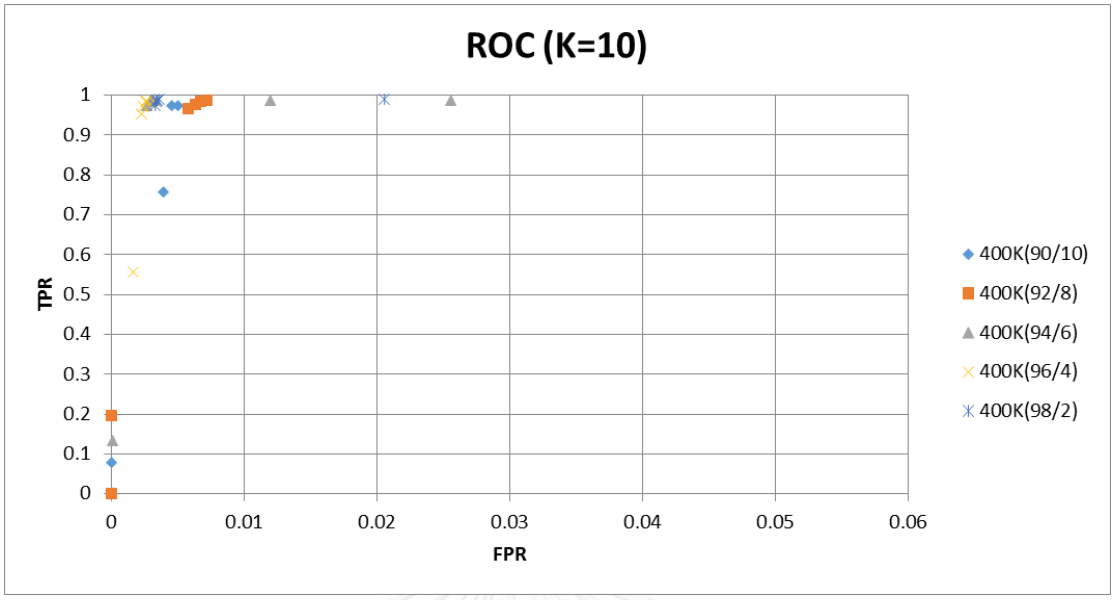
ภาพที่ 40 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=5)



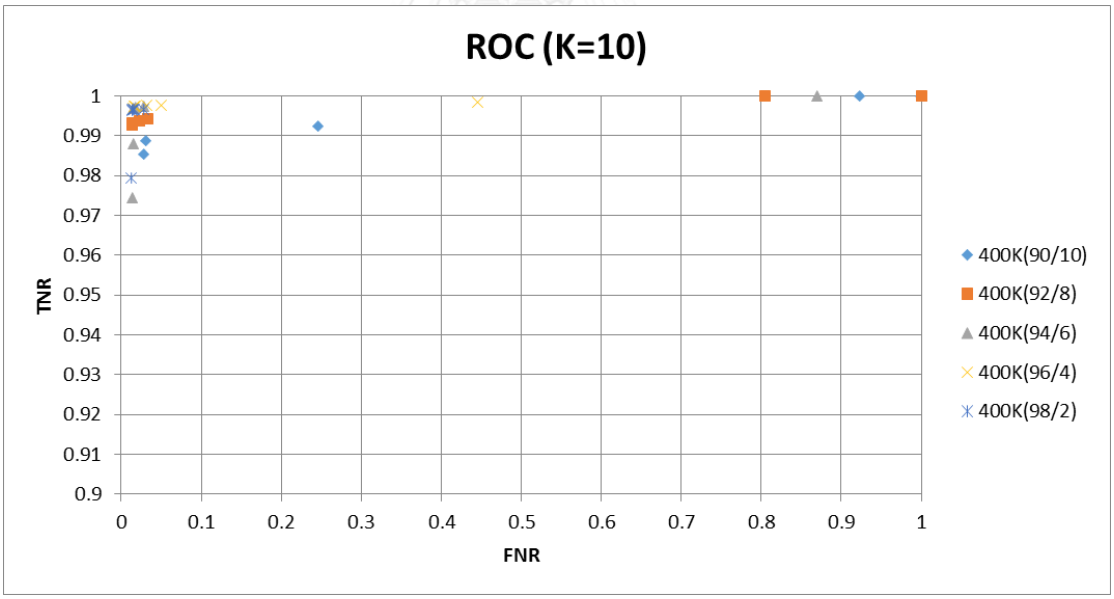
ภาพที่ 41 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=5)



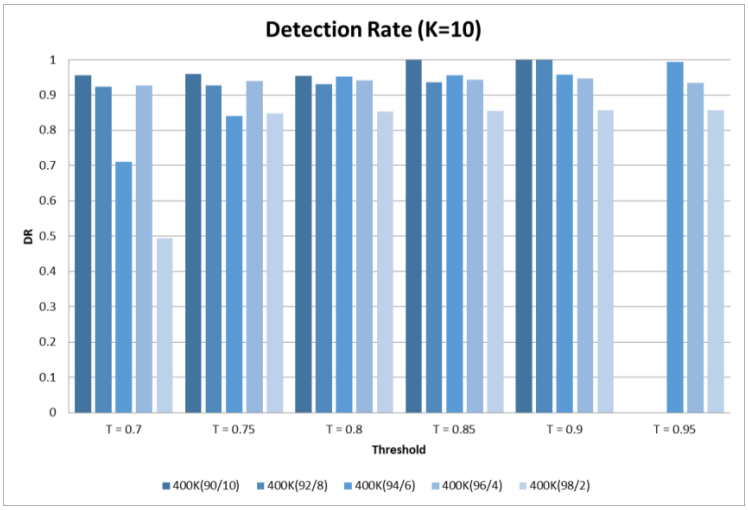
ภาพที่ 42 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=5)



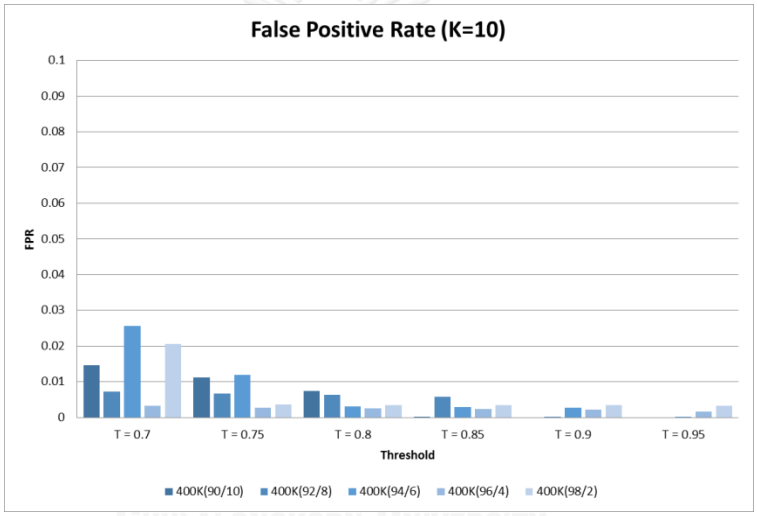
ภาพที่ 43 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=10)



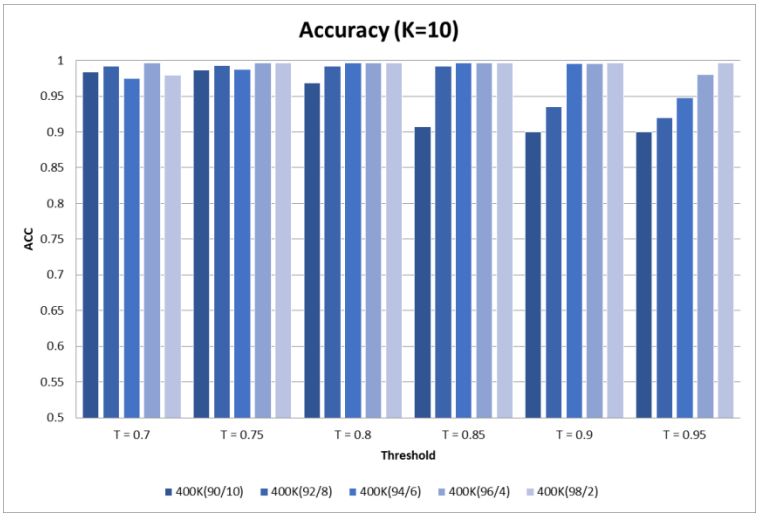
ภาพที่ 44 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=10)



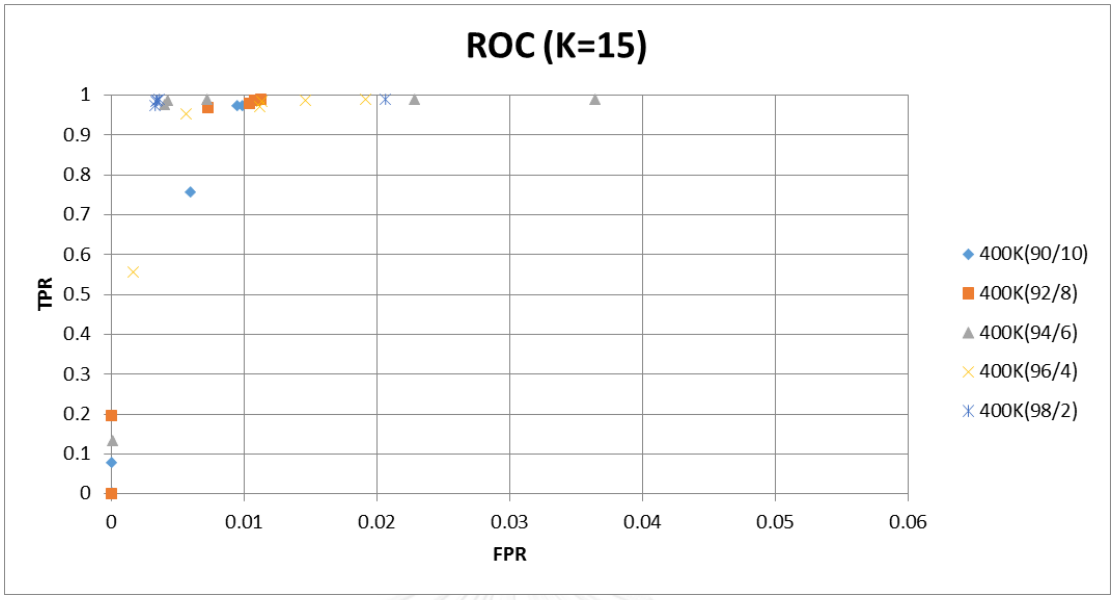
ภาพที่ 45 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=10)



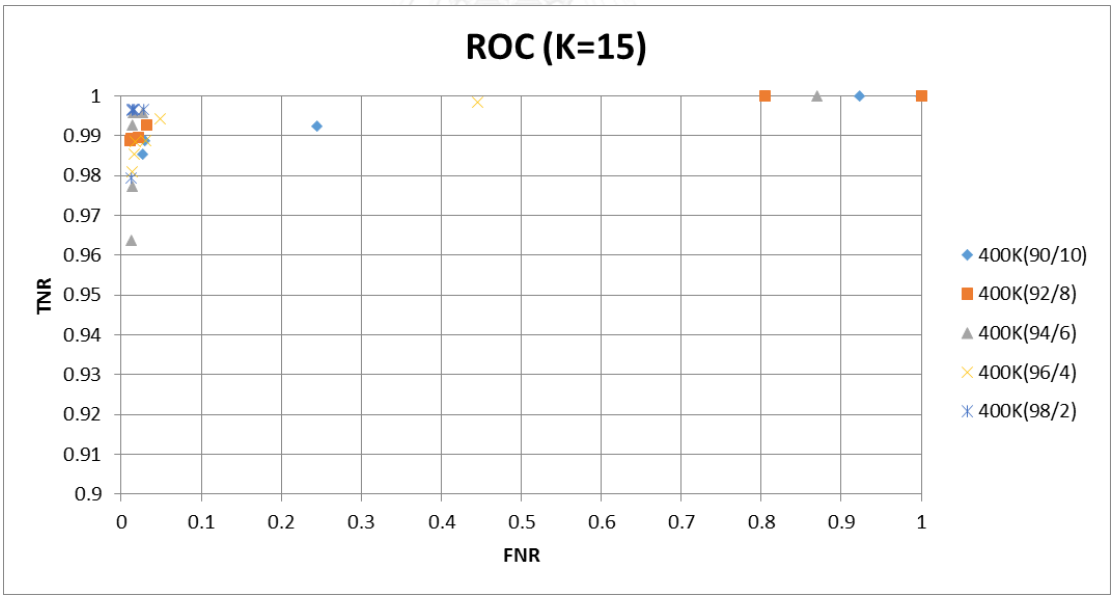
ภาพที่ 46 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=10)



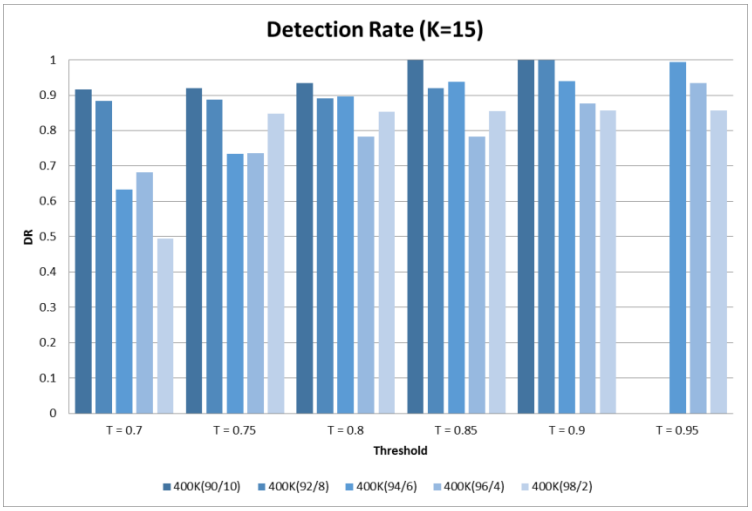
ภาพที่ 47 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=10)



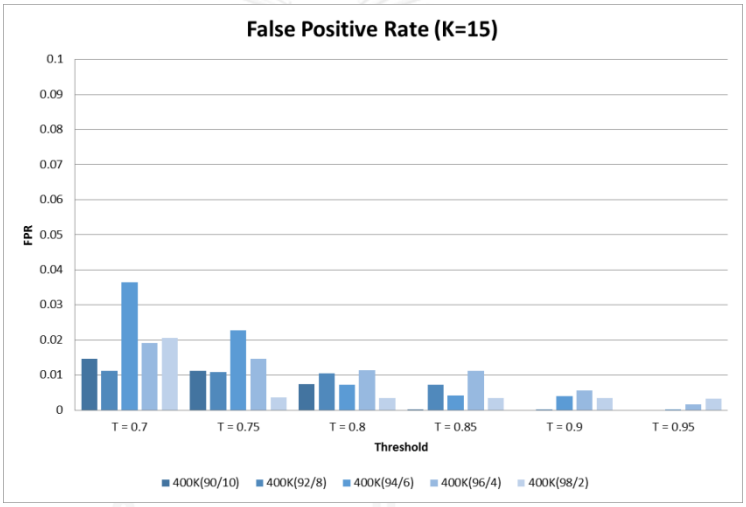
ภาพที่ 48 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=15)



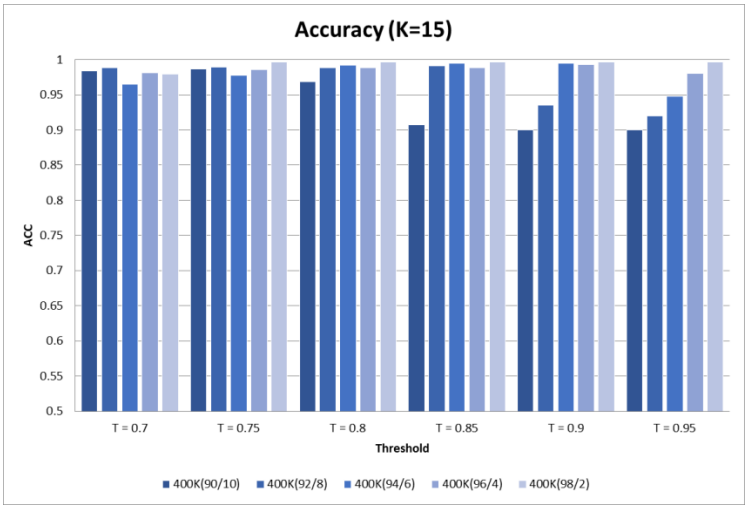
ภาพที่ 49 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=15)



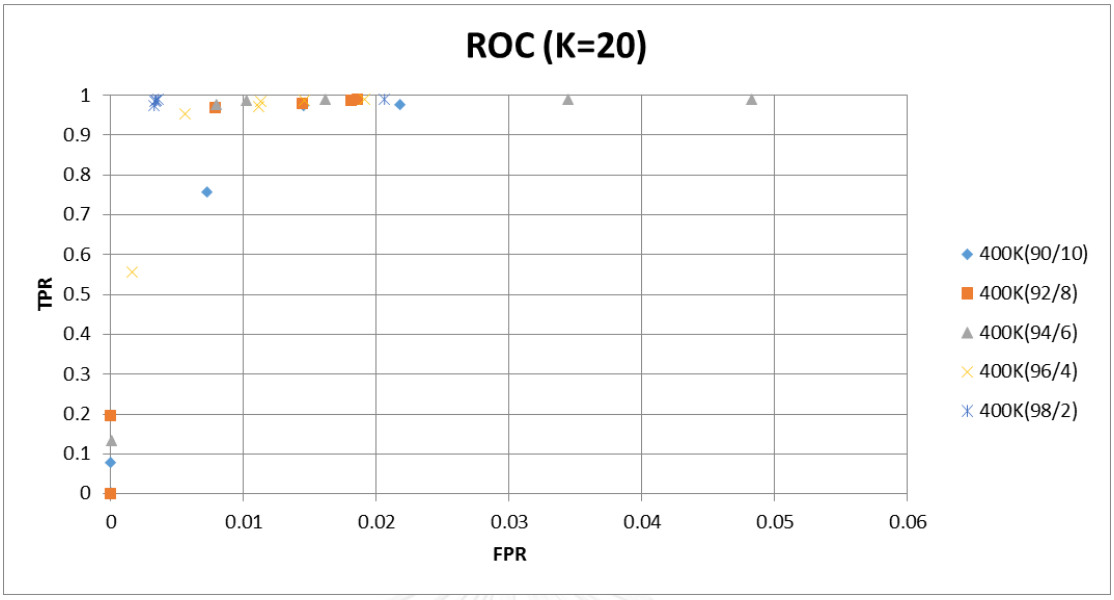
ภาพที่ 50 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=15)



