การดูแลและวิเคราะห์สมรรถนะของระบบคลาวด์ด้วยการประมวลผลแบบกระจาย

DISTRIBUTED CLOUD HEALTH MONITORING & REAL-TIME ANALYSIS

พีรวิชญ์ ประพันธ์วิทยา
อภิรัตน์ พุทธรักษา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

การดูแลและวิเคราะห์สมรรถนะของระบบคลาวด์ด้วยการประมวลผลแบบกระจาย

DISTRIBUTED CLOUD HEALTH MONITORING & REAL-TIME ANALYSIS

พีรวิชญ์ ประพันธ์วิทยา

อภิรัตน์ พุทธรักษา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

ปริญญานิพนธ์ปีการศึกษา 2557

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การดูแลและวิเคราะห์สมรรถนะของระบบคลาวด์ด้วยการประมวลผลแบบกระจาย

DISTRIBUTED CLOUD HEALTH MONITORING & REAL-TIME ANALYSIS

ผู้จัดทำ

1. นายพีรวิชญ์ ประพันธ์วิทยา รหัสนักศึกษา 54010944
2. นายอภิรัตน์ พุทธรักษา รหัสนักศึกษา 54011491

..... อาจารย์ที่ปรึกษา

(ดร.อรทัย สังข์เพ็ชร)

..... อาจารย์ที่ปรึกษาร่วม

(ดร.อักฤทธิ์ สังข์เพ็ชร)

การดูแลและวิเคราะห์สมรรถนะของระบบคลาวด์ด้วยการประมวลผลแบบกระจาย

นายพีรวิชญ์	ประพันธ์วิทยา	54010944
นายอภิรัตน์	พุทธรักษา	54011491
ดร.อรทัย	สังข์เพ็ชร	อาจารย์ที่ปรึกษา
ดร.อักฤทธิ์	สังข์เพ็ชร	อาจารย์ที่ปรึกษาร่วม

ปีการศึกษา 2557

บทคัดย่อ

เนื่องจากปัจจุบันมีการใช้ระบบประมวลผลแบบคลาวด์กันอย่างแพร่หลายมากขึ้น ระบบนี้มักจะมีขนาดใหญ่และซับซ้อนเนื่องจากเครื่องแม่ข่ายนั้นสามารถให้บริการเครื่องคอมพิวเตอร์แบบเสมือนได้เป็นจำนวนมาก เมื่อมีความผิดพลาดหรือปัญหาเกิดขึ้นในระบบ การแก้ไขปัญหาได้อย่างรวดเร็วนั้นยังเป็นไปได้ยากเนื่องจากผู้ดูแลระบบจำเป็นต้องวิเคราะห์ข้อมูลเป็นจำนวนมาก ซึ่งทำให้เกิดความล่าช้าในการแก้ไขปัญหาที่เกิดขึ้น อีกทั้งผู้ดูแลระบบยังไม่สามารถคาดเดาถึงผลกระทบที่อาจจะเกิดขึ้นในส่วนอื่นๆ เนื่องจากปัญหาที่พบนี้ได้

เพื่อช่วยให้ผู้ดูแลระบบสามารถวิเคราะห์ข้อมูลในระบบได้รวดเร็วขึ้น จึงได้จัดทำโครงงานนี้ขึ้น โดยโครงงานนี้จะนำข้อมูลเกี่ยวกับเครื่องคอมพิวเตอร์เสมือนและเครื่องแม่ข่ายแต่ละตัว ซึ่งอาจจะเป็นข้อมูลเกี่ยวกับการใช้ทรัพยากรหรือความผิดพลาดและเหตุการณ์ต่างๆที่เกิดขึ้น มาทำการหาความสัมพันธ์ที่เป็นไปได้ต่างๆ โดยใช้ กระบวนการแก้ไขปัญหาทางคณิตศาสตร์และสถิติ กระบวนการเรียนรู้ของเครื่อง (Machine learning algorithms) หรือกระบวนการทำเหมืองข้อมูล (Data mining algorithms) ซึ่งความสัมพันธ์เหล่านี้จะช่วยผู้ดูแลระบบในการวิเคราะห์ข้อมูลที่น่าจะเกี่ยวข้องก่อนเพื่อช่วยลดเวลาและผลกระทบที่อาจจะเกิดขึ้น ระบบที่สร้างในโครงงานนี้จำเป็นต้องรองรับและวิเคราะห์ข้อมูลในปริมาณมาก จึงได้ออกแบบให้ระบบสามารถทำงานแบบกระจายซึ่งจะทำให้สามารถทำการประมวลผลแบบขนานได้

Distributed Health Cloud Monitoring & Real-Time Analysis

Mr. Peerawit	Praphanwittaya	54010944
Mr. Apirat	Puttaraksa	54011491
Dr. Orathai	Sangpetch	Advisor
Dr. Akkarit	Sangpetch	Co-Advisor

Academic Year 2014

Abstract

Cloud computing has been widely used to help simplify IT in organizations. The continuous growth of cloud computing results in larger and more complex datacenter, which may contain many thousands of physical and virtual devices. When an interruption or problem happens in the datacenter, administers or operators may take a significant amount of time to sieve through the large amount of data, e.g. resource usages and errors, in order to identify the problem. The longer it takes, the more degraded cloud services. To alleviate this problem, in this project we explore and study how to utilize existing statistical methods, machine learning algorithms and data mining techniques to learn and analyze the data we have collected from the devices. The selected algorithms will be implemented in a distributed manner in order to improve the scalability and reduce the analysis time.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ได้รับคำแนะนำ และคำปรึกษาเกี่ยวกับการวิจัยและการค้นคว้าเป็นอย่างดีจากอาจารย์ ดร. อรทัย สังข์เพ็ชร อาจารย์ที่ปรึกษาและอาจารย์ ดร.อักฤทธิ์ สังข์เพ็ชร อาจารย์ที่ปรึกษาร่วม ซึ่งทางคณะผู้จัดทำรู้สึกซาบซึ้งเป็นอย่างมากในความอนุเคราะห์จากอาจารย์ทั้งสองที่คอยให้การสนับสนุนในการทำปริญญานิพนธ์นี้เสมอมา อีกทั้งอาจารย์ รศ. ดร.เกียรติกูล เจียรนัยธนะกิจ ที่คอยช่วยเหลือ ให้ความรู้ที่เป็นประโยชน์ในการทำงานจนสำเร็จ รวมไปถึงห้องวิจัย ISAG (Information Security Advisory Group) ภายในภาควิชาวิศวกรรมคอมพิวเตอร์และห้องวิจัย CSAG (Computer system administrator group) ที่ได้อำนวยความสะดวก เป็นทั้งสถานที่ทำงานที่ศึกษาหาความรู้ประกอบการทำวิจัย

และปริญญานิพนธ์ฉบับนี้จะสำเร็จลงไม่ได้หากไม่ได้รับความอนุเคราะห์จากบริษัทอินเตอร์เน็ตประเทศไทย จำกัด (มหาชน) หรือ INET ที่ได้ให้ความช่วยเหลือในเรื่องของทรัพยากรในการทำวิจัยต่างๆ

คณะผู้จัดทำขอขอบพระคุณเป็นอย่างสูง และหวังอย่างยิ่งว่าปริญญานิพนธ์ฉบับนี้จะเป็นประโยชน์ต่อทุกท่าน และสามารถให้คำแนะนำแก่นักศึกษารุ่นต่อไปในอนาคตได้

พีรวิชญ์ ประพันธ์วิทยา

อภิรัตน์ พุทธรักษา

สารบัญ

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

โดยทั่วไปการให้บริการระบบการประมวลผลแบบคลาวด์ภายในศูนย์ข้อมูลจะมีการไหลเวียนของข้อมูลเป็นจำนวนมากตลอดเวลา เช่น ข้อมูลสถานะของเครื่องคอมพิวเตอร์เสมือนและเครื่องแม่ข่าย ข้อมูลการใช้งานทรัพยากรต่างๆ ของเครื่องต่างๆ ซึ่งเมื่อต้องการตรวจสอบการทำงาน ณ ขณะใดขณะหนึ่งของระบบจึงมีความจำเป็นจะต้องทำการวิเคราะห์ข้อมูลต่างๆ ที่มีความซับซ้อนเป็นจำนวนมากและเนื่องจากมนุษย์ซึ่งโดยปกติแล้วจะไม่สามารถวิเคราะห์ข้อมูลจำนวนมากได้ในระยะเวลาอันสั้น จึงทำให้ผู้ดูแลระบบจึงไม่สามารถนำข้อมูลต่างๆ นั้นมาใช้ให้เกิดประโยชน์ได้ทันท่วงที

เนื่องจากปัญหาในขั้นต้นนั้นจึงจำเป็นที่จะต้องมีระบบที่สามารถนำข้อมูลเกี่ยวกับเครื่องต่างๆ ในระบบการประมวลผลแบบคลาวด์มาวิเคราะห์หาความสัมพันธ์ระหว่างเครื่อง เพื่อช่วยผู้ดูแลระบบในการตรวจสอบการทำงานและตรวจสอบข้อผิดพลาดหรือปัญหาต่างๆ ที่เกิดขึ้นในระบบ ซึ่งความสัมพันธ์นี้สามารถเปลี่ยนแปลงได้ตลอดเวลา ในปัจจุบันนั้นมีเครื่องมือหรือระบบที่ช่วยในการตรวจสอบต่างๆ มากมายแต่เครื่องมือเหล่านั้นจะต้องมีค่าใช้บริการต่างๆ และเมื่อมีข้อมูลเข้ามาเป็นจำนวนมากเครื่องมือเหล่านั้นไม่สามารถประมวลผลข้อมูลได้ในเวลาอันสั้น อีกทั้งยังไม่สามารถที่จะหาความสัมพันธ์ระหว่างเครื่องต่างๆ ภายในระบบได้