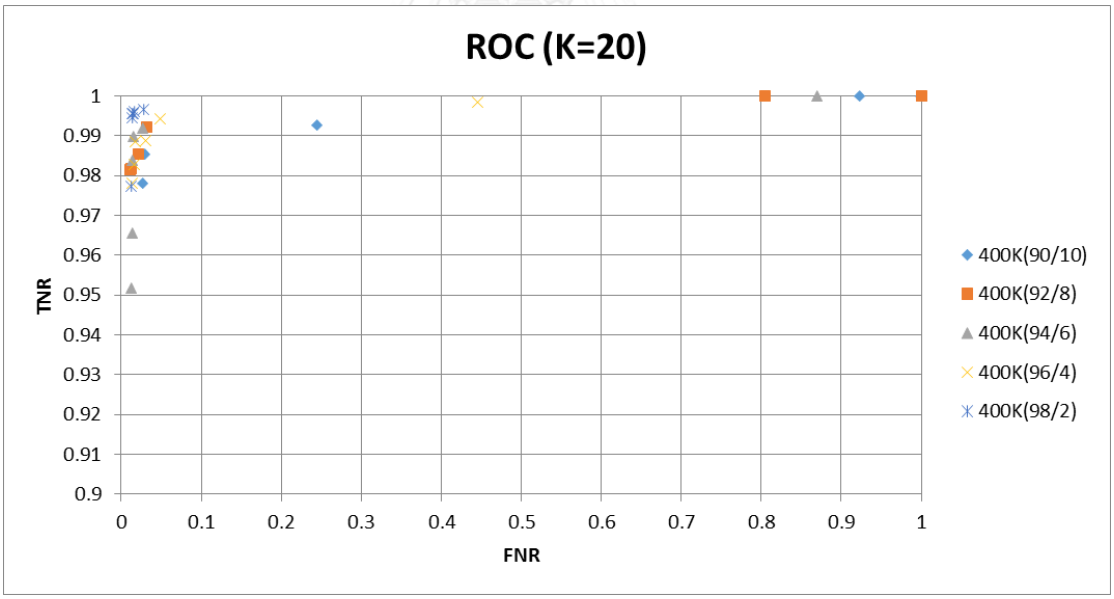
ภาพที่ 51 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=15)



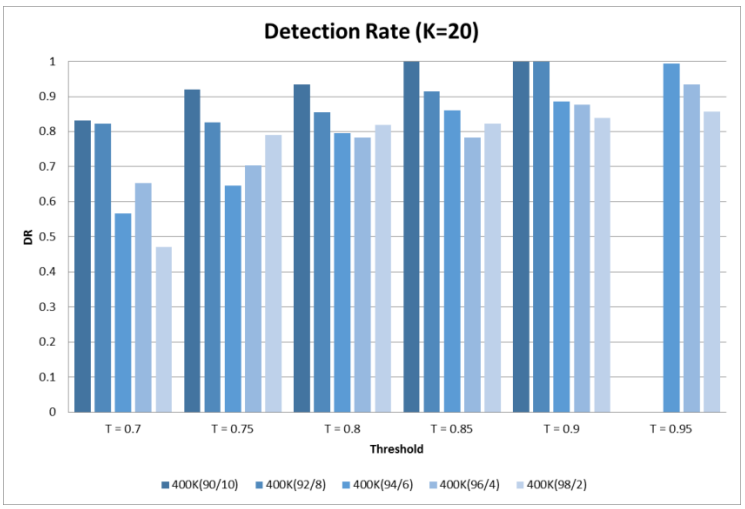
ภาพที่ 52 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=15)



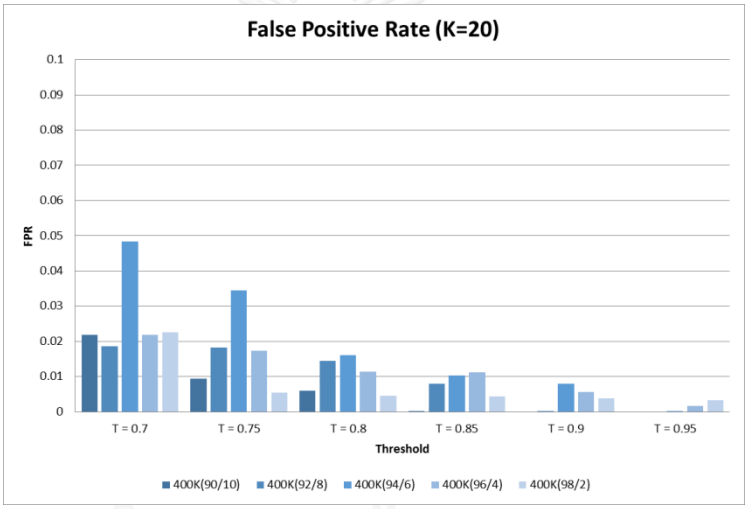
ภาพที่ 53 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=20)



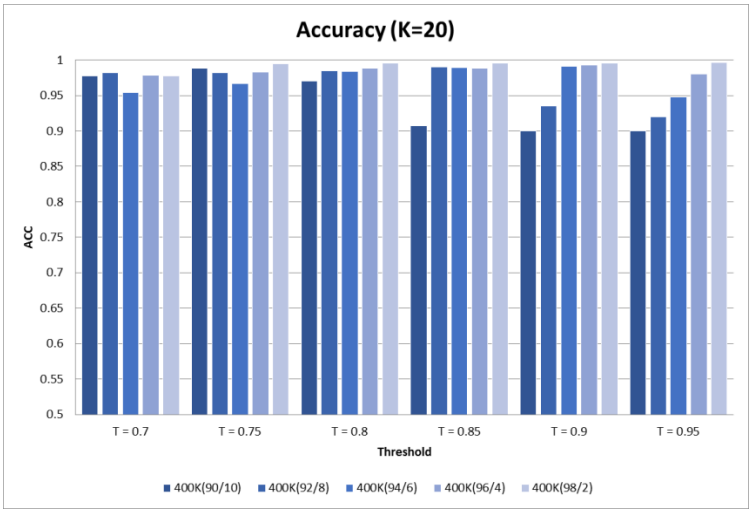
ภาพที่ 54 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=20)



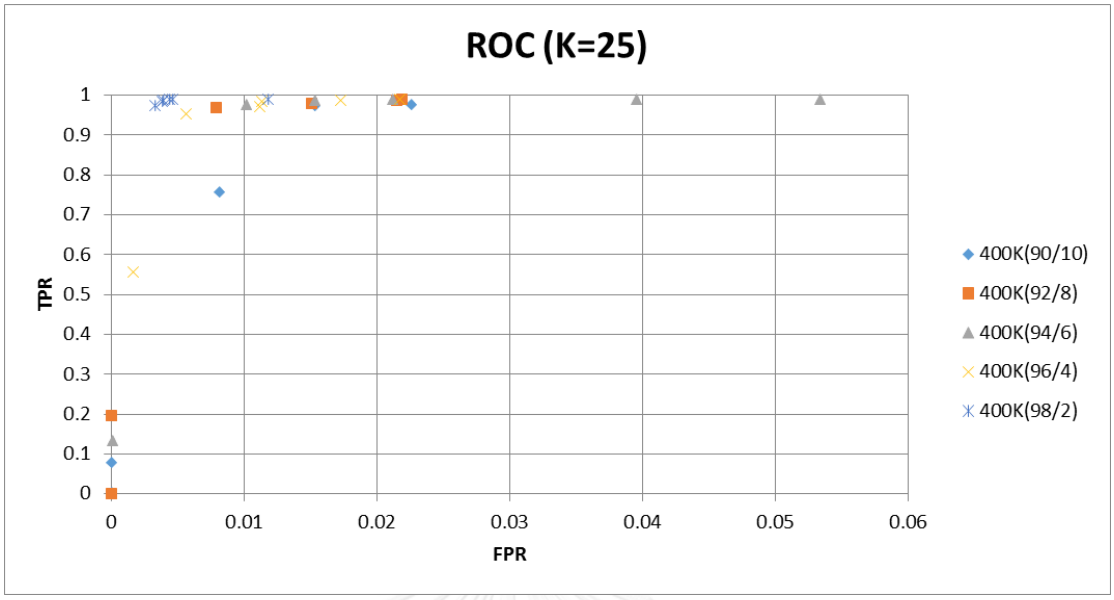
ภาพที่ 55 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=20)



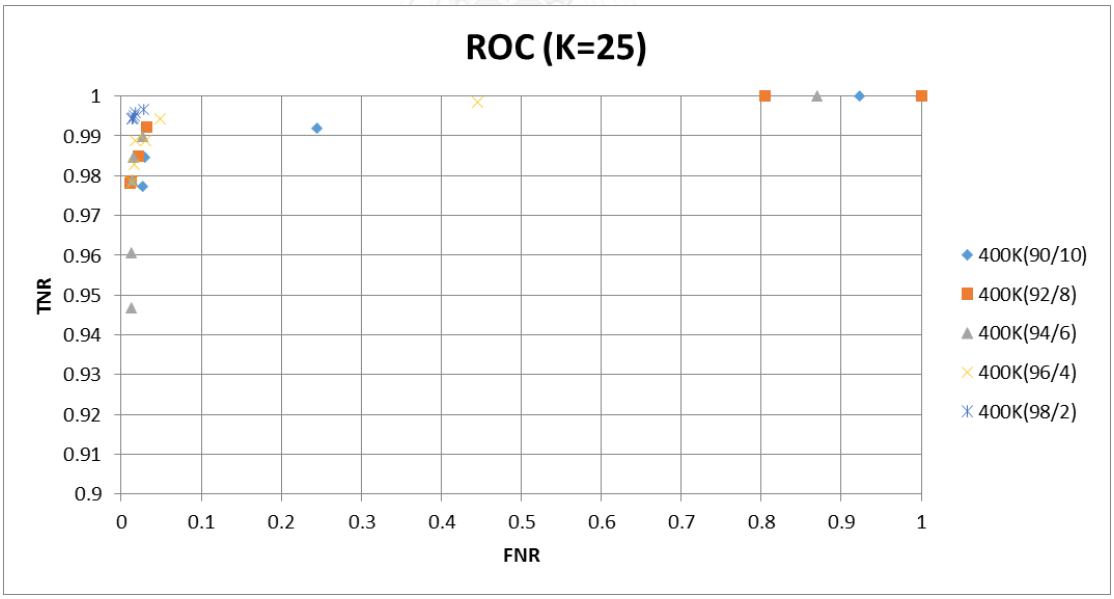
ภาพที่ 56 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=20)



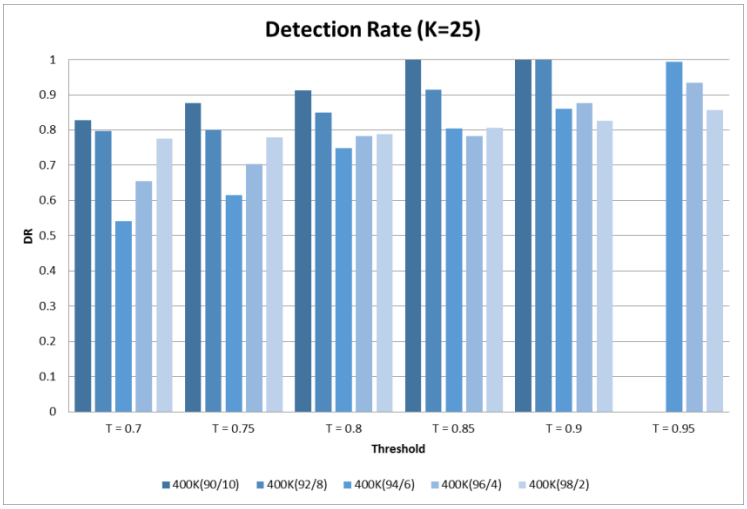
ภาพที่ 57 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=20)



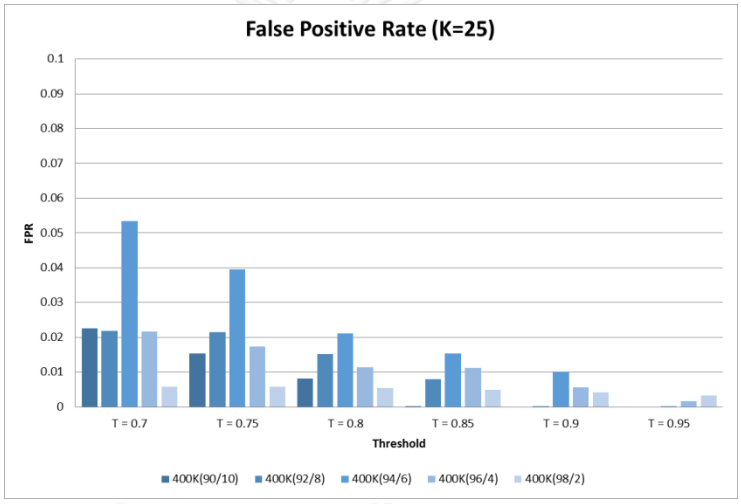
ภาพที่ 58 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=25)



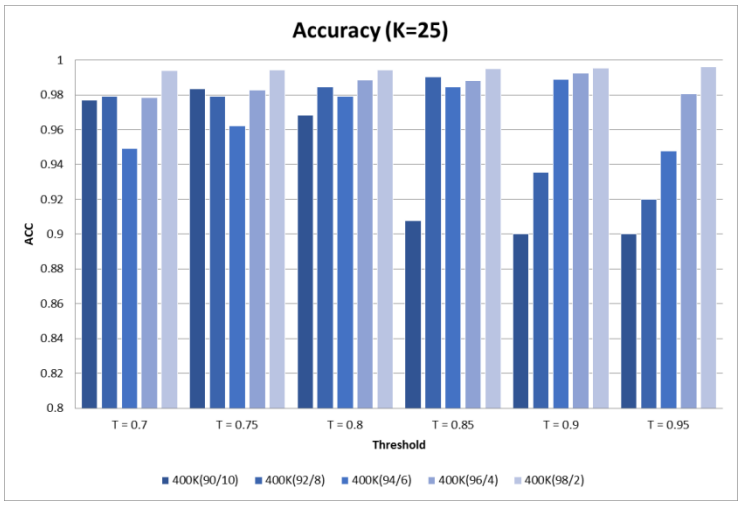
ภาพที่ 59 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=25)



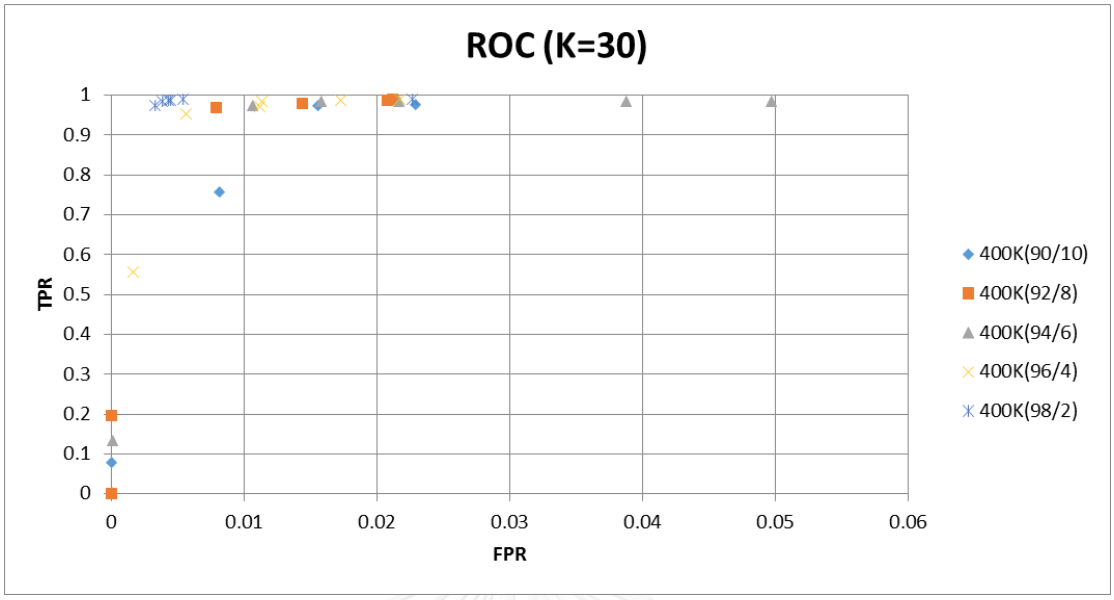
ภาพที่ 60 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=25)



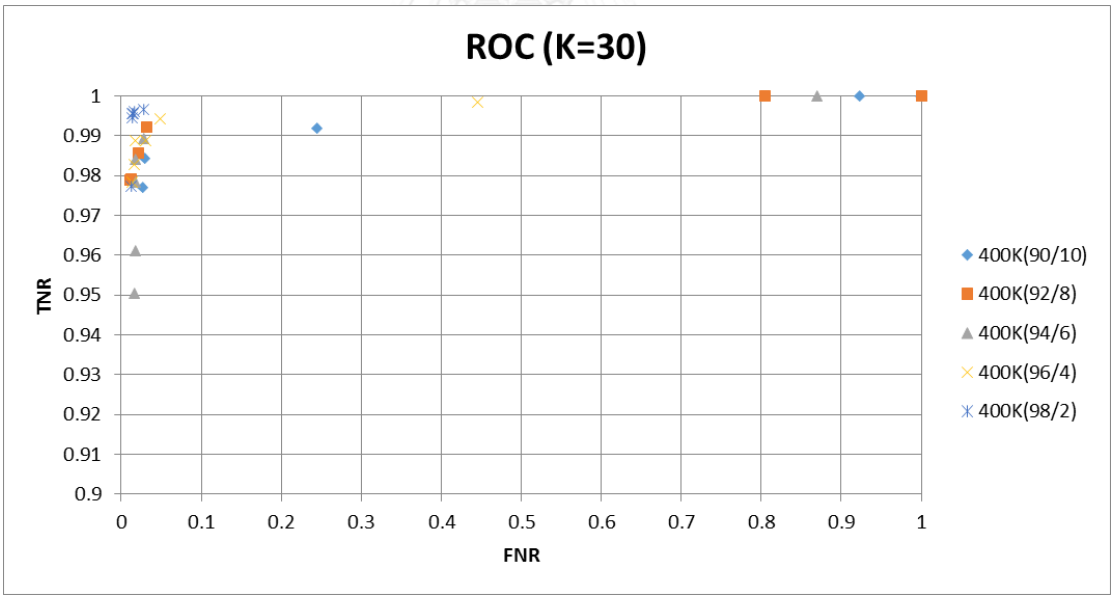
ภาพที่ 61 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=25)