1.2 วัตถุประสงค์ของโครงงาน

1) ผู้พัฒนาได้ศึกษาและเลือกใช้กระบวนการแก้ไขปัญหาทางคณิตศาสตร์และสถิติกระบวนการเรียนรู้ของเครื่อง (Machine learning algorithms) หรือกระบวนการทำเหมืองข้อมูล (Data mining algorithms) มาวิเคราะห์และหาความสัมพันธ์ต่างๆ ของข้อมูลที่ได้จากเครื่องคอมพิวเตอร์เสมือนและเครื่องแม่ข่าย
2) นำความสัมพันธ์ที่ได้จากข้อ 1 มาประยุกต์ในการแจ้งเตือนความผิดปกติและผลกระทบที่อาจจะเกิดขึ้น สืบเนื่องจากเครื่องคอมพิวเตอร์เสมือนที่กำลังมีปัญหา

3) ระบบที่ถูกพัฒนาขึ้นถูกออกแบบให้รองรับข้อมูลจากแหล่งอื่นๆ เช่น ข้อมูลเกี่ยวกับระบบจัดเก็บข้อมูล ข้อมูลเกี่ยวกับระบบเครือข่าย เพื่อทำให้การวิเคราะห์ข้อมูลมีความสมบูรณ์ยิ่งขึ้นในอนาคต และระบบได้ทำการประมวลผลแบบกระจายเพื่อให้การวิเคราะห์เป็นไปอย่างรวดเร็วขึ้น

1.3 ขอบเขตของโครงงาน

1) ระบบที่ถูกพัฒนาขึ้นสามารถนำเข้าข้อมูลจาก VMware vCenter Server
2) ระบบที่ถูกพัฒนาขึ้นสามารถหาความสัมพันธ์ต่างๆ ของเครื่องภายในระบบระบบคลาวด์ได้โดยอัลกอริธึม Pearson's Correlation, Self-Organizing Map และ Naïve Bayes แล้วนำความสัมพันธ์ที่หาได้นี้มาประยุกต์ใช้ในการวิเคราะห์ปัญหาที่เกิดขึ้นในกับเครื่องคอมพิวเตอร์ภายในระบบคลาวด์
3) ระบบที่ถูกพัฒนาขึ้นใช้การประมวลผลแบบกระจายเพื่อให้การประมวลผลข้อมูลเป็นไปได้อย่างรวดเร็วขึ้น

1.4 วิธีการดำเนินการ

1) ศึกษาการใช้งานโปรแกรม VMware vSphere Client
2) ศึกษาการเขียนโปรแกรมเพื่อติดต่อกับ VMware vCenter
3) ทำการพัฒนาโปรแกรมดึงข้อมูลจาก VMware vCenter
4) ออกแบบระบบโดยรวม
5) ทำการติดตั้งและศึกษาการใช้งาน Apache Hadoop
6) พัฒนาโปรแกรมต้นแบบในส่วนตรวจสอบสถานะเพื่อใช้งานกับ MapReduce Framework ของ Apache Hadoop
7) ศึกษาทฤษฎีและการใช้งานของ Pearson's Correlation
8) ทำการพัฒนาโปรแกรมต้นแบบในส่วนของการหาความสัมพันธ์ของเครื่องคอมพิวเตอร์เสมือนภายในระบบโดยใช้ Pearson's Correlation
9) ศึกษาทฤษฎีและการใช้งานของ Self-Organizing Map
10) ทำการพัฒนาโปรแกรมต้นแบบในส่วนของการหาความสัมพันธ์ของเครื่องคอมพิวเตอร์แบบเสมือนภายในระบบโดยใช้ Self-Organizing Map
11) ศึกษาทฤษฎีและการใช้งานของ Naïve Bayes

12) ทำการพัฒนาโปรแกรมต้นแบบในส่วนของการหาความสัมพันธ์ของเครื่องคอมพิวเตอร์แบบเสมือนภายในระบบโดยใช้ Naïve Bayes
13) ทำการทดลองเพื่อศึกษาความสามารถของแต่ละอัลกอริทึมเพื่อเลือกใช้ให้เหมาะแต่ละสถานการณ์เพื่อให้เกิดความแม่นยำมากขึ้น
14) ทำการพัฒนาระบบทุกๆ ส่วนให้สามารถทำงานแบบกระจายและสามารถตอบสนองการทำงานแบบทันทีทันใดได้ (Real-Time)
15) ทำการทดสอบระบบโดยรวม
16) สรุปผลและจัดทำรูปเล่มรายงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) สามารถนำโครงงานนี้ไปใช้ในการวิเคราะห์และหาความสัมพันธ์ระหว่างเครื่องภายในระบบประมวลผลแบบคลาวด์โดยใช้อัลกอริธึม Pearson's Correlation และ Self-Organizing Map ซึ่งการวิเคราะนี้จะช่วยบอกผลกระทบที่จะเกิดขึ้นเนื่องจากปัญหาที่เกิดขึ้นได้ ทำให้การแก้ไขหรือบรรเทาปัญหาเป็นไปได้รวดเร็วขึ้น
2) สามารถนำโครงงานนี้ไปช่วยตรวจสอบการทำงานของเครื่องคอมพิวเตอร์เสมือนหรือเครื่องแม่ข่ายที่อาจจะผิดปกติโดยใช้อัลกอริธึม Naïve Bayes ซึ่งในโครงงานนี้ผู้พัฒนาให้อัลกอริธึมเรียนรู้รูปแบบการใช้ทรัพยากรเวลาถูกโจมตีแบบ Denial of Service (DOS)
3) เพื่อให้การตรวจจับความผิดปกติหรือการหาผลกระทบที่อาจจะเกิดจากปัญหาที่พบเป็นไปได้เร็วขึ้นและลดความเสียหายที่อาจจะเกิดขึ้น ผู้พัฒนาได้นำ Apache Spark และ Apache Hadoop มาประมวลผลให้เร็วขึ้นเวลาที่มีข้อมูลเป็นจำนวนมาก