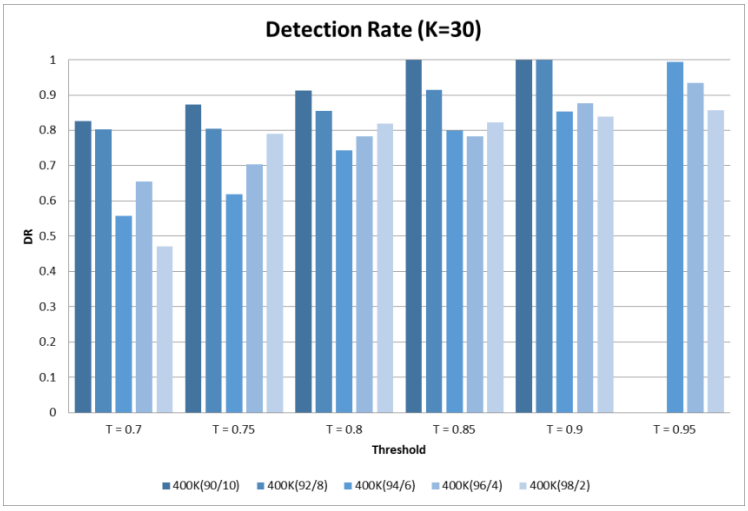
ภาพที่ 62 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=25)



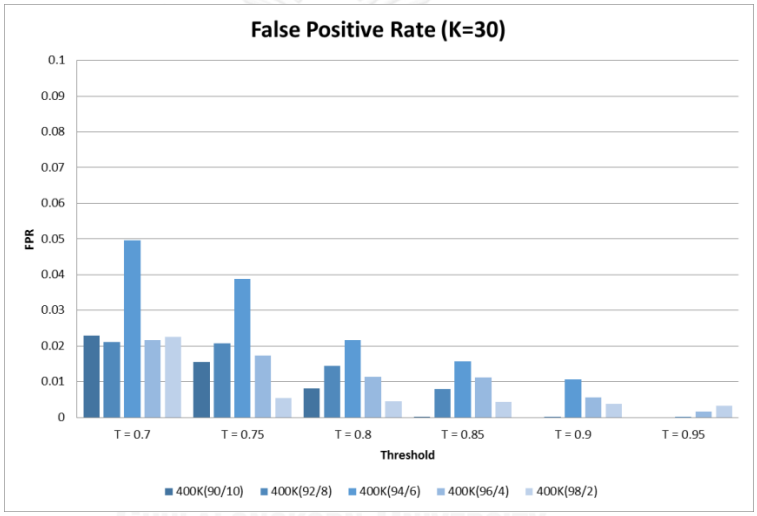
ภาพที่ 63 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=30)



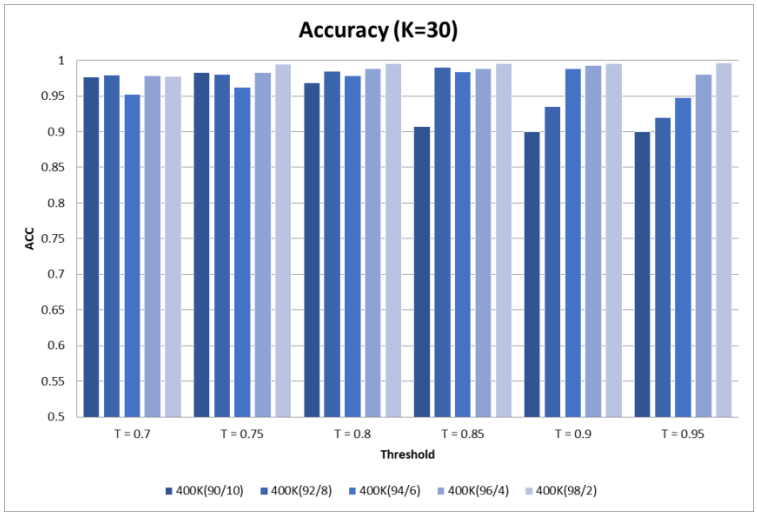
ภาพที่ 64 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=30)



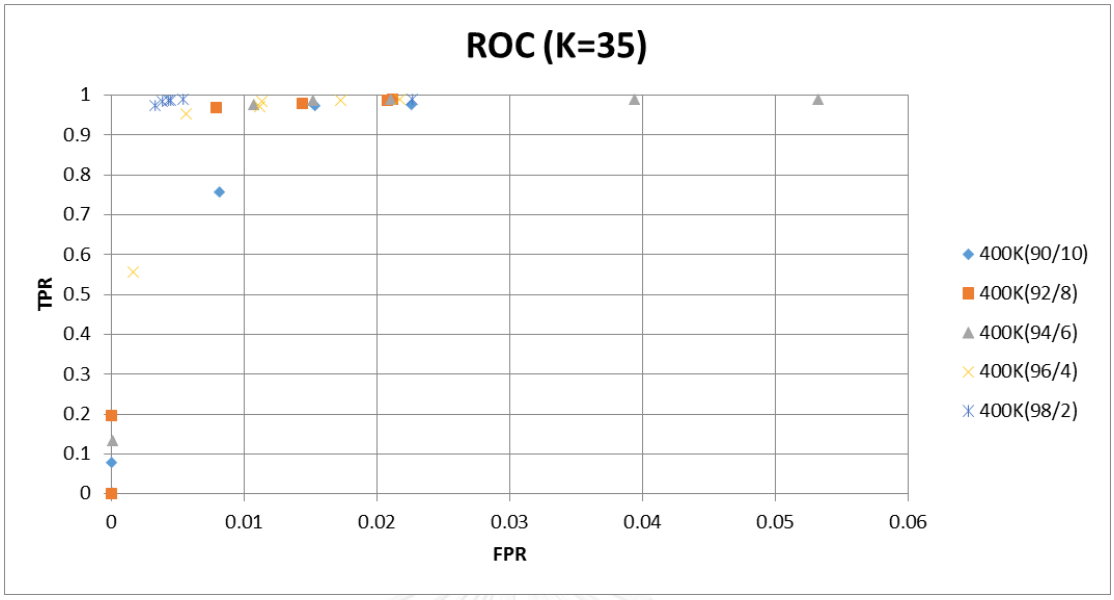
ภาพที่ 65 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=30)



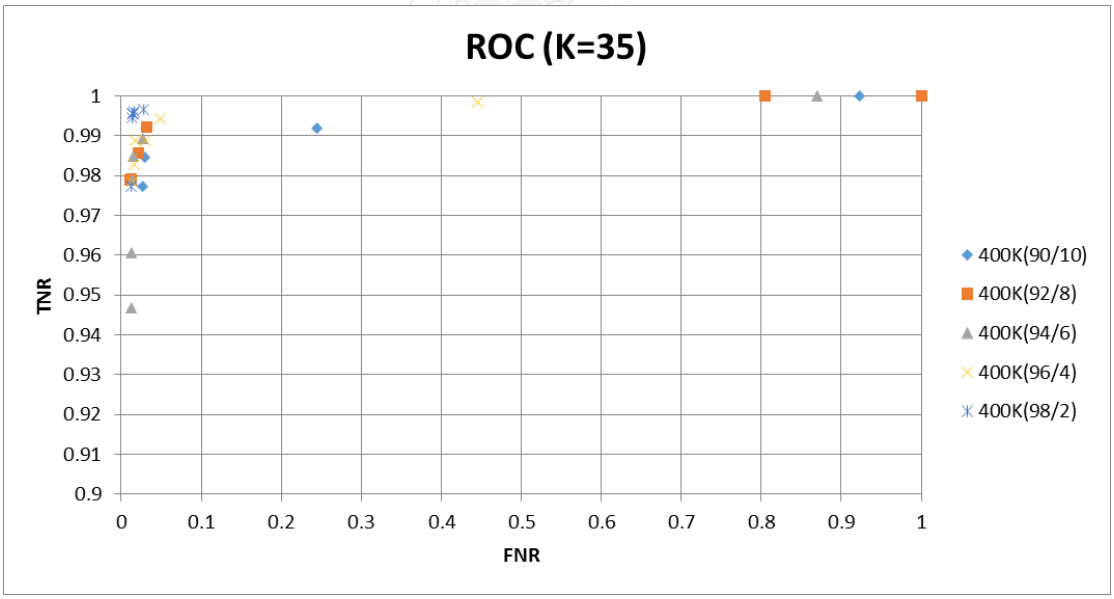
ภาพที่ 66 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=30)



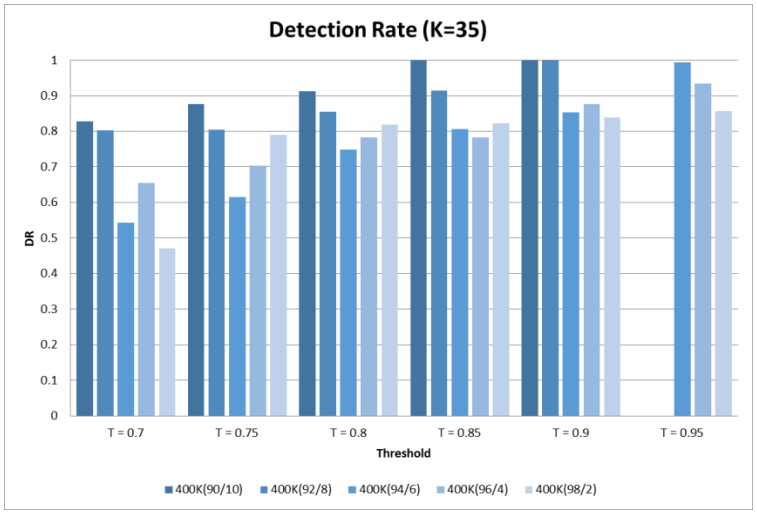
ภาพที่ 67 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=30)



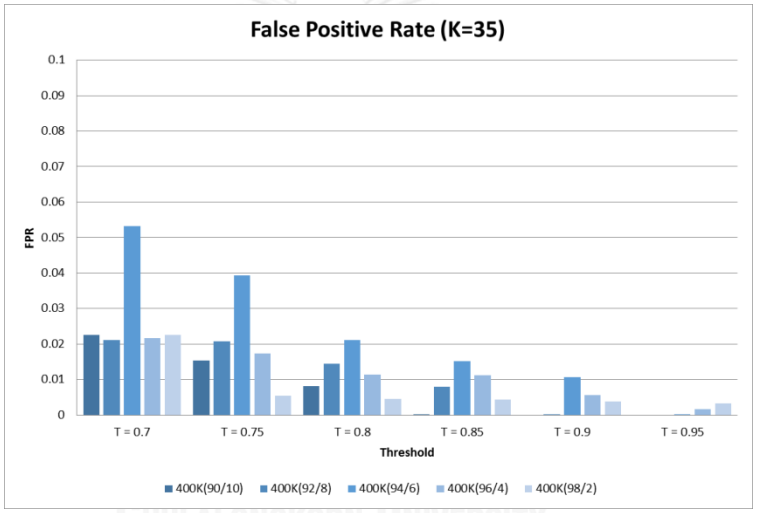
ภาพที่ 68 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=35)



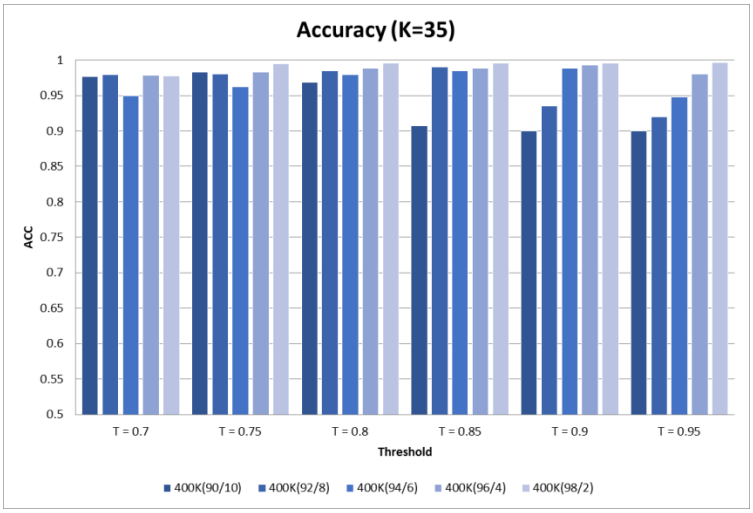
ภาพที่ 69 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=35)



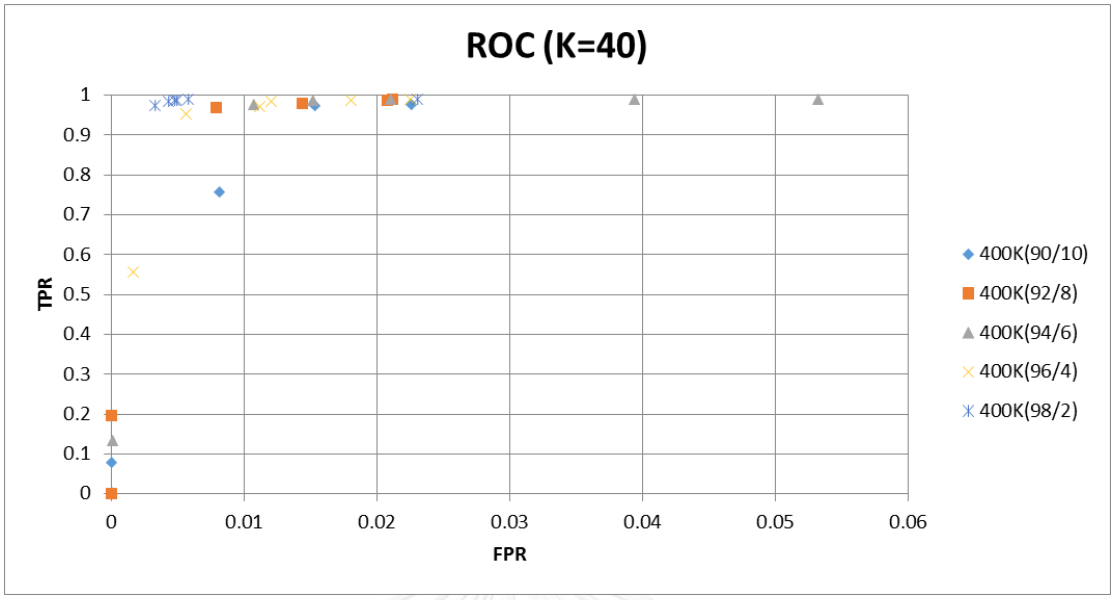
ภาพที่ 70 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=35)



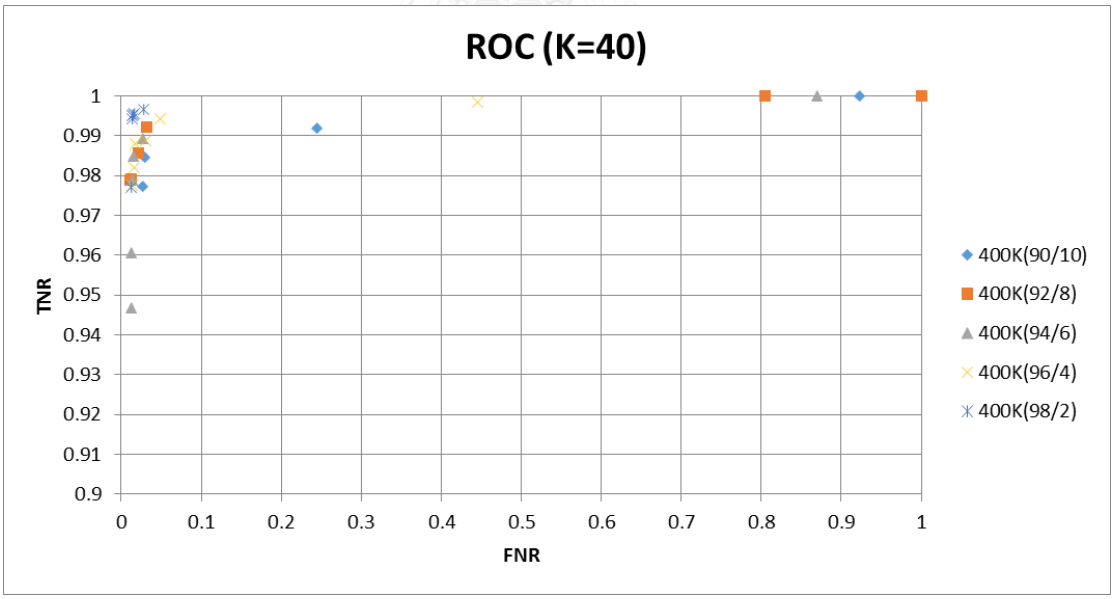
ภาพที่ 71 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=35)



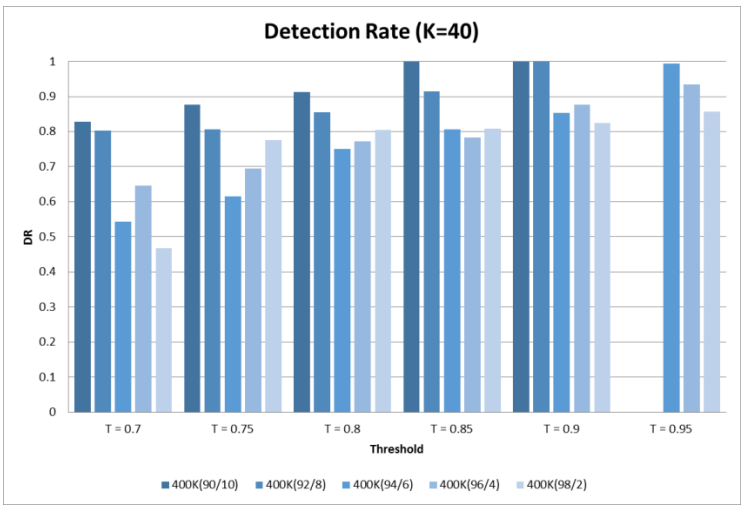
ภาพที่ 72 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=35)



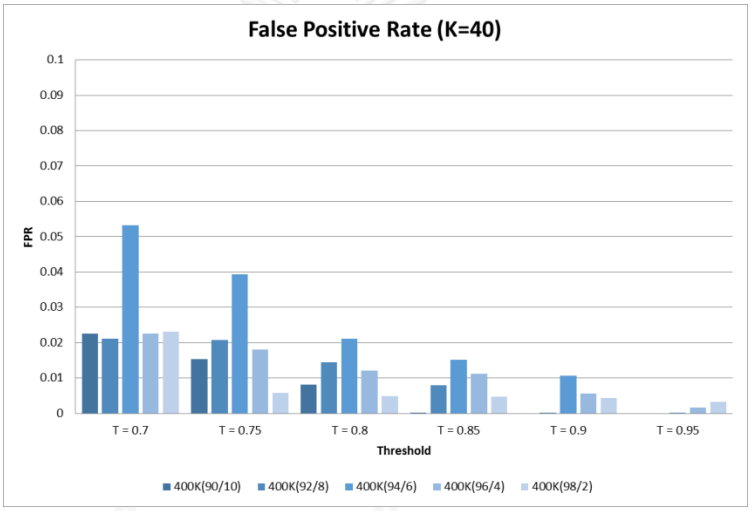
ภาพที่ 73 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=40)