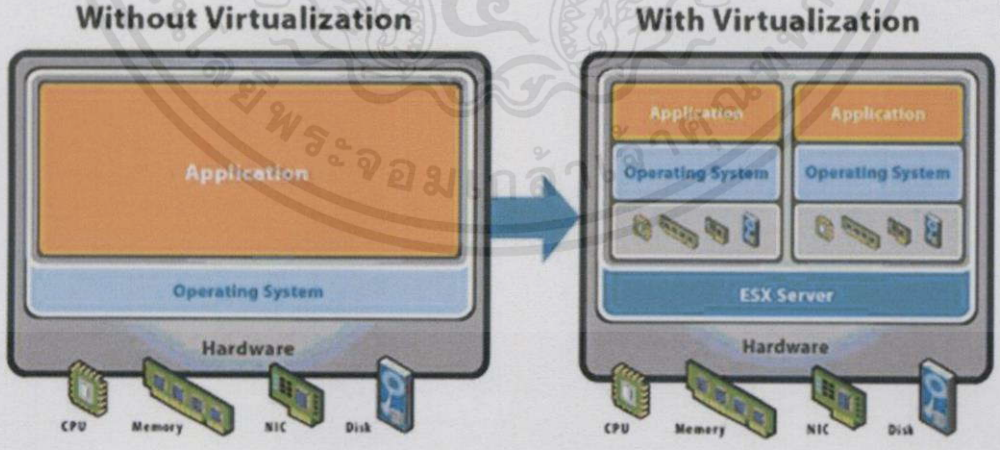
บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 การประมวลผลแบบกระจาย (Distributed computing)

การประมวลผลแบบกระจายเป็นการนำเครื่องคอมพิวเตอร์หลายๆ เครื่องมาเชื่อมต่อกันด้วยระบบเครือข่ายเป็นระบบคอมพิวเตอร์ทำให้คอมพิวเตอร์แต่ละเครื่องสามารถส่งคำสั่งหรือให้บริการไปยังเครื่องอื่นๆ ภายในระบบเดียวกันได้อย่างรวดเร็ว ทำให้สามารถกระจายภาระการประมวลผลไปยังเครื่องคอมพิวเตอร์ต่างๆ และนำผลลัพธ์ที่ได้จากแต่ละเครื่องมารวมกัน ซึ่งทำให้เพิ่มประสิทธิภาพในการประมวลผลโดยรวม รวมทั้งยังสามารถลดจำนวนข้อมูลที่ส่งผ่านเครือข่ายอีกด้วย [1][14]

2.2 เวอร์ชวลไลเซชั่น (Virtualization Technology)

เวอร์ชวลไลเซชั่นเป็นเทคโนโลยีสำหรับการจำลองสภาพแวดล้อมให้เสมือนมีคอมพิวเตอร์หลายเครื่องทำงานอยู่ในคอมพิวเตอร์หลักเครื่องเดียว โดยอาศัยการทำงานของซอฟต์แวร์ด้านเวอร์ชวลไลเซชั่นเป็นตัวจัดการในเรื่องต่างๆ เช่น การใช้งานฮาร์ดแวร์, ระบบปฏิบัติการ, ระบบเครือข่าย, ระบบไฟล์ และไฟล์วอลล์ เป็นต้น [2]



รูปที่ 2.1 ภาพแสดงการเปรียบเทียบระหว่างระบบปกติกับระบบที่ใช้งาน Virtualization

จากรูปที่ 2.1 จะเห็นได้ว่าในการใช้งานเครื่องคอมพิวเตอร์โดยไม่มีเทคโนโลยีเวอร์ชัวไลเซชั่นเข้ามาช่วย ทำให้มีข้อจำกัดคือเครื่องคอมพิวเตอร์หนึ่งเครื่องสามารถใช้ระบบปฏิบัติการได้เพียงระบบปฏิบัติการเดียวในเวลาเดียวกัน ซึ่งเป็นการใช้ทรัพยากรของเครื่องคอมพิวเตอร์อย่างไม่คุ้มค่า โดยเทคโนโลยีเวอร์ชัวไลเซชั่นได้เข้ามาแก้ไขปัญหานี้ทำให้สามารถใช้งานทรัพยากรของเครื่องคอมพิวเตอร์ได้อย่างคุ้มค่ามากยิ่งขึ้น

2.3 ระบบประมวลผลแบบคลาวด์ (Cloud computing)

เป็นรูปแบบบริการใช้ทรัพยากรคอมพิวเตอร์ร่วมกันกับผู้อื่น (เช่น เครือข่าย เครื่องเซิร์ฟเวอร์ เครื่องบันทึกข้อมูล ระบบซอฟต์แวร์ และบริการอื่นที่เกี่ยวข้อง) ผ่านเครือข่าย ผู้ใช้สามารถปรับเพิ่มหรือลดทรัพยากรคอมพิวเตอร์ได้ง่ายและรวดเร็วตามความต้องการของผู้ใช้ การบริการคลาวด์ (Cloud Services) มีคุณสมบัติสำคัญ (Essential Characteristics) ห้าประการคือ บริการด้วยตัวเองเมื่อต้องการ (On-demand self-service), เข้าถึงทรัพยากรคอมพิวเตอร์ได้ในวงกว้างผ่านเครือข่าย (Broad network access), ทรัพยากรถูกรวบรวมจากที่ต่าง ๆ (Resource pooling), มีความยืดหยุ่นและปรับตัวได้รวดเร็ว (Rapid elasticity), การบริการที่วัดได้ (Measured service) [3]

2.4 การทำเหมืองข้อมูล (Data Mining)

การทำเหมืองข้อมูล (Data Mining) เป็นกระบวนการวิเคราะห์ข้อมูลเพื่อค้นหารูปแบบ (Pattern) หรือความสัมพันธ์ (Relationship) ระหว่างข้อมูลรวมถึงสร้างรูปแบบการคาดเดา (Predictive Modeling) โดยใช้ขั้นตอนทางสถิติ (Statistic) การเรียนรู้ของเครื่อง (Machine learning) และการรู้จำแบบ (Pattern recognition) [19][20]

2.4.1 Pearson's Correlation

เป็นการหาความสัมพันธ์ทางสถิติเชิงเส้นระหว่างตัวแปรสองตัวแปร (x,y) โดยค่าที่ได้จากการหาความสัมพันธ์อยู่ในช่วง -1 ถึง 1 [16]

$$R = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}} \quad (2.1)$$

R = สัมประสิทธิ์สหสัมพันธ์เพียร์สัน

คุณสมบัติของค่า R

1) ค่าของ R นั้นไม่ขึ้นกับหน่วยในการวัดของตัวแปรทั้งสอง ถ้า x เป็นความสูง ซึ่งอาจมีหน่วยเป็นเมตร ถ้าหากเปลี่ยนหน่วยเป็นนิ้ว หรือเซนติเมตรแล้ว ค่าสหสัมพันธ์ที่คำนวณได้จะไม่เปลี่ยนแปลง หรือ y คืออุณหภูมิ อาจจะเป็นองศาเซลเซียสหรือองศาฟาเรนไฮท์ ค่าสหสัมพันธ์ที่คำนวณได้ก็ยังคงเดิม
2) ค่าของ R อยู่ระหว่าง -1.00 ถึง 1.00 ถ้าหากค่า R มีค่ามากกว่า 0 แล้วจะเป็นความสัมพันธ์ทางบวก ถ้าหากมีค่าน้อยกว่า 0 แล้วจะมีความสัมพันธ์ทางลบ ตัวแปรจะสัมพันธ์กันสูง ปานกลางหรือต่ำมีเกณฑ์ดังนี้ [21]

สัมพันธ์กันสูง	$R \geq 0.70$ หรือ $R \leq -0.70$
สัมพันธ์กันปานกลาง	$0.30 < R < 0.70$ หรือ $-0.70 < R < -0.30$
สัมพันธ์กันต่ำ	$-0.30 \leq R \leq 0.30$

2.5 กระบวนการเรียนรู้ของเครื่อง (Machine learning)

การเรียนรู้ของเครื่องเป็นสาขาหนึ่งของวิชาปัญญาประดิษฐ์ (Artificial intelligence) ที่เกี่ยวข้องกับการพัฒนาเทคนิควิธีเพื่อให้คอมพิวเตอร์สามารถเรียนรู้ โดยเน้นวิธีการสร้างโปรแกรมคอมพิวเตอร์จากการวิเคราะห์ชุดข้อมูล การเรียนรู้จึงเกี่ยวข้องอย่างมากกับวิชาสถิติศาสตร์ การเรียนรู้ของเครื่องถูกใช้เพื่อเพิ่มประสิทธิภาพในการแก้ปัญหาด้านต่าง ๆ เช่น การสร้างให้คอมพิวเตอร์สามารถแยกแยะวัตถุ เสียง หรือตัวอักษรได้ หรือจำแนกข้อมูลจำนวนมากที่ไม่สามารถทำได้โดยมนุษย์ เป็นต้น ซึ่งลักษณะทั่วไปของการเรียนรู้ของเครื่องเป็นการสร้างขั้นตอนวิธี (Algorithms) หรือโปรแกรมคอมพิวเตอร์ จากการให้ข้อมูลฝึก (Training data) สำหรับสอนให้คอมพิวเตอร์เรียนรู้เพื่อให้ได้สมมติฐาน (Hypothesis) ในการนำมาใช้แยกแยะวัตถุอื่นได้ [17]