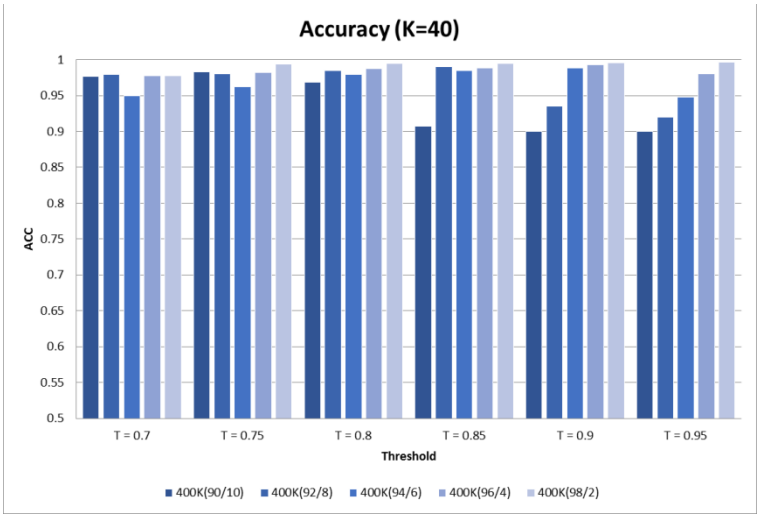
ภาพที่ 74 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=40)



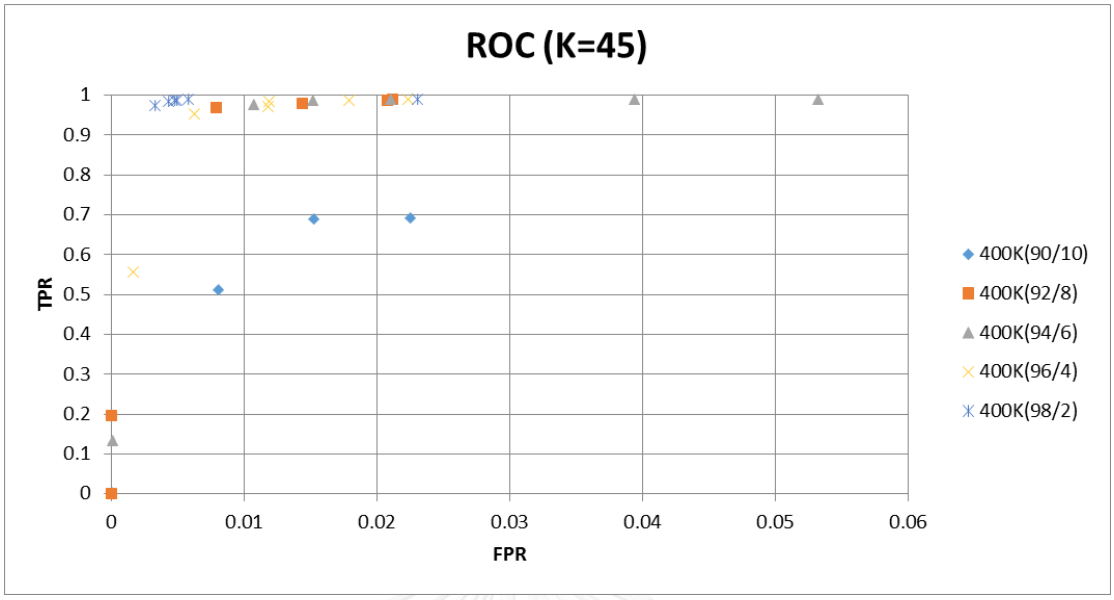
ภาพที่ 75 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=40)



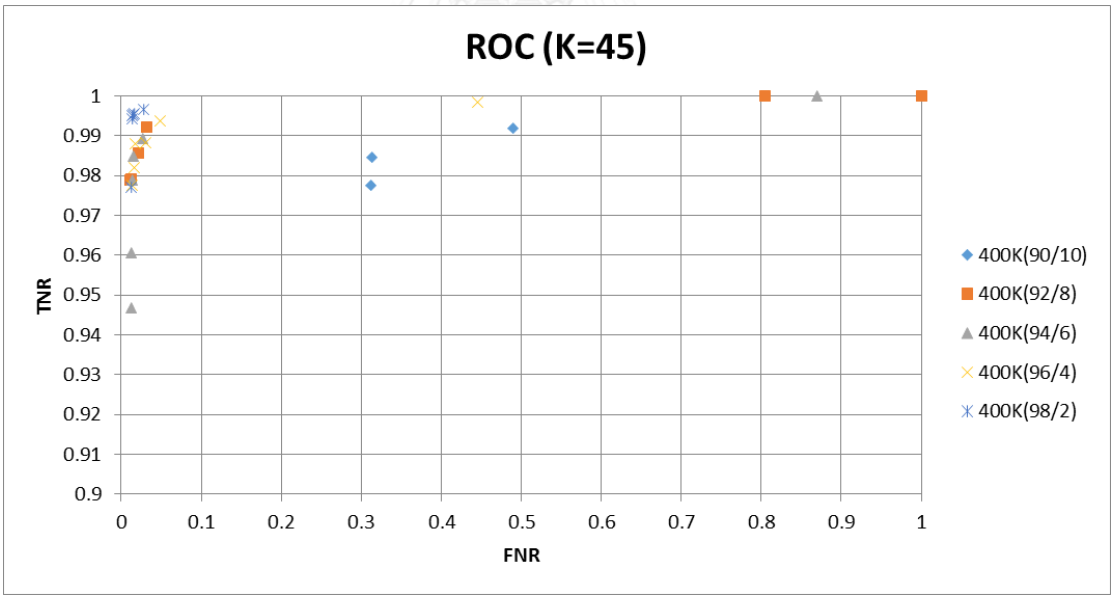
ภาพที่ 76 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=40)



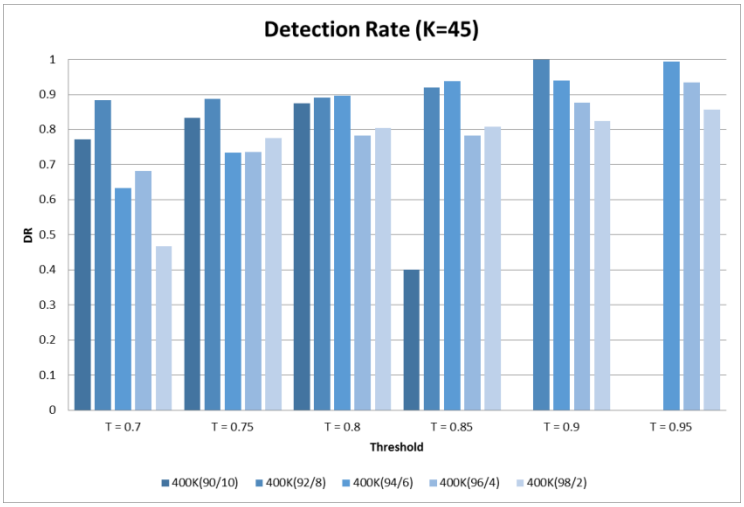
ภาพที่ 77 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=40)



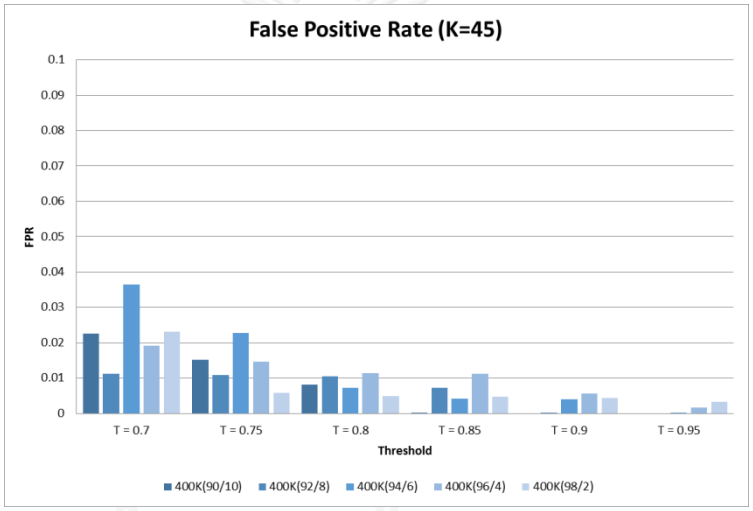
ภาพที่ 78 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=45)



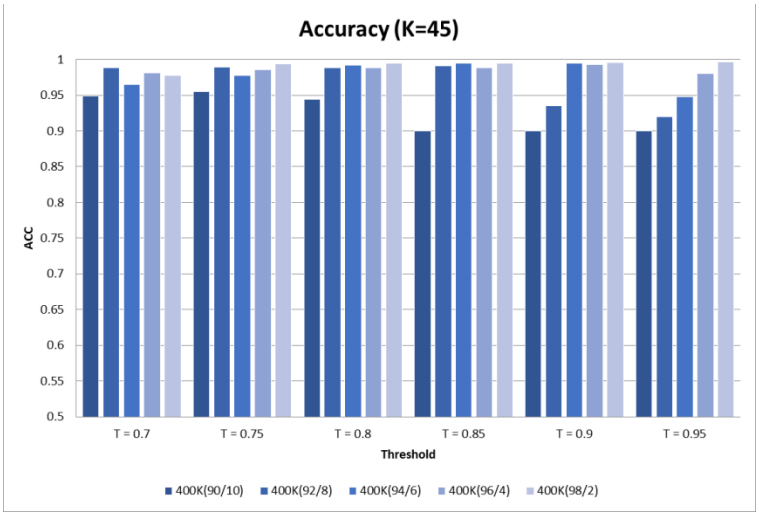
ภาพที่ 79 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=45)



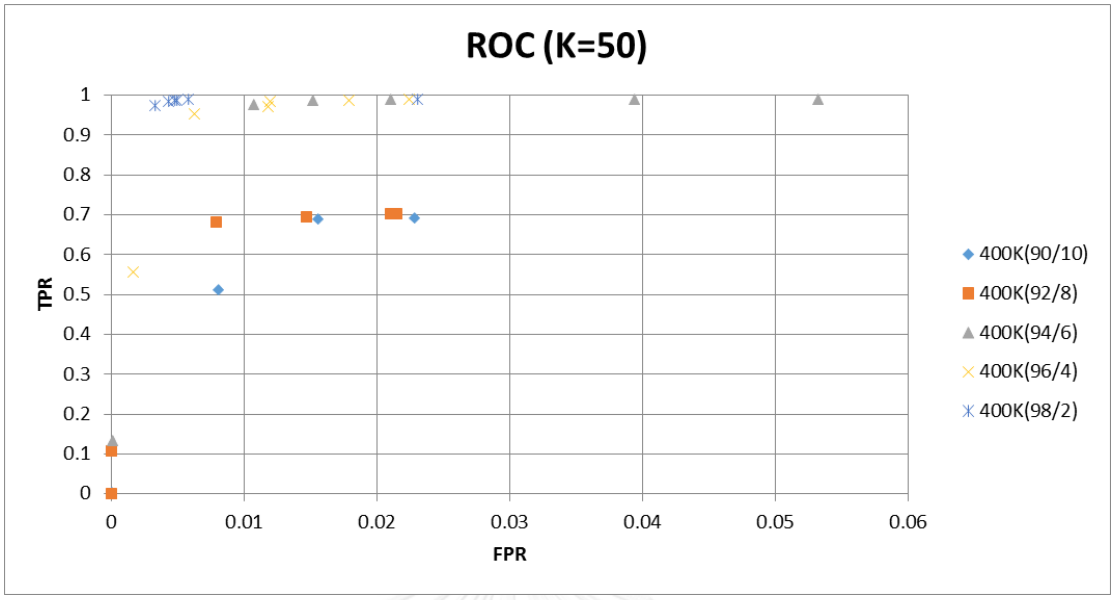
ภาพที่ 80 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=45)



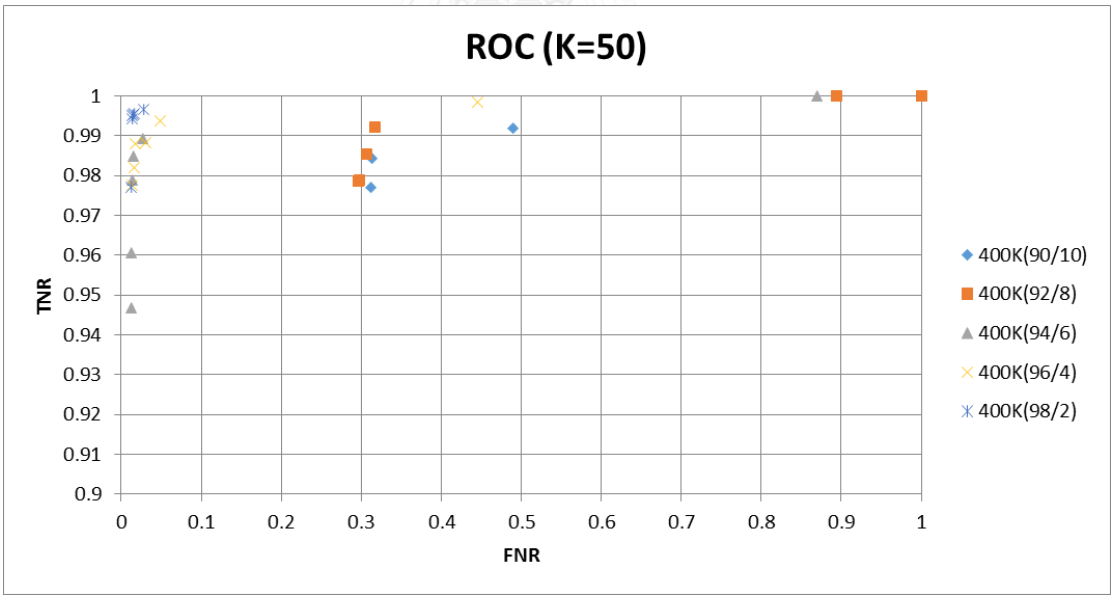
ภาพที่ 81 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=45)



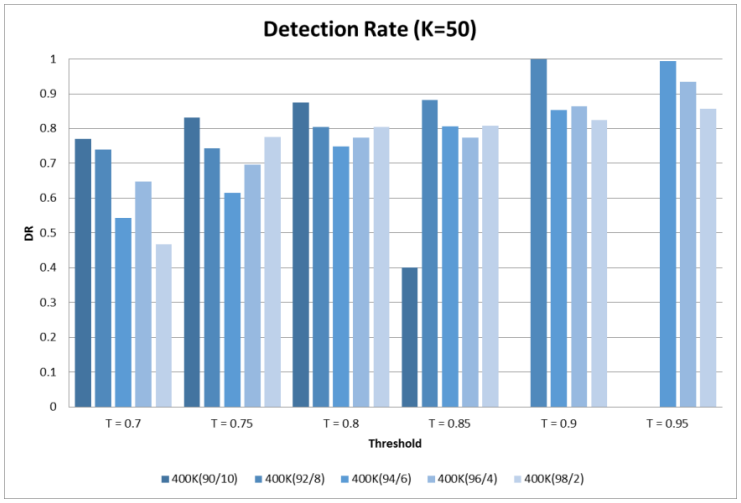
ภาพที่ 82 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=45)



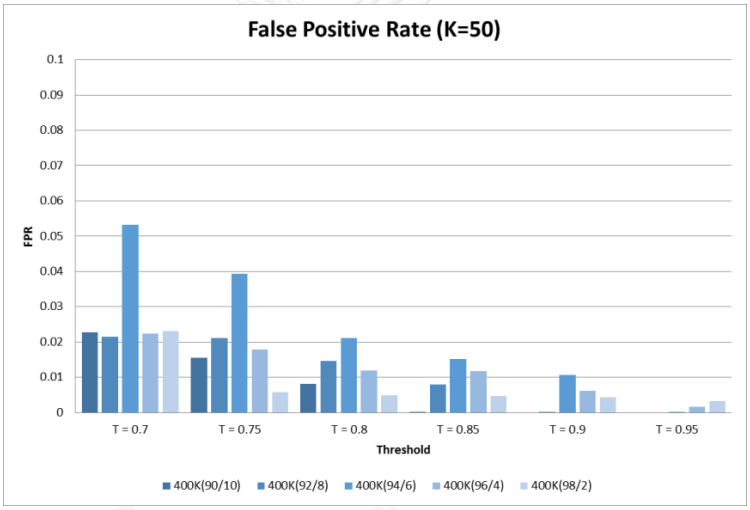
ภาพที่ 83 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=50)



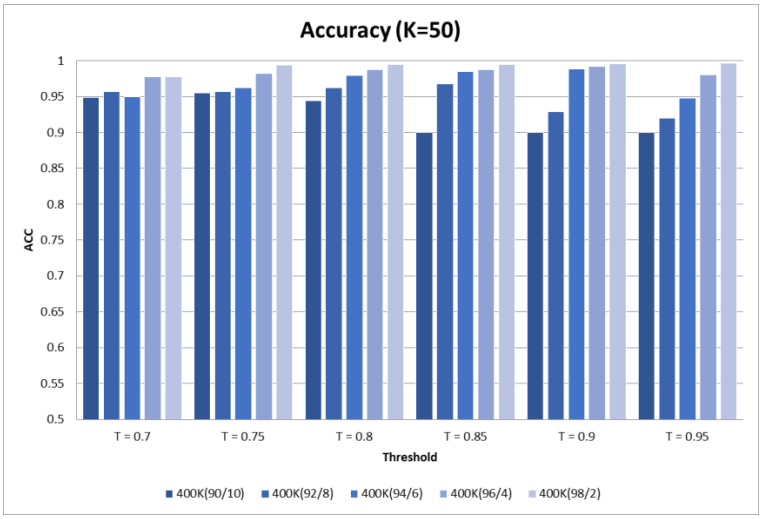
ภาพที่ 84 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=50)