การเรียนรู้ของเครื่องจักร แบ่งได้ 4 ประเภทตามลักษณะการใช้ข้อมูลฝึก

1. การเรียนรู้แบบมีครู (Supervised Learning) เทคนิคการเรียนรู้ของเครื่องประเภทนี้ต้องใช้การเรียนรู้จากข้อมูลฝึกที่มีการใส่ฉลาก (Label) ให้กับข้อมูลฝึกไว้แล้ว เพื่อให้คอมพิวเตอร์จดจำรูปแบบและได้สมมติฐานเพื่อทำงานกับข้อมูลในภายหน้าได้ ตัวอย่างเทคนิคประเภทนี้ได้แก่ การเรียนรู้แบบตัวจำแนกแบบเบย์สอย่างง่าย และการเรียนรู้แบบต้นไม้ตัดสินใจ เป็นต้น
2. การเรียนรู้แบบไม่มีครู (Unsupervised Learning) ใช้ข้อมูลฝึกที่ไม่มีการใส่ฉลากให้กับข้อมูล และเรียนรู้โดยการนำข้อมูลไปผ่านกระบวนการหาความคล้ายคลึงของชุดข้อมูล

จนกระทั่งได้กลุ่มข้อมูลที่จัดเป็นประเภทอย่างเหมาะสม เทคนิคประเภทนี้ได้แก่ การแบ่งกลุ่ม (Clustering)

3. การเรียนรู้แบบมีครูบางส่วน (Semi Supervised Learning) ใช้การเรียนรู้จากข้อมูลฝึกที่มีการใส่ฉลากเพียงบางส่วนจากข้อมูลฝึกทั้งหมด สำหรับส่วนที่ไม่มีฉลากนั้นจะใช้กระบวนการเรียนรู้เพื่อใส่ฉลากและปรับความถูกต้องให้กับการเรียนรู้ต่อไป เทคนิคประเภทนี้ได้แก่ Expectation-Maximization (EM) algorithm
4. การเรียนรู้แบบการเสริมกำลัง (Reinforcement Learning) เทคนิคนี้ต่างจาก 3 เทคนิคข้างต้น โดยไม่จำเป็นต้องมีข้อมูลฝึกเพื่อใช้สอน แต่ใช้ลักษณะการควบคุมการเรียนรู้ (Control Learning) โดยที่ระหว่างโปรแกรมคอมพิวเตอร์ทำงานจะมีการให้คำแนะนำเสมือนเป็นการให้รางวัลเมื่อผลการทำงานถูกต้อง หรือทำโทษเมื่อผลการทำงานไม่ถูกต้อง และระบบจะทำการเรียนรู้ด้วยการทดลองต่าง ๆ เอง เพื่อให้การทำงานไปในแนวทางที่ถูกต้อง เช่น การนำเทคนิคนี้ใช้ควบคุมการเดินของหุ่นยนต์

ในโครงงานนี้ผู้พัฒนาได้เลือกศึกษา Self-Organizing Map (SOM)

2.5.1 Self-Organizing Map (SOM)

SOM คือ โครงข่ายประสาทเทียม (Neural Network) แบบไม่มีผู้สอนซึ่งประกอบไปด้วย เวกเตอร์ข้อมูลรับเข้า x ขนาด n มิติ และโหนดของนิวรอนขนาดสองมิติ ซึ่งแต่ละโหนดของนิวรอนประกอบไปด้วยเวกเตอร์น้ำหนักแทนด้วย W_i โดยที่ขนาดของเวกเตอร์น้ำหนักจะต้องมี ขนาดเท่ากับเวกเตอร์ข้อมูลรับเข้า [13]

$$x(t) = \{x_1, x_2, x_3, x_4, \dots\dots, x_n\} \quad (2.2)$$

$$w(t) = \{w_1, w_2, w_3, w_4, \dots\dots, w_n\} \quad (2.3)$$

กระบวนการเรียนรู้ของ SOM นั้นเกิดขึ้นจากการปรับเวกเตอร์น้ำหนักที่มีต่อเวกเตอร์ข้อมูลรับเข้า ซึ่งในแต่ละรอบ t ของการเรียนรู้จะทำการสุ่มเลือกเวกเตอร์ข้อมูลรับเข้า x(t) มาทำการเปรียบเทียบกับโหนดทุกโหนดเพื่อที่จะหาโหนดชนะสำหรับเวกเตอร์ข้อมูลรับเข้านั้น ฟังก์ชั่นที่ใช้ในการเปรียบเทียบหาระยะห่างของข้อมูลคือฟังก์ชั่นวัดระยะห่างแบบยูคลิดเดียน (Euclidean distance) ระยะห่างระหว่างเวกเตอร์ข้อมูลรับเข้ากับโหนดน้อยที่สุดจะเป็นโหนดชนะดังสมการที่ 2.4