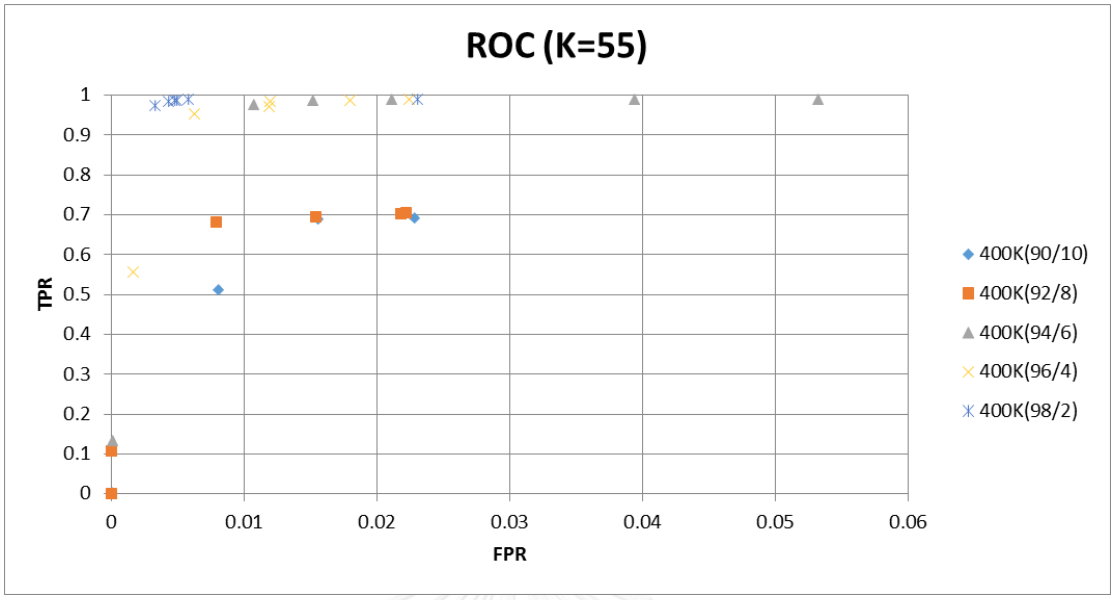
ภาพที่ 85 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=50)



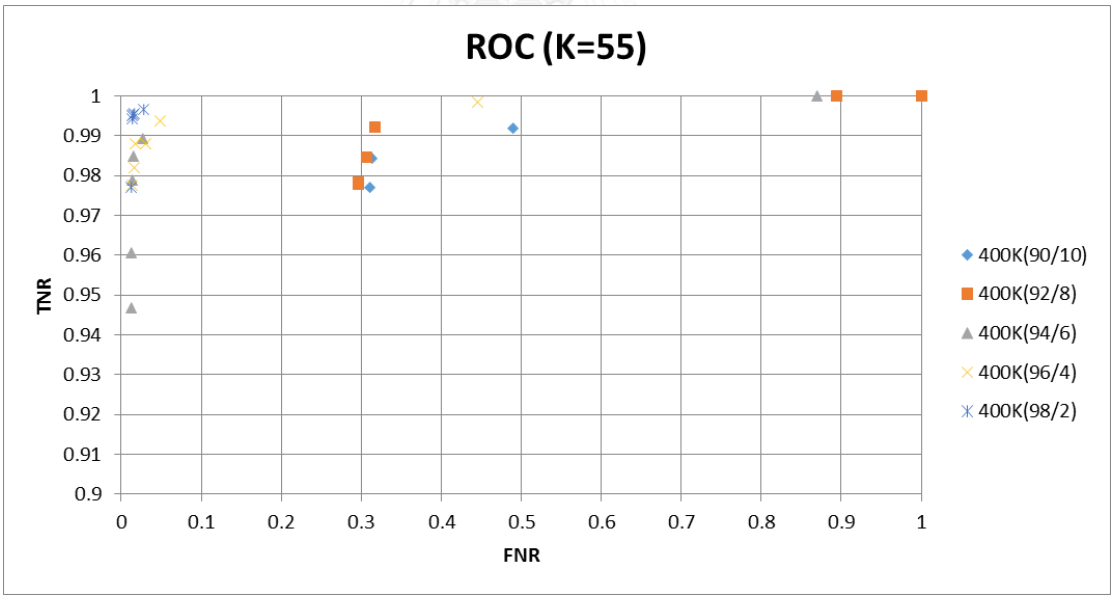
ภาพที่ 86 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=50)



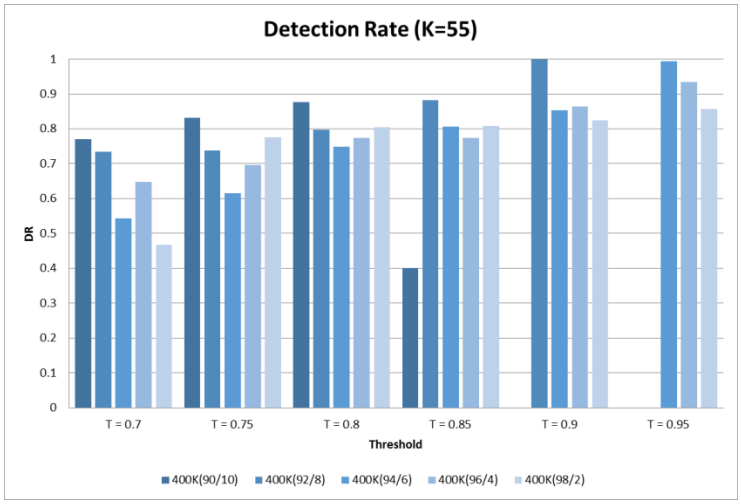
ภาพที่ 87 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=50)



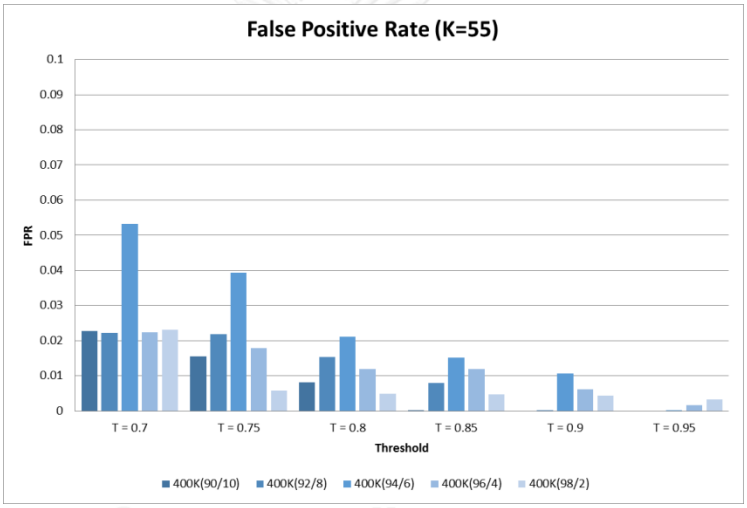
ภาพที่ 88 ROC ระหว่าง TPR และ FPR ของ วิธีการที่นำเสนอ (K=55)



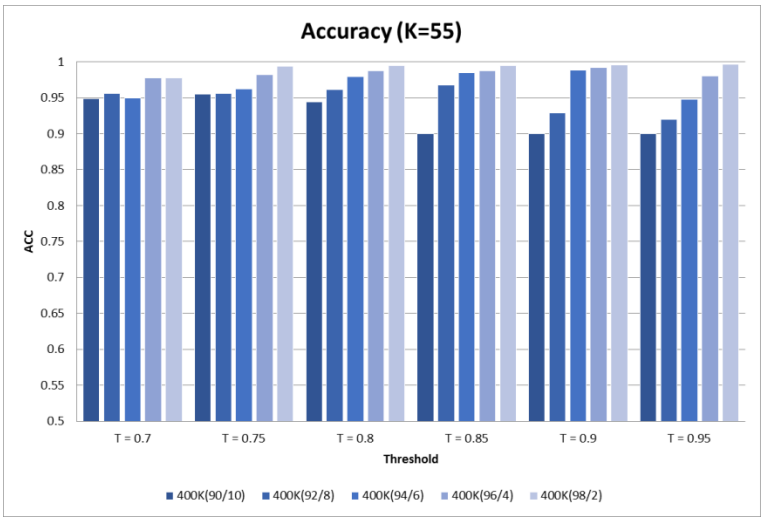
ภาพที่ 89 ROC ระหว่าง TNR และ FNR ของ วิธีการที่นำเสนอ (K=55)



ภาพที่ 90 ผลการทดลอง Detection Rate ของ วิธีการที่นำเสนอ (K=55)



ภาพที่ 91 ผลการทดลอง False Positive Rate ของ วิธีการที่นำเสนอ (K=55)



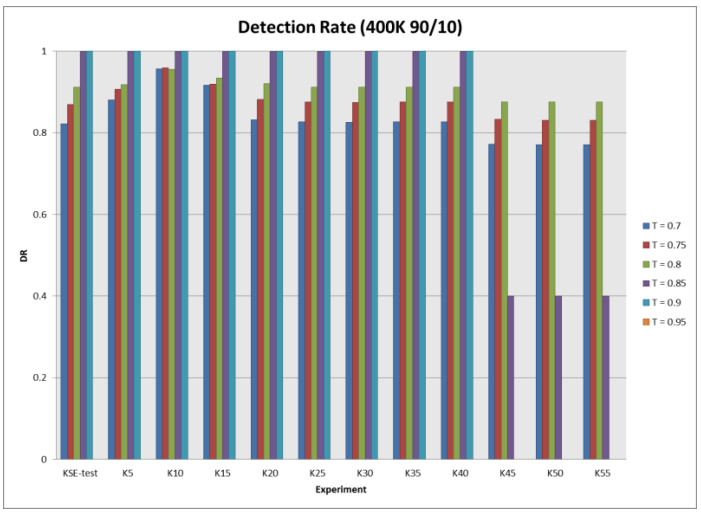
ภาพที่ 92 ผลการทดลอง Accuracy ของ วิธีการที่นำเสนอ (K=55)

จากผลการทดลองจะสังเกตได้ว่า การเพิ่มค่า Threshold ที่ใช้จำแนก จะส่งผลต่อการเพิ่มขึ้นของ Detection Rate เป็นแนวโน้มเพิ่มขึ้น และ ในทางกลับกันจะส่งผลให้ค่า False Positive Rate มีค่าลดต่ำลง

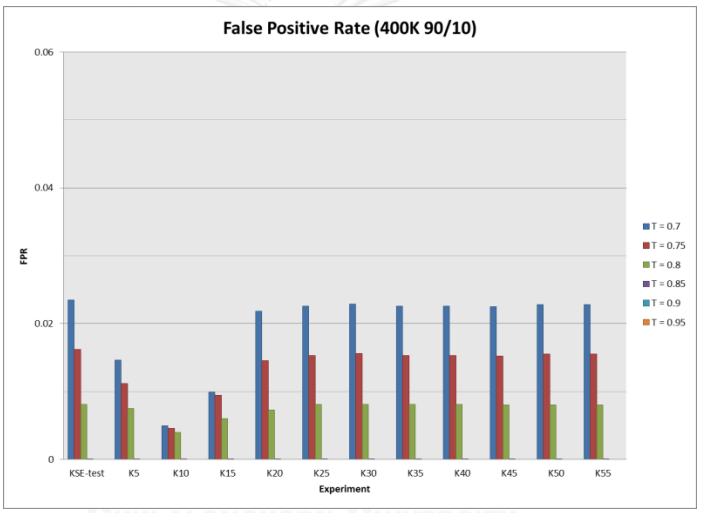
การใช้ค่า Threshold ที่มากเกินไป ถึงแม้จะให้ค่า Detection Rate ที่สูง แต่จะมี Accuracy ที่ต่ำลง เนื่องจากมีจำนวนที่จำแนกน้อยไม่ครอบคลุมจำนวนของการบุกรุกทั้งหมด หรือไม่มีการจำแนกเลย โดยสังเกตจาก Accuracy ที่ต่ำลง ดังที่สามารถเห็นได้ในส่วนของ Threshold 0.90 และ 0.95 ที่ไม่มีข้อมูลใดได้คะแนนเกินกว่า Threshold จึงไม่มีการจำแนก ทำให้มี Detection Rate เท่ากับ 0, ค่า True Positive Rate เป็น 0 และ False Positive Rate เป็น 1 ซึ่งในส่วนของค่า Threshold ที่สูงเกินไปนั้น เราจะไม่นำพิจารณานำมาใช้

จะเห็นได้ว่าการเลือกใช้ช่วงของค่า Threshold ในช่วงของ 0.80-0.85 เป็นช่วงที่ให้ประสิทธิภาพในการจำแนกสูงที่สุด คือ Detection Rate สูง False Positive Rate ต่ำ และให้ค่า Accuracy ที่สูงที่สุด

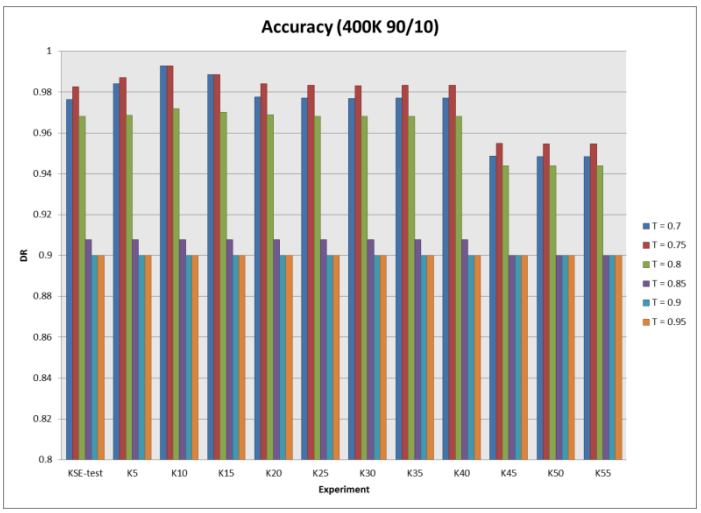
ซึ่งส่วนนี้จะมีการจำแนกข้อมูลปกติปะปนมาเยอะ จึงเป็นส่วนที่เราสนใจจะเพิ่มความแม่นยำจะอยู่ในช่วง 0.7-0.85 ซึ่งเป็นช่วงที่มีการจำแนกครอบคลุมจำนวนข้อมูลการบุกรุก ซึ่งจากผลการทดลองแสดงให้เห็นว่า วิธีการเพิ่ม K-Means algorithm เข้ามาจัดกลุ่มและสร้าง Normal Profile เพื่อคัดกรองผลลัพธ์นั้นจะทำให้ สามารถลดส่วนของการจำแนกข้อมูลปกติเป็นการบุกรุกได้ ทำให้ความแม่นยำโดยรวมสูงขึ้น มีคุณภาพของผลลัพธ์ที่ดีขึ้นกว่าการใช้ KSE-test เพียงอย่างเดียว สังเกตได้จาก กราฟ ROC ได้แสดงให้เห็นว่า ผลการทดลองส่วนใหญ่มีค่า True Positive Rate และ True Negative Rate ที่สูง และ ผลการจำแนกมี Detection Rate ที่เพิ่มขึ้น และ False Positive Rate ต่ำลงจากกรณี KSE-test ปกติ ดังผลการทดลองที่ และ เราจะทำการวิเคราะห์หาค่า K ที่มีความเหมาะสมจากการเปรียบเทียบดังแสดงในภาพที่ 93 ถึง ภาพที่ 107 คือ การเปรียบเทียบผลลัพธ์แต่ละค่า K ของข้อมูลทั้ง 5 ชุด ว่า เพื่อสังเกตว่า K ค่าใดที่สามารถเพิ่มความแม่นยำให้กับผลลัพธ์ได้มีประสิทธิภาพที่สุด



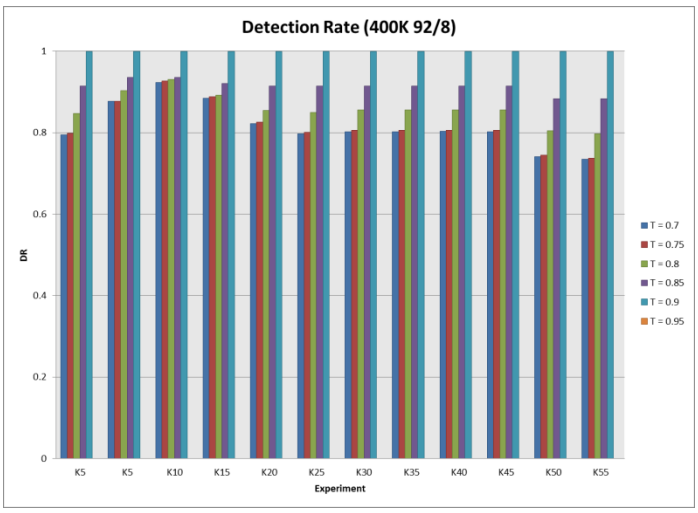
ภาพที่ 93 เปรียบเทียบ Detection Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 1



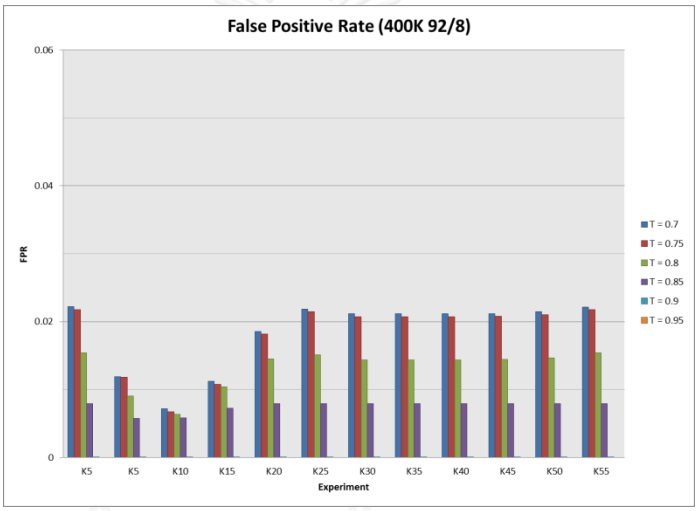
ภาพที่ 94 เปรียบเทียบ False Positive Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 1



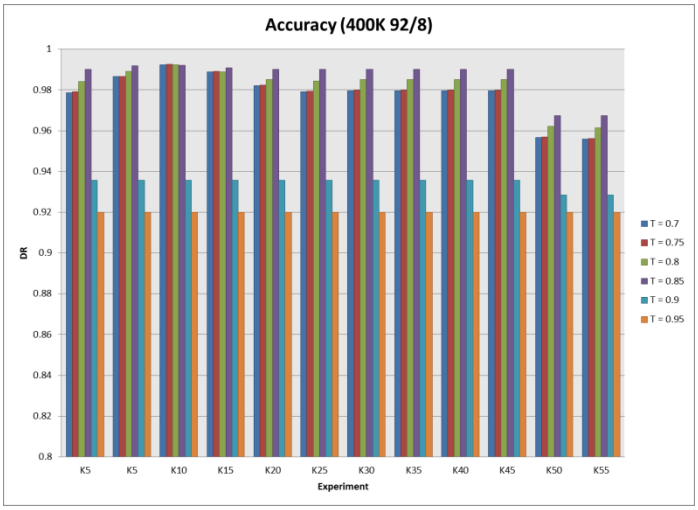
ภาพที่ 95 เปรียบเทียบ Accuracy ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 1



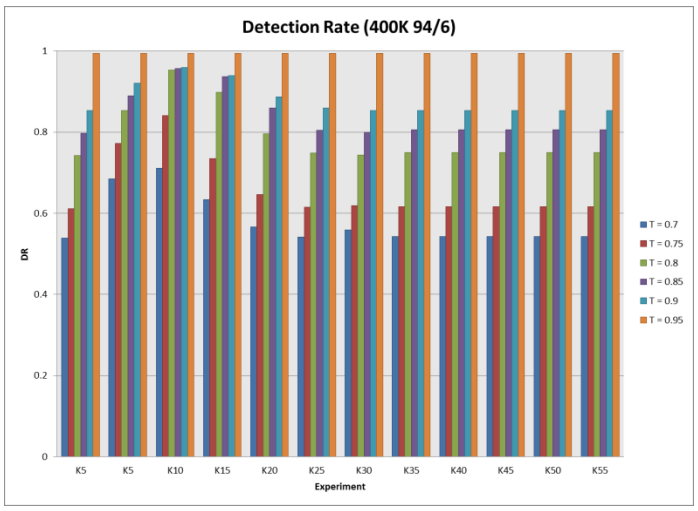
ภาพที่ 96 เปรียบเทียบ Detection Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 2



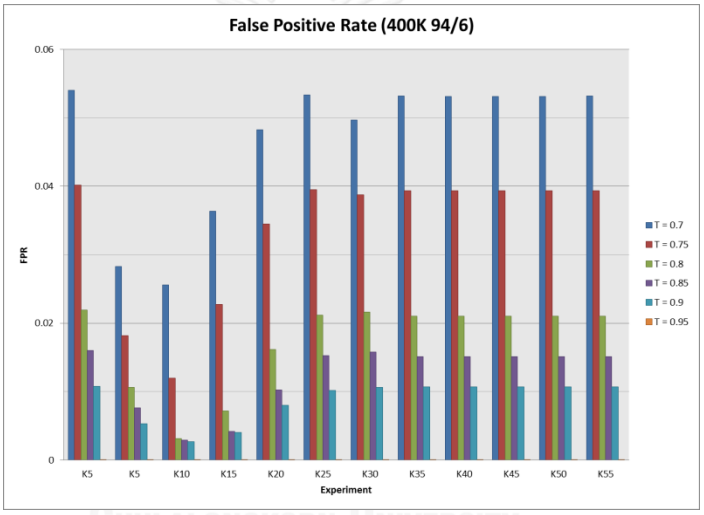
ภาพที่ 97 เปรียบเทียบ False Positive Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 2



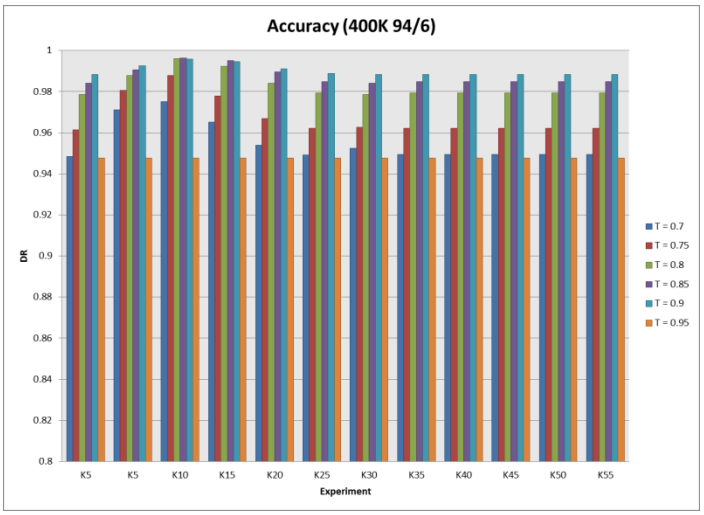
ภาพที่ 98 เปรียบเทียบ Accuracy ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 2



ภาพที่ 99 เปรียบเทียบ Detection Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 3

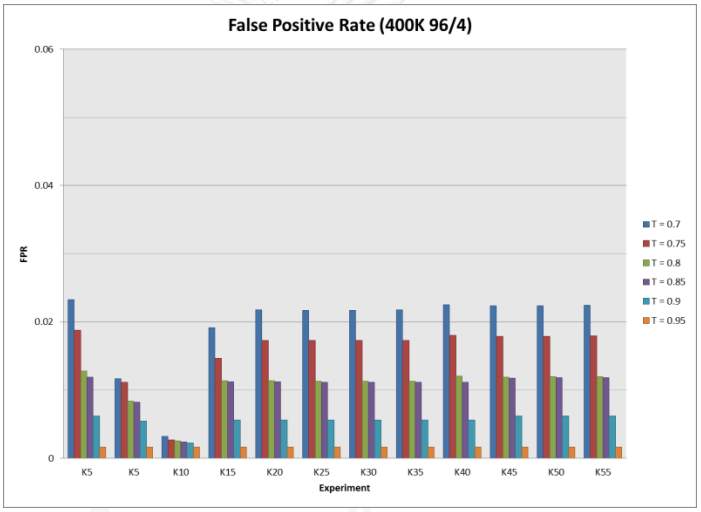


ภาพที่ 100 เปรียบเทียบ False Positive Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 3

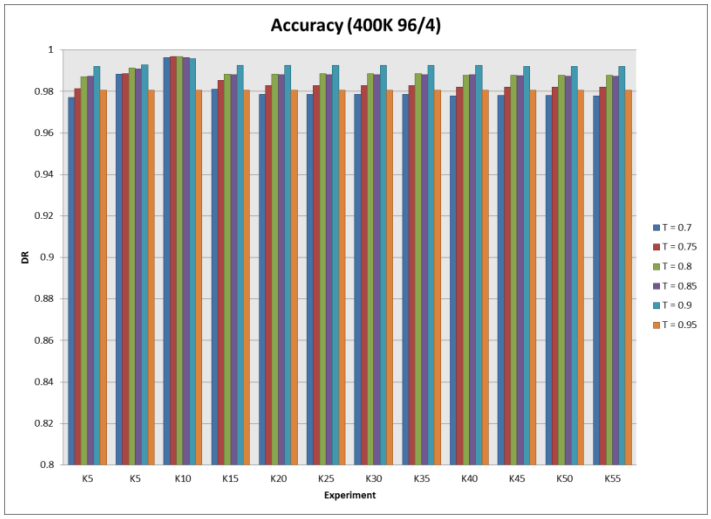


ภาพที่ 101 เปรียบเทียบ Accuracy ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 3

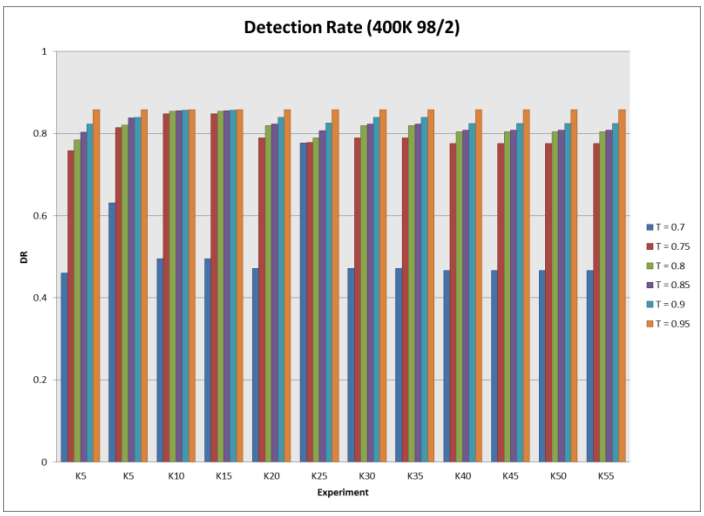
ภาพที่ 102 เปรียบเทียบ Detection Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 4



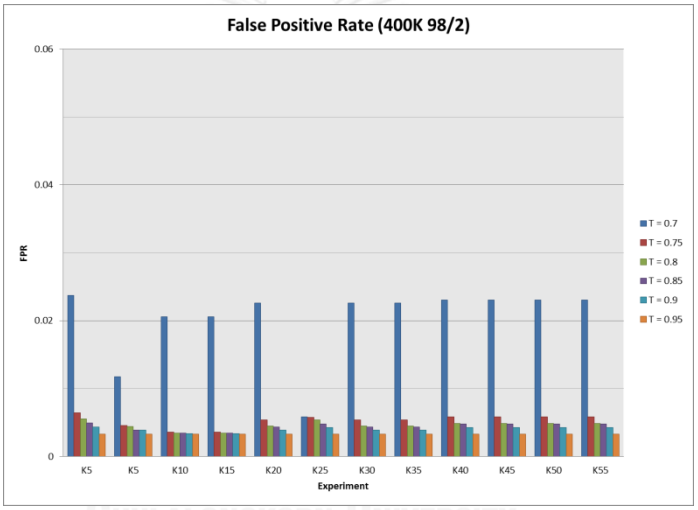
ภาพที่ 103 เปรียบเทียบ False Positive Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 4



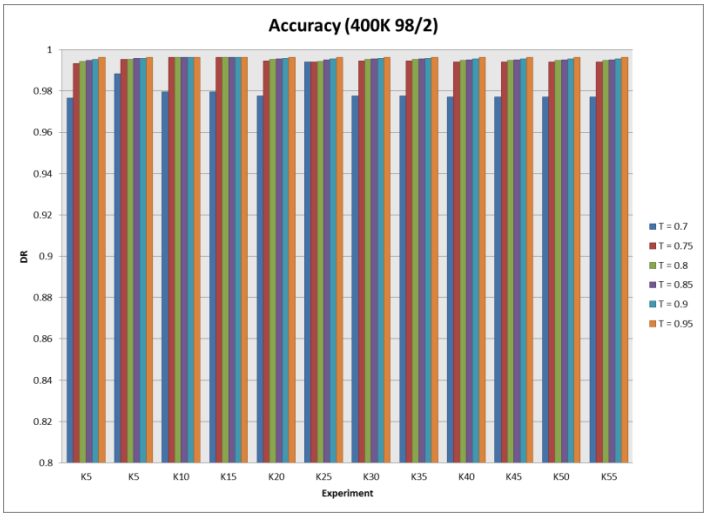
ภาพที่ 104 เปรียบเทียบ Accuracy ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 4



ภาพที่ 105 เปรียบเทียบ Detection Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 5



ภาพที่ 106 เปรียบเทียบ False Positive Rate ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 5



ภาพที่ 107 เปรียบเทียบ Accuracy ระหว่างค่า K ที่เลือกใช้ สำหรับข้อมูลชุดที่ 5

จากผลการเปรียบเทียบ พบว่า ค่า K ที่ทำให้ Detection Rate เพิ่มขึ้น และ ลด False Positive Rate ได้มีประสิทธิภาพมากที่สุดคือ ค่าที่อยู่ระหว่าง 5 ถึง 15 เป็นค่าที่ทำให้ผลลัพธ์ของการทดลองที่มี Threshold ในช่วง 0.70-0.85 มีความแม่นยำเพิ่มมากขึ้น

แต่จะเห็นได้ว่า การเลือกใช้ค่า K ที่มากเกินไป อาจทำให้ความถูกต้องแม่นยำลดลง เนื่องจากการแบ่งกลุ่มที่ละเอียดเกินไป ทำให้นัยสำคัญในด้านขนาดของกลุ่มข้อมูลถูกลดทอนลงไปด้วย ทำให้ความแม่นยำในการเรียงอันดับของข้อมูลแตกต่างกันออกไปตามผลลัพธ์ที่ได้จากการกำหนดค่า K

ข้อเสนอแนะในการพิจารณาเลือกค่า K ที่เหมาะสม ไม่ให้มีขนาดใหญ่หรือเล็กเกินไปนั้น ให้สังเกตการจุดกลุ่มที่ได้จากการทำการประมวลผล K-Means algorithm กว่ากลุ่มข้อมูลที่ได้ออกมานั้น กลุ่มข้อมูลขนาดใหญ่ที่สุดนั้น ไม่ควรมีจำนวนสมาชิกเกิน 20% ของชุดข้อมูลทั้งหมด และ ไม่ควรน้อยกว่า 10% ของชุดข้อมูลทั้งหมด

และ วิธีการพิจารณา ระดับ Threshold ที่เหมาะสมของข้อมูล คิดได้จาก การนำคะแนนที่มากที่สุดของการคำนวณ KSE-score จากชุดข้อมูล มาลบด้วย 0.1 จะได้ค่าประมาณของ Threshold ที่เหมาะสม

โดยสรุปผลลัพธ์การประมาณค่าตัวแปรที่เหมาะสม ตามวิธีการข้างต้นโดยใช้ค่าประมาณ Threshold ที่ใกล้เคียงกับ ที่คิดได้จากวิธีข้างต้น และ ประมาณค่าของ K ในช่วงระหว่าง 5 ถึง 15 จึงนำผลลัพธ์ที่ได้มาเฉลี่ย เพื่อหาค่าประมาณของผลลัพธ์จากการใช้วิธีการที่ได้นำเสนอไปข้างต้น ได้ผลลัพธ์ดังแสดงในตารางที่ 7

ตารางที่ 7 ค่าเฉลี่ยของผลลัพธ์จากวิธีการประมาณค่าตัวแปรที่นำเสนอ

Dataset	$KSEscore_{MAX}$	$KSEscore_{MAX}$ - 0.1	Threshold	K	DR	FPR	ACC
400K 90/10	0.9166	0.8166	0.8	K5	0.902933	0.009046	0.989095
				K10	0.930534	0.006342	0.992328
				K15	0.891216	0.010389	0.988743
400K 92/8	0.9763	0.8763	0.85	K5	0.935785	0.005742	0.991698
				K10	0.935417	0.005802	0.99197
				K15	0.92092	0.007228	0.990793
400K 94/6	0.9763	0.8763	0.85	K5	0.889064	0.007657	0.990485
				K10	0.956037	0.002886	0.996275
				K15	0.937178	0.004213	0.995115
400K 96/4	0.9992	0.8992	0.9	K5	0.878907	0.005458	0.992793
				K10	0.946311	0.002247	0.99587
				K15	0.875972	0.005609	0.992648
400K 98/2	0.9732	0.8732	0.85	K5	0.837818	0.00389	0.995883
				K10	0.854895	0.003411	0.99635
				K15	0.854895	0.003411	0.99635
average					0.903192	0.005555	0.993093

5.6 เปรียบเทียบผลลัพธ์การจำแนกการบุกรุกกับวิธีการอื่นๆ

ในส่วนของประสิทธิภาพการจำแนกข้อมูลการบุกรุกระบบ โดยเปรียบเทียบกับวิธีอื่นๆที่ทำการทดลองโดยใช้ข้อมูล KDD'99 เป็นชุดข้อมูลในการทดลอง ได้แก่ K-Means algorithm, PSO-KM algorithm[6], Modified K-Means algorithm[32], DBSCAN, IDBG, IIDBG[28], KMIDS (K-Means IDS)[28], VoteOut algorithm[27], NADO[33] และ TreeClus[32] เพื่อเปรียบเทียบกับวิธีการที่นำเสนอ แสดงการเปรียบเทียบผลลัพธ์ดังตารางที่ 8

ตารางที่ 8 เปรียบเทียบความสามารถในการจำแนกการบุกรุกระบบ

Algorithm	DR	FPR	Time Complexity
Proposed Method	0.903192	0.005555	O(n)
KSE-test	0.858184	0.008627	O(n)
K-Means algorithm	0.418000	0.026300	O(nKL)
PSO-KM algorithm	0.521000	0.017300	O(n)
Modified K-Means algorithm	0.883000	0.048960	O(n)
DBSCAN	0.581300	0.264000	$O(n^2)$
IDBG	0.780000	0.047100	$O(n^2)$
IIDBG (SemiSupervised)	0.866330	0.038300	$O(n^2)$
KMIDS (K-Means ensemble)	0.660900	0.003286	$O(n^2)$
VoteOut algorithm	0.942370	0.016300	$O(n^2)$
NADO	0.956310	0.045680	O(Kn)+O(Rnlogn)
TreeClus	0.989800	0.010200	$O(Kn^2)$

5.7 วิเคราะห์ประสิทธิภาพเชิงเวลาของวิธีการที่นำเสนอ และ เปรียบเทียบกับวิธีการประเภทอื่นๆ

ในส่วนของประสิทธิภาพเชิงเวลาของระบบ ได้ทำการวิเคราะห์ประสิทธิภาพเชิงเวลาของวิธีการที่นำเสนอ และ วิเคราะห์เปรียบเทียบกับวิธีการอื่นๆในแง่ของประสิทธิภาพเชิงเวลา และคุณภาพของผลลัพธ์(อ้างอิงจากกลุ่มของวิธีการที่เคยถูกนำมาทดสอบกับชุดข้อมูล KDD'99)

ในส่วนนี้เราจะอธิบายถึง ประสิทธิภาพเชิงเวลาของวิธีการที่นำเสนอ โดยส่วนแรก คือ ส่วนของการทำ KSE-test จะมีประสิทธิภาพเชิงเวลาเป็น O(C1C2n) โดยที่ n คือจำนวนของข้อมูลทั้งหมดในชุดข้อมูล C1 และ C2 คือขนาดของ sample1 และ sample2 ตามลำดับ ซึ่งในการทดลองของเรากำหนดให้ขนาดของ sample1 และ sample2 มีขนาดเท่ากัน ในส่วนต่อมาคือการจัดกลุ่มโดยใช้ K-means algorithm มีประสิทธิภาพเชิงเวลาเป็น O(nKL) โดย n คือจำนวนของข้อมูลทั้งหมดในชุดข้อมูล K คือจำนวนของกลุ่มข้อมูลผลลัพธ์ และ L คือ จำนวนรอบในการทำงานมากที่สุดที่ผู้ใช้กำหนด ส่วนสุดท้ายคือการทำงานในส่วนของการตัด normal instance และ การทำ majority vote ใช้ประสิทธิภาพเชิงเวลา O(n) ทำให้สรุปประสิทธิภาพเชิงเวลาทั้งหมดของวิธีการที่นำเสนอเป็นเชิงเส้น คือ O(n)