$$\|x(t) - w_c(t)\| = \min_i \|x(t) - w_i(t)\| \quad (2.4)$$

จากนั้นเวกเตอร์น้ำหนักของโหนดชนะจะถูกปรับค่า โดยการปรับจะพิจารณาจากผลต่างของเวกเตอร์ข้อมูลรับเข้าและเวกเตอร์น้ำหนัก โดยอัตราการเรียนรู้แต่ละรอบจะค่อย ๆ ลดลง นอกจากการเรียนรู้ที่เกิดขึ้นที่โหนดชนะแล้วโหนดใกล้เคียง (Neighborhood nodes) จะเกิดการเรียนรู้ด้วย โดยเวกเตอร์น้ำหนักของโหนดใกล้เคียงจะปรับค่าให้เข้าใกล้กับเวกเตอร์ข้อมูลรับเข้าเดียวกัน เพื่อเพิ่มโอกาสให้ข้อมูลรับเข้าใหม่ที่ใกล้เคียงกับข้อมูลรับเข้าเดิมสามารถที่จะมีโหนดชนะใหม่ใกล้กับโหนดชนะเดิมได้ สมการในการปรับค่าสามารถแสดงได้ดังสมการที่ 2.5

$$w_i(t+1) = w_t(t) + \alpha(t)h_{ci}(t)[x(t) - w_i(t)] \quad (2.5)$$

เมื่อ x(t) คือเวกเตอร์ข้อมูลรับเข้า

$w_i(t)$ คือเวกเตอร์น้ำหนักของโหนด i

$\alpha(t)$ คืออัตราการเรียนรู้ในแต่ละรอบ ซึ่งแสดงเป็นสมการเชิงเส้นได้ดังสมการ 2.6

t คือรอบในการเรียนรู้

$$\alpha(t+1) = \alpha(0)e^{-\frac{t}{T}} \quad (2.6)$$

เมื่อ T คือจำนวนรอบทั้งหมด

t คือจำนวนรอบที่ปัจจุบัน

$h_{ci}(t)$ คือฟังก์ชั่นที่ใช้ในการกำหนดขนาดของโหนดใกล้เคียงดังแสดงตามสมการที่ 2.7

$$h_{ci}(t) = e^{-\frac{\|r_i - r_c t\|^2}{2\sigma^2(t)}} \quad (2.7)$$

เมื่อ $\|r_c - r_i\|$ คือระยะห่างระหว่างโหนด i กับโหนดชนะ c

$\sigma(t)$ คือรัศมีของบริเวณโหนดใกล้เคียงโดยปกติรัศมีจะค่อยๆ ลดลงตามจำนวนรอบในการเรียนรู้ ดังแสดงตามสมการที่ 2.8

$$\sigma(t+1) = 1 + (\sigma(t) - 1) \times \frac{T-1}{T} \qquad (2.8)$$

2.5.2 Naïve Bayes

เป็นกระบวนการเรียนรู้ของเครื่องที่ใช้หลักการของความน่าจะเป็นซึ่งมีพื้นฐานมาจากทฤษฎีของเบย์ (Bayes Theorem) [6] เพื่อเข้ามาช่วยในการเรียนรู้ จุดมุ่งหมายเพื่อสร้างรูปแบบในการทำนายให้อยู่ในรูปของความน่าจะเป็นสำหรับแยกแยะประเภทข้อมูลโดยใช้ความรู้ที่เคยเรียนรู้มาแล้ว (Prior Knowledge) ซึ่งเป็นไปดังสมการที่ 2.2 [18]

$$P(C|A) = \frac{P(A|C) * P(C)}{P(A)} \qquad (2.2)$$

- Posterior probability หรือ P(C|A) คือ ค่าความน่าจะเป็นที่ข้อมูลที่มีคุณสมบัติ A จะมีคลาส C
- Likelihood หรือ P(A|C) คือ ค่าความน่าจะเป็นที่ข้อมูลการสอน ที่มีคลาส C และมีคุณสมบัติ A โดยที่ $A = a_1 \cap a_2 \ldots \cap a_M$ โดยที่ M คือจำนวนคุณลักษณะในชุดข้อมูลการสอน
- Prior probability หรือ P(C) คือ ค่าความน่าจะเป็นของคลาส C

2.9 VMware Management Tool

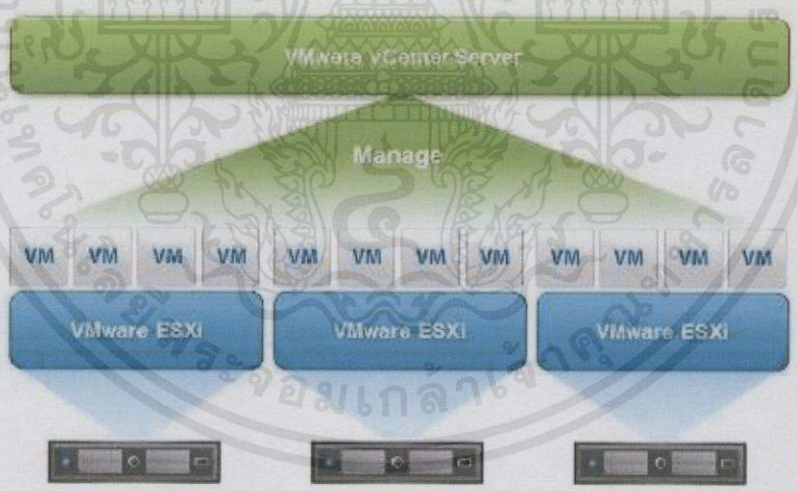
2.9.1 VMware ESXi

VMware ESxi เป็นไฮเปอร์ไวเซอร์ (Hypervisor) ซึ่งทำหน้าที่ในการควบคุมและจัดสรรการใช้งานทรัพยากรต่างๆ เช่น การใช้งานระบบไฟล์, การใช้งานระบบเครือข่าย, การใช้งานหน่วยประมวลผลหรือหน่วยความจำเป็นต้น ซึ่งทำให้คอมพิวเตอร์เสมือนหลายๆ เครื่องสามารถทำงานได้พร้อมๆ กันภายในคอมพิวเตอร์เครื่องเดียว

2.9.2 VMware vCenter Sever

VMware vCenter Server เป็นโปรแกรมจัดการเครื่องคอมพิวเตอร์เสมือนต่างๆ ภายในระบบคลาวด์ ใช้ในการควบคุมเครื่องคอมพิวเตอร์ (Physical Server) และเครื่องคอมพิวเตอร์เสมือน (Virtual machine) ทำให้ผู้ดูแลระบบสามารถจัดการดูแลหรือควบคุมเครื่องคอมพิวเตอร์หลายๆ เครื่องผ่าน VMware vCenter ได้

นอกเหนือจากทำหน้าที่ในการควบคุมเครื่องคอมพิวเตอร์หลายๆ ตัวไว้ด้วยกันแล้ว VMware vCenter Server ยังเป็นตัวดำเนินงานทำการต่างๆ และเป็นตัวกำหนดค่าในการใช้งานหรือสั่งให้ทำงานได้ เช่น การทำ VMware High Availability (HA) ที่ช่วยในการย้ายคอมพิวเตอร์เสมือนไปยังเครื่องคอมพิวเตอร์เครื่องอื่นๆ เมื่อเกิดปัญหาที่คอมพิวเตอร์ขึ้นเพื่อช่วยลดระยะเวลาที่เครื่องคอมพิวเตอร์เสมือนไม่สามารถให้บริการได้ (Downtime) เป็นต้น และอีกภาระหน้าที่หนึ่งของ VMware vCenter Server คือการเก็บค่าสถิติการใช้งานโดยรวมของระบบ รวมถึงการแจ้งเตือนความผิดปกติทั้งหลายให้แก่ผู้ดูแลระบบได้ทราบอย่างทันท่วงที ซึ่งลักษณะโครงสร้างการทำงานเป็นไปดังรูปที่ 2.2 [8]



รูปที่ 2.2 แสดงโครงสร้างการทำงานของ VMware vCenter Server

2.9.3 VMware vSphere Web Service API

เป็นชุดคำสั่งในการเขียนโปรแกรมเพื่อติดต่อกับ VMware vCenter

2.10 Apache Hadoop

Apache Hadoop เป็นโครงการ Open source Software พัฒนาขึ้นด้วยภาษา JAVA สำหรับสร้างระบบประมวลผลแบบกระจายที่มีความเสถียรสูง ทำให้สามารถประมวลผลชุดข้อมูลขนาดใหญ่ภายในระบบคอมพิวเตอร์โดยถูกออกแบบมาให้สามารถเพิ่มจำนวนเครื่องคอมพิวเตอร์ที่ใช้ในการประมวลผลได้ [7]

2.10.1 Hadoop Distributed File System (HDFS)

เป็นระบบแฟ้มข้อมูลแบบกระจายที่ใช้ใน Apache Hadoop โดยข้อมูลที่มีขนาดใหญ่จะถูกแบ่งเป็นส่วนเล็กๆ เรียกว่าบล็อกของข้อมูลจากนั้นทำการเก็บข้อมูลแบบกระจายภายในคลัสเตอร์ (Cluster) ซึ่งทำให้การทำงานของ MapReduce สามารถทำงานกับข้อมูลส่วนเล็กๆ ในข้อมูลขนาดใหญ่ทำให้มีการประมวลผลที่เร็วยิ่งขึ้น [4]

2.10.2 MapReduce

MapReduce เป็น Framework ในการเขียนโปรแกรมแบบขนานเพื่อช่วยในการประมวลผลที่มีชุดข้อมูลจำนวนมาก โดยใช้เครื่องคอมพิวเตอร์หลายๆ เครื่องช่วยกันทำงาน การทำงานของ MapReduce ถูกแบ่งออกเป็น 2 ขั้นตอนหลักๆ คือ ขั้นตอน Map ทำหน้าที่ในการแปลงข้อมูลรับเข