ทั้งนี้ได้แสดงการเปรียบเทียบประสิทธิภาพเชิงเวลาจากวิธีการที่นำเสนอกับวิธีการจำแนกการบุกรุกระบบประเภทต่างๆ ดังที่ได้แสดงไว้ในตารางที่ 9

ตารางที่ 9 เปรียบเทียบคุณภาพผลลัพธ์ และ ความซับซ้อนเชิงเวลา ของวิธีการต่างๆ

เปรียบเทียบประสิทธิภาพการทำงาน และ ความซับซ้อนเชิงเวลา					
Detection Techniques		Detection Quality	Time Complexity	หมายเหตุ	ตัวอย่าง
Proposed Method		สูง	O(n)	-	-
Outlier Detection (Anomaly Detection)		ปานกลาง-สูง	O(n)	ขึ้นกับวิธีการที่เลือกใช้	KSE-test
		สูง	O(n log n)	ขึ้นกับวิธีการที่เลือกใช้	NADO
		ปานกลาง-สูง	$O(n^2)$	ขึ้นกับวิธีการที่เลือกใช้	COF, LOF, LDOF, LoOP, TreeClus
Clustering	K-Means	ปานกลาง	O(n)	-	K-Means, PSO-KM, ODC
	DBSCAN	สูง	$O(n^2)$	-	DBSCAN, IDBG, IIDBG
K-Nearest Neighbor		สูง	$O(n^2)$	-	-
Ensemble	Clustering	ปานกลาง	O(n)	ขึ้นกับวิธีการที่เลือกใช้ และ การรวมผลลัพธ์	K-Means Ensemble(MajorityVote)
		ปานกลาง-สูง	$O(n^2)$	ขึ้นกับวิธีการที่เลือกใช้ และ การรวมผลลัพธ์	KMIDS(EA)
	Outlier Detection	สูง	$O(n^2)$	ขึ้นกับวิธีการที่เลือกใช้ และ การรวมผลลัพธ์	VoteOut algorithm

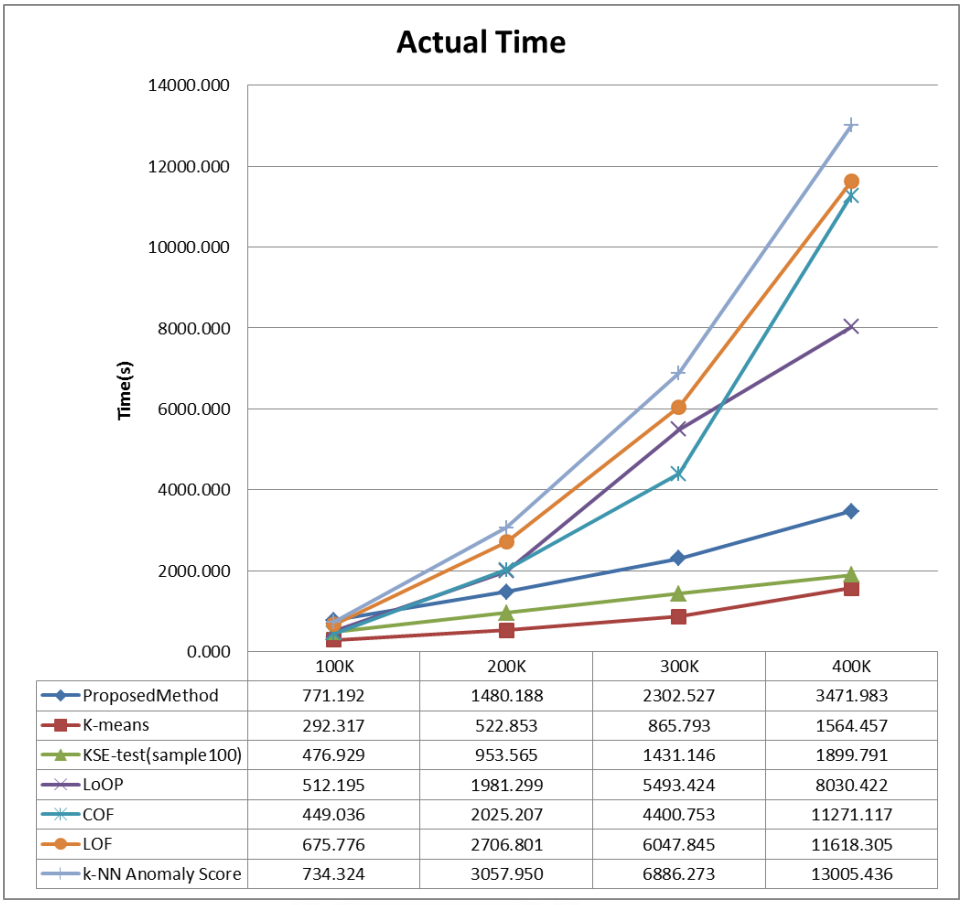
5.8 การทดสอบเวลาในการประมวลผลจริงเปรียบเทียบกับวิธีการอื่นๆ

ทำการประมวลผลเปรียบเทียบกับวิธีการพื้นฐานที่เป็นที่นิยมนำมาประยุกต์ใช้ในการตรวจสอบการบุกรุกระบบ ได้แก่ K-Means clustering algorithm[5], DBSCAN algorithm[34], COF[35], LOF[36], LoOP[37] และ เปรียบเทียบกับ KSE-Test[12] แบบปกติ ได้แสดงผลลัพธ์ไว้ในตารางที่ 10 โดยทำการประมวลผลแต่ละวิธีด้วยการแปรผันขนาดข้อมูลต่างๆดังนี้ 100,000 – 400,000 แถวข้อมูล เพิ่มขึ้นครั้งละ 100,000 แถวข้อมูล ได้ผลลัพธ์จากการทดลองดังแสดงในตารางที่ 10

ตารางที่ 10 เปรียบเทียบเวลาในการประมวลผลจริง แปรผันตามขนาดของข้อมูล

Algorithm	Parameters	ขนาดของชุดข้อมูล			
		100K	200K	300K	400K
ProposedMethod	SampleSize=100, K=50	771.192	1480.188	2302.527	3471.983
K-means	K=50	292.317	522.853	865.793	1564.457
KSE-test(sample100)	SampleSize=100	476.929	953.565	1431.146	1899.791
LoOP	K=10, NormalizationFactor=3	512.195	1981.299	5493.424	8030.422
COF	K=25	449.036	2025.207	4400.753	11271.117
LOF	Kmin=20, Kmax=40	675.776	2706.801	6047.845	11618.305
k-NN Anomaly Score	K=10	734.324	3057.950	6886.273	13005.436
DBSCAN	Epsilon=0.03, m=10	4454.39	18230.44	39902.72	70472.05

สามารถแสดงผลเวลาในการประมวลผลเป็นกราฟ(โดยขอตัดส่วนข้อมูลเวลาของ DBSCAN ออก เพื่อให้สามารถเห็นรายละเอียดเวลาของวิธีการอื่นๆได้ชัดเจนยิ่งขึ้น) ได้ดังภาพที่ 108



	100K	200K	300K	400K
ProposedMethod	771.192	1480.188	2302.527	3471.983
K-means	292.317	522.853	865.793	1564.457
KSE-test(sample100)	476.929	953.565	1431.146	1899.791
LoOP	512.195	1981.299	5493.424	8030.422
COF	449.036	2025.207	4400.753	11271.117
LOF	675.776	2706.801	6047.845	11618.305
k-NN Anomaly Score	734.324	3057.950	6886.273	13005.436

ภาพที่ 108 กราฟเปรียบเทียบเวลาในการประมวลผลจริงของวิธีการแบบต่างๆ

จากภาพแสดงให้เห็นว่า K-Means, KSE-test และ วิธีการที่นำเสนอ มีประสิทธิภาพเชิงเวลาเป็นเชิงเส้น O(n) คือ เมื่อขนาดข้อมูลเพิ่มขึ้น เวลาที่ใช้ในการประมวลผล ก็เพิ่มในสัดส่วนเดียวกัน ในขณะที่วิธีการอื่นๆ มีประสิทธิภาพเชิงเวลาเป็น $O(n^2)$ คือเมื่อขนาดของชุดข้อมูลเพิ่มขึ้นจากเดิม เวลาที่ใช้ในการประมวลผลจะเพิ่มขึ้นเป็นเป็น Exponential

5.9 การทดสอบการประมวลผลแบบขนาน บน platform Apache Spark

ในเนื้อหาส่วนนี้จะเป็นการต่อยอดจาก KSE algorithm มาใช้ในการประมวลผลงานขนาดใหญ่ โดยทำการแปลงให้อยู่ในรูปแบบของ RDD เพื่อสามารถประมวลผลบน Spark ซึ่งมีข้อดีคือสามารถประมวลผลภาระงานแบบขนานได้ โดยใช้ประโยชน์จากการประมวลผลบนหน่วยความจำ (หรือเรียกว่า การประมวลผลแบบ in-memory) คือการเก็บข้อมูลระหว่างการประมวลผลไว้บนหน่วยความจำ ทำให้ไม่จำเป็นต้องเขียนผลลัพธ์ระหว่างการทำงานลงในหน่วยบันทึกข้อมูล

ซึ่งในส่วนของการประมวลผล KSE-test algorithm เพื่อกำหนด KSE-score สำหรับข้อมูลแต่ละตัวในชุดข้อมูล มีการทำงานในแต่ละครั้งเป็นอิสระไม่ขึ้นต่อกัน ทำให้สามารถประมวลผลแบบขนานโดยการแปลงให้อยู่ในรูป Transformation และ Action (แนวทางคล้ายกับการทำงานของ MapReduce Model) ได้อย่างง่ายดาย และ นอกเหนือจากนั้นยังสามารถใช้ประโยชน์จาก Shared Variable ในการ Broadcast ค่าที่ต้องการกระจายไปยังทุก Node ใน cluster ได้อีกด้วย โดยการทดลองนี้จะทำการดัดแปลง KSE-test ให้อยู่ในรูปแบบของโปรแกรม โดยได้ทำการปรับปรุงการทำงานของโปรแกรมอ้างอิงรูปแบบการพัฒนาโปรแกรมสำหรับ Apache Spark อธิบายไว้เป็น Pseudo code ลำดับขั้นตอนการทำงานตามที่ได้แสดงในภาพที่ 109

```
function calculateEuclideanDistance(input_record_RDD,sample2,sample_size){
    function Map(input_record_RDD,sample2,sample_size){
        distanceArray[] = create_1D_Array(sample_size);
        for(i=0 to sample_size){
            distanceArray[i] = EuclideanDistance(input_record_RDD.getValue(),sample2[i].getValue()) ;
        }
        key = record_id;
        value[] = distanceArray;
        return tuple<key,value[]>;
    }
}

function KSEtest(input_distanceArray_RDD,sample1,sample_size){
    function Map(input_distanceArray_RDD,sample1,sample_size){
        sum = 0;
        for(i=0 to sample_size){
            distanceArray_x [] = input_distanceArray_RDD.getValue();
            distanceArray_y [] = sample1.getValue();
            sum+= KStest(distanceArray_x , distanceArray_y);
        }
        key = record_id;
        value = sum/sample_size;
        return tuple<key,value>;
    }
}

function Parallel_KSEtest(inputFile,sample_size){

    data[] = readRecord(input);
    dataRDD = parallelize(data);

    sample2[] = dataRDD.takeSample(sample_size);
    broadcast(sample2);
    distanceArrayRDD [] = calculateEuclideanDistance(dataRDD,sample2,sample_size);

    sample1[] = distanceArrayRDD.takeSample(sample_size);
    broadcast(sample1);
    KSEscoreRDD [] = KSEtest(distanceArrayRDD,sample1,sample_size);

    writeFile(KSEscoreRDD);

}
```

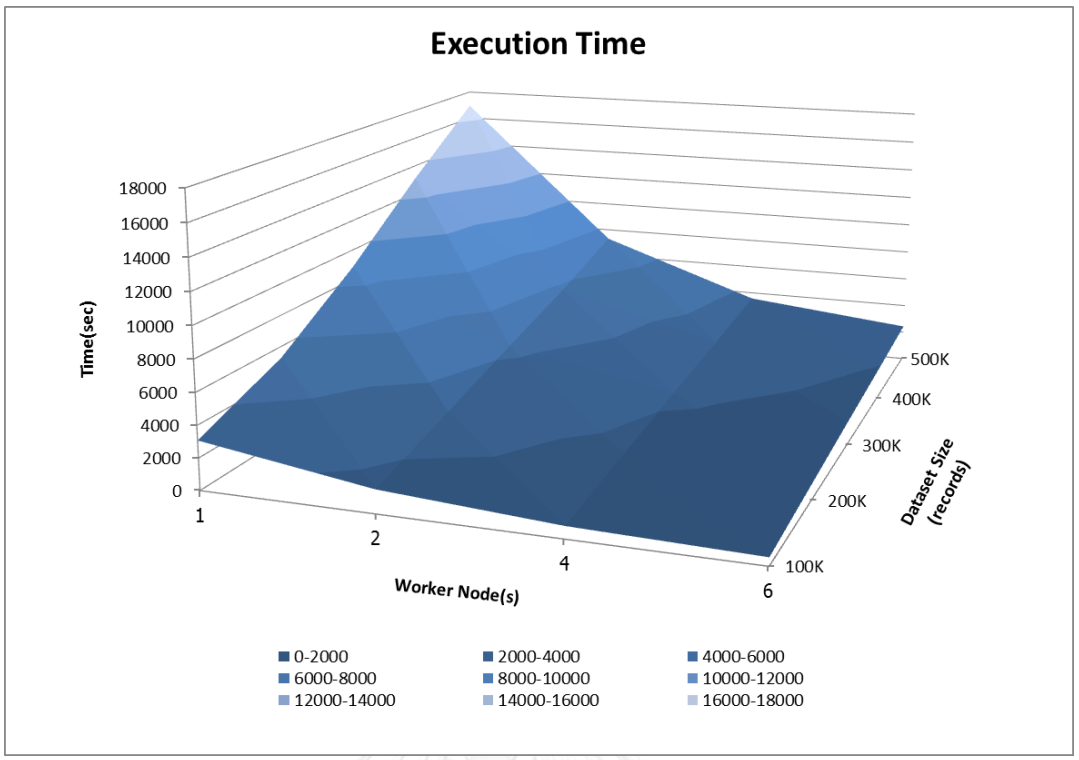
ภาพที่ 109 ลำดับขั้นตอนการทำงานของโปรแกรมคำนวณ KSE-Score แบบขนาน

จากที่ได้แสดงลำดับขั้นตอนการทำงานไว้ในภาพที่ 109 การทำงานได้แบ่งออกเป็นสองส่วนหลักๆ คือ การคำนวณ Euclidean distance ของข้อมูลแต่ละตัว และ ขั้นตอนการคำนวณคะแนน KSE-score ของข้อมูลในแต่ละตัวในชุดข้อมูล โดยมีส่วนที่มีการสลับการทำงานจากวิธีเดิมเพื่อเพิ่มประสิทธิภาพ และ ลดการคำนวณซ้ำซ้อนลง คือ การสลับลำดับการสุ่มชุดข้อมูล sample1 หลังการคำนวณ Euclidean distance แทนการสุ่ม sample1 และ sample2 พร้อมกันแล้วสร้าง Distance Matrix ก่อนการคำนวณ

การทดลองนี้ดำเนินการโดยใช้ Spark cluster จำนวน 7 node (เป็น Master node และ Worker node อีก 6 node) โดยที่แต่ละ node มีประกอบด้วย หน่วยประมวลผล Intel(R) Xeon(R) CPU E5-2670 v2 2.50GHz หน่วยความจำ 4 GB โดยจะทำการประมวลผลข้อมูลที่มีขนาดแตกต่างกัน 5 ขนาด แต่ละขนาดมีจำนวน 5 ชุด คำนวณ outlier score ด้วย KSE-test (โดยใช้ขนาด sample size = 500) เพื่อเปรียบเทียบเวลาที่ใช้ในการประมวลผลโดยใช้จำนวน Node ต่างกัน เพื่อสังเกตความสัมพันธ์ระหว่างจำนวน Node ที่ใช้ กับเวลาในการประมวลผล ผลลัพธ์ โดยจะทำการทดลองโดยการจับเวลาในการประมวลผลจริง โดยแปรผันจำนวน node ในการทำงานโดยใช้ค่าดังนี้ 1,2,3 และ 6 node ตามลำดับ ดังแสดงผลลัพธ์เวลาในการประมวลผลในตารางที่ 11

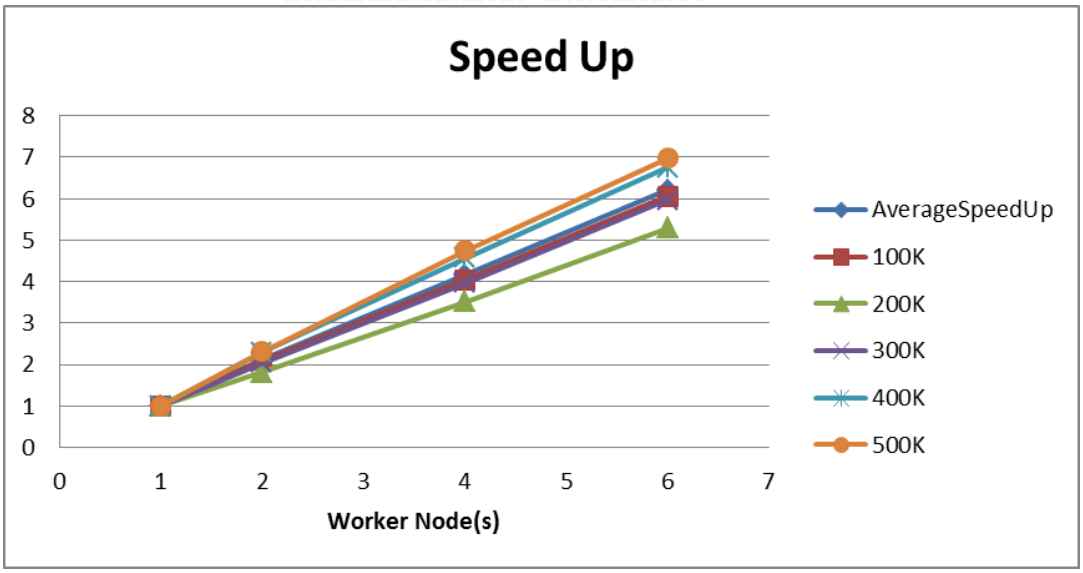
ตารางที่ 11 เวลาในการประมวลผล แบบต่างๆแปรผันตามขนาดของข้อมูล และจำนวน node

Execution Time					
Dataset	Sequential	Number of Worker Node			
		1	2	4	6
100K(1)	6277.505	3097.366	1455.048	759.743	514.408
100K(2)	6318.1	3031.907	1467.849	775.129	515.231
100K(3)	6464.475	3095.818	1483.165	763.516	512.889
100K(4)	6464.475	3090.08	1482.618	770.759	508.714
100K(5)	6400.502	3154.525	1493.476	753.661	499.608
Average	6385.011	3093.939	1476.431	764.5616	510.17
200K(1)	12675.7	5230.965	2867.224	1493.725	984.942
200K(2)	12556.29	5297.727	2877.257	1511.257	991.553
200K(3)	12892.21	5201.553	2887.858	1503.107	1000.908
200K(4)	13046.99	5408.893	2937.403	1519.508	1010.079
200K(5)	12836.44	5296.978	2926.204	1501.851	1008.533
Average	12801.53	5287.223	2899.189	1505.89	999.203
300K(1)	18835.73	8847.231	4364.061	2198.425	1473.007
300K(2)	18930.88	8929.744	4275.244	2274.326	1495.997
300K(3)	19251.18	8836.432	4304.359	2204.706	1484.265
300K(4)	19139.66	8825.705	4346.336	2267.242	1486.796
300K(5)	19164.87	8768.612	4318.915	2209.77	1467.423
Average	19064.46	8841.545	4321.783	2230.894	1481.498
400K(1)	25011.9	12854.93	5620.123	2837.233	1917.235
400K(2)	25276.75	13045.6	5874.797	2876.557	1924.26
400K(3)	25496.4	12918.71	5802.523	2838.561	1915.461
400K(4)	25655.96	13301.49	5480.455	2899.778	1958.685
400K(5)	25810.56	13255.37	5642.254	2879.789	1974.704
Average	25450.31	13075.22	5684.03	2866.384	1938.069
500K(1)	31918.26	16894.89	6921.898	3516.174	2414.972
500K(2)	31652.83	16905.01	7353.557	3507.639	2413.531
500K(3)	31675.2	17117.45	7333.137	3561.274	2409.419
500K(4)	31889.42	16753.58	7570.358	3589.25	2422.396
500K(5)	31889.91	17012.42	7337.57	3644.411	2476.435
Average	31805.12	16936.67	7303.304	3563.75	2427.351



ภาพที่ 110 แผนภูมิแสดงความสัมพันธ์ระหว่าง ขนาดของข้อมูล จำนวน node และ เวลาที่ใช้

จากผลการทดลองพบว่า เวลาที่ใช้ในการประมวลผล มีความสัมพันธ์กับจำนวนของ Node ที่ใช้ในการประมวลผล โดยเมื่อเพิ่มจำนวน Node มากขึ้น จะใช้เวลาในการประมวลผลน้อยลง และความสัมพันธ์ของขนาดข้อมูลกับเวลาที่ใช้มีความสัมพันธ์กันแบบเชิงเส้น และ สามารถสรุปเป็นแผนภาพ Speed Up ต่อจำนวน Worker Node ได้ดังภาพที่ 111



ภาพที่ 111 แผนภาพแสดง Speed Up ต่อจำนวนของ Worker Node

บทที่ 6
สรุปผลการทดลอง

6.1 สรุปผลการทดลอง

งานวิจัยนี้ได้นำเสนอวิธีการในการจำแนกข้อมูลการบุกรุกระบบจาก Log File ขนาดใหญ่ โดยเป้าหมายเพื่อการเพิ่มประสิทธิภาพในการประมวลผล ให้สามารถจำแนกได้ถูกต้อง มีการผิดพลาดน้อย และ สามารถทำงานได้รวดเร็ว นอกจากนั้นยังมีคุณสมบัติในการขยายระบบได้ง่าย เพื่อรองรับกับภาระงานขนาดใหญ่ ผลการทดสอบในด้านของความสามารถในการจำแนกการบุกรุก โดยใช้ Confusion Matrix และ การคำนวณ Detection Rate และ False Positive Rate พบว่าสามารถปรับปรุงความสามารถในการตรวจสอบการบุกรุก และ ลดความผิดพลาดในการจำแนกลงได้ตามเป้าหมาย โดยที่ยังสามารถทำงานได้รวดเร็วกว่าวิธีการที่ให้ความแม่นยำในระดับใกล้เคียงกัน และ ในด้านของเวลา วิธีการที่ออกแบบนั้น สามารถทำงานได้โดยมีประสิทธิภาพเชิงเวลาเป็นแบบเชิงเส้น ทำให้มีความรวดเร็วในการประมวลผลข้อมูล ส่วนในด้านของการขยายระบบเพื่อทำการประมวลผลแบบกระจาย ได้ทำการทดสอบการแปลง KSE-test algorithm ให้สามารถประมวลผลแบบขนานได้บน platform Apache Spark เพื่อพิสูจน์ความสามารถในการประมวลผลแบบขนาน

โดยในส่วนของประสิทธิภาพของการจำแนกที่ได้นั้นขึ้นอยู่กับการกำหนดค่าตัวแปรต่างๆ ดังนี้ Threshold, K และ Nsize ส่วนขนาดของ Sample ที่เลือกใช้นั้น ไม่ทำให้ผลลัพธ์การจำแนกแตกต่างกันอย่างมีนัยสำคัญ เพียงเลือกใช้ขนาดของ Sample ที่ไม่น้อยจนเกินไป

ในส่วนของ Threshold นั้น เป็นส่วนที่สำคัญที่สุดในการจำแนกผลลัพธ์ โดยการเลือกใช้ค่า Threshold ที่มากนั้น จะทำให้การจำแนกมีความถูกต้องสูง(Detection Rate สูง) และ มีความผิดพลาดน้อย(False Positive Rate ต่ำ) แต่การเลือกใช้ Threshold ที่มากเกินไป จะทำให้จำแนกการบุกรุกได้ไม่ครอบคลุมจำนวนการบุกรุกทั้งหมดที่แอบแฝงอยู่ในข้อมูล ทำให้ความแม่นยำ(Accuracy)โดยรวมนั้นต่ำลงเมื่อใช้ Threshold ที่มากเกินไป แต่ในทางตรงกันข้าม เมื่อพิจารณาเลือกใช้ค่า Threshold ที่ต่ำลง จะทำให้มีโอกาสจำแนกข้อมูลปกติ กลายเป็นข้อมูลการบุกรุกได้มากขึ้น ทำให้ความถูกต้องแม่นยำต่ำลง และ มีอัตราการผิดพลาดสูงขึ้น

การนำเสนอวิธีการเพิ่มเติม เพื่อลดผลกระทบจากการเลือกใช้ Threshold เพื่อให้สามารถเลือกใช้ Threshold ที่ต่ำลง และ มีความยืดหยุ่นในการกำหนดค่ามากขึ้น ด้วยการนำวิธีการจัดกลุ่มข้อมูลเข้ามาลดความผิดพลาดในผลลัพธ์การจำแนก(จำแนกข้อมูลปกติผิดพลาดกลายเป็นการบุกรุก เมื่อเลือกใช้ระดับ Threshold ที่ต่ำลง) เมื่อมีการใช้ขั้นตอนเพิ่มเติมขึ้นมา ทำให้ต้องมี

การกำหนดค่าของตัวแปรที่ใช้ในขั้นตอนดังกล่าว คือ มีการกำหนดสัดส่วนของข้อมูลปกติ Nsize และ จำนวนกลุ่ม K สำหรับขั้นตอนการจัดกลุ่ม ที่เหมาะสม เพื่อให้สามารถกำหนดและจำแนกรายการของข้อมูลปกติได้อย่างแม่นยำ เพื่อนำมาใช้ตัดข้อมูลปกติที่ทำนายผิดพลาดในขั้นตอนแรก เพื่อให้การใช้ค่า Threshold ที่ต่ำลง โดยประสิทธิภาพในการจำแนกไม่ลดลงมากจนเกินไป เพื่อเป็นการลดจุดอ่อนจากการจำแนกในขั้นตอนแรก การกำหนด Nsize และ ค่า K ที่เหมาะสมมีความสัมพันธ์กับสัดส่วนของข้อมูลในชุดข้อมูล

ซึ่งการกำหนด Nsize เพื่อสร้าง Normal Profile ให้มีประสิทธิภาพที่สุดนั้น คือ การเลือกให้ครอบคลุมส่วนที่เป็นข้อมูลปกติมากที่สุด โดยจำเป็นต้องมีการเผื่อขอบเขตไม่ให้ครอบคลุมไปถึงส่วนข้อมูลที่เป็นการบุกรุกเข้ามาใน Normal Profile ด้วย ในส่วนของค่า K ที่เหมาะสมนั้น ไม่ควรที่จะมีจำนวนกลุ่มเยอะเกินไป จะทำให้นัยสำคัญในด้านขนาดของกลุ่มนั้นลดลง ทำให้ความถูกต้องในการจำแนกข้อมูลปกติลดลง จึงนำเสนอวิธีการประมาณขนาดของกลุ่มที่เหมาะสมโดยพิจารณาจากขนาดของกลุ่มข้อมูลขนาดใหญ่ที่สุดที่ได้จากขั้นตอนการจัดกลุ่ม โดยขนาดของกลุ่มที่มีขนาดใหญ่ที่สุด ไม่ควรมีขนาดเกิน 20 เปอร์เซ็นต์ของข้อมูลทั้งหมด และ ไม่ควรน้อยกว่า 10 เปอร์เซ็นต์ของข้อมูลทั้งหมด เป็นการประเมินว่าค่า K ที่ใช้นั้นอยู่ในช่วงที่เหมาะสมสำหรับการสร้าง Normal Profile และ การเลือกค่า Threshold ที่เหมาะสมกับวิธีการที่นำเสนอ พิจารณาจากคะแนนความแปลกแยกที่สูงที่สุดที่ได้จากขั้นตอน KSE-test นำมาลบด้วย 0.1 เป็นการประมาณค่า Threshold เพื่อให้ได้จำนวนของการจำแนกการบุกรุกที่เหมาะสม โดยวิธีการหาค่าตัวแปรที่เหมาะสมของข้อมูลแต่ละชุดเพื่อให้ได้ผลลัพธ์ที่แม่นยำนั้น ควรมีการทดลองเพื่อหาวิธีประมาณค่าที่ดีที่สุดต่อไป

6.2 ประโยชน์ที่ได้รับจากงานวิจัย

1. ได้วิธีการที่มีประสิทธิภาพในการวิเคราะห์ข้อมูลบนปูมขนาดใหญ่ด้วยวิธีการแบบไม่มีการชี้นำ ซึ่งสามารถแทนการใช้ผู้เชี่ยวชาญในการวิเคราะห์และจำแนกข้อมูลทั้งหมด
2. สามารถทำการจำแนกข้อมูลแปลกแยกที่มีความแม่นยำมากขึ้น และ มีความผิดพลาดน้อยลง
3. ได้วิธีการวิเคราะห์ข้อมูลการบุกรุกระบบที่มีประสิทธิภาพเชิงเวลาเป็นเชิงเส้น
4. สามารถทำการประมวลผลตรวจจับการบุกรุกในปูมข้อมูลขนาดใหญ่ได้รวดเร็วขึ้นโดยใช้ประโยชน์จากความสามารถในการประมวลผลแบบขนาน และ ขยายระบบเพื่อรองรับภาระงานที่มีขนาดเพิ่มขึ้นได้ง่าย

6.3 แนวทางการวิจัยในอนาคต

งานวิจัยนี้สามารถนำไปปรับปรุงพัฒนาให้มีประสิทธิภาพดียิ่งขึ้นโดยการ นำไปทดสอบทดลองเพื่อหาค่าตัวแปรที่เหมาะสมต่อไป และ ยังสามารถประยุกต์เทคนิควิธีการอื่นๆร่วมกับแนวคิดพื้นฐานที่งานวิจัยนี้ได้นำเสนอ เพื่อให้มีความสามารถทั้งด้านความแม่นยำของผลลัพธ์ และ เพิ่มความรวดเร็วในการประมวลผลให้ดียิ่งขึ้นในอนาคต

สามารถนำวิธีการที่นำเสนอไปประยุกต์ใช้กับงานในด้านอื่นๆ หรือ มีการนำไปต่อยอดเพิ่มเติมเพื่อให้สามารถทำงานในการประมวลผลภาระงานแบบทันกาล(real-time)ได้

รายการอ้างอิง

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.,* vol. 41, pp. 1-58, 2009.

[2] F. Sabahi and A. Movaghar, "Intrusion Detection: A Survey," in *Systems and Networks Communications,* 2008. *ICSNC '*08. 3*rd International Conference on*, 2008, pp. 23-26.

[3] M. A. Dalal and N. D. Harale, "A survey on clustering in data mining," presented at the Proceedings of the International Conference & Workshop on Emerging Trends in Technology, Mumbai, Maharashtra, India, 2011.

[4] M. Ahmed and A. Naser, "A novel approach for outlier detection and clustering improvement," in *Industrial Electronics and Applications (ICIEA),* 2013 8*th IEEE Conference on*, 2013, pp. 577-582.

[5] (2015). *k-means clustering*. Available: https://en.wikipedia.org/wiki/K-means_clustering

[6] L. Zhengjie, L. Yongzhong, and X. Lei, "Anomaly Intrusion Detection Method Based on K-Means Clustering Algorithm with Particle Swarm Optimization," in *Information Technology, Computer Engineering and Management Sciences (ICM),* 2011 *International Conference on*, 2011, pp. 157-161.

[7] W. Fangfei, J. Qingshan, S. Liang, and W. Nannan, "An Intrusion Detection System Based on the Clustering Ensemble," in *Anti-counterfeiting, Security, Identification,* 2007 *IEEE International Workshop on*, 2007, pp. 121-124.

[8] (2015). *Kolmogorov–Smirnov test*. Available: https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test

[9] "KS-test," 2015.

[10] (2015). *Kolmogorov-Smirnov Goodness-of-Fit Test*. Available: http://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm

[11] (2015). *Empirical distribution function*. Available: https://en.wikipedia.org/wiki/Empirical_distribution_function

[12] M. S. Kim, "Robust, Scalable Anomaly Detection for Large Collections of Images," in *Social Computing (SocialCom),* 2013 *International Conference on*, 2013, pp. 1054-1058.

[13] (2015). *MapReduce*. Available: https://en.wikipedia.org/wiki/MapReduce

[14] (2015). *MapReduce Tutorial*. Available: http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

[15] (2015). *Apache Hadoop*. Available: http://hadoop.apache.org/

[16] (2015, Apache Spark. Available: https://spark.apache.org/

[17] (2015). *Wikipedia - Apache Spark*. Available: https://en.wikipedia.org/wiki/Apache_Spark

[18] T. da Silva Morais, "Survey on Frameworks for Distributed Computing: Hadoop, Spark and Storm."

[19] (2015). *Confusion matrix*. Available: http://en.wikipedia.org/wiki/Confusion_matrix

[20] (2014). *Sensitivity and specificity*. Available: http://en.wikipedia.org/wiki/Sensitivity_and_specificity

[21] R. Pamula, J. K. Deka, and S. Nandi, "Pruning based method for outlier detection," in *Emerging Applications of Information Technology (EAIT),* 2012 *Third International Conference on*, 2012, pp. 210-213.

[22] J. Therdphapiyanak and K. Piromsopa, "Applying Hadoop for log analysis toward distributed IDS," presented at the Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication, Kota Kinabalu, Malaysia, 2013.

[23] (2015). *Apache Mahout*. Available: http://mahout.apache.org/

[24] J. Therdphapiyanak and K. Piromsopa, "An analysis of suitable parameters for efficiently applying K-means clustering to large TCPdump data set using Hadoop framework," in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON),* 2013 10*th International Conference on*, 2013, pp. 1-6.

[25] L. Xiuqin, W. Peng, and W. Bin, "Log analysis in cloud computing environment with Hadoop and Spark," in *Broadband Network & Multimedia Technology (IC-BNMT),* 2013 5*th IEEE International Conference on*, 2013, pp. 273-276.

[26] M. N. Nisha, S. Mohanavalli, and R. Swathika, "Improving the quality of clustering using cluster ensembles," in *Information & Communication Technologies (ICT),* 2013 *IEEE Conference on*, 2013, pp. 88-92.

[27] H. Bin, L. Wen-fang, C. De-li, and S. Liang, "An Intrusion Detection Method Based on Outlier Ensemble Detection," in *Networks Security, Wireless Communications and Trusted Computing,* 2009. *NSWCTC '*09. *International Conference on*, 2009, pp. 600-603.

[28] X.-y. Li, G.-h. Gao, and J.-x. Sun, "A New Intrusion Detection Method Based on Improved DBSCAN," in *Information Engineering (ICIE),* 2010 *WASE International Conference on*, 2010, pp. 117-120.

[29] (2015). *KDD Cup* 1999 *Computer network intrusion detection*. Available: http://sigkdd.org/kdd-cup-1999-computer-network-intrusion-detection

[30] (2015). *WEKA*. Available: http://www.cs.waikato.ac.nz/ml/weka/

[31] (2015). *Receiver Operating Characteristic (ROC)*. Available: https://en.wikipedia.org/wiki/Receiver_operating_characteristic

[32] T. Li and J. Wang, "Research on Network Intrusion Detection System Based on Improved K-means Clustering Algorithm," in *Computer Science-Technology and Applications,* 2009. *IFCSTA '*09. *International Forum on*, 2009, pp. 76-79.

[33] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "NADO: network anomaly detection using outlier approach," presented at the Proceedings of the 2011 International Conference on Communication, Computing & Security, Rourkela, Odisha, India, 2011.

[34] (2015). *DBSCAN*. Available: https://en.wikipedia.org/wiki/DBSCAN

[35] J. Tang, Z. Chen, A.-c. Fu, and D. Cheung, "Enhancing Effectiveness of Outlier Detections for Low Density Patterns," in *Advances in Knowledge Discovery and Data Mining*. vol. 2336, M.-S. Chen, P. Yu, and B. Liu, Eds., ed: Springer Berlin Heidelberg, 2002, pp. 535-548.

[36] M. M. Breunig, H.-P. Kriegel, R. T. Ng, #246, and r. Sander, "LOF: identifying density-based local outliers," presented at the Proceedings of the 2000 ACM SIGMOD international conference on Management of data, Dallas, Texas, USA, 2000.

[37] H.-P. Kriegel, P. Kr, #246, ger, E. Schubert, and A. Zimek, "LoOP: local outlier probabilities," presented at the Proceedings of the 18th ACM conference on Information and knowledge management, Hong Kong, China, 2009.

ประวัติผู้เขียนวิทยานิพนธ์

นายธนชัย จิระจันทร์ เชื้อชาติ ไทย สัญชาติไทย ภูมิลำเนาอยู่ในจังหวัดกรุงเทพมหานคร สำเร็จการศึกษา วิศวกรรมศาสตรบัณฑิต สาขาวิชา วิศวกรรมคอมพิวเตอร์ จากคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2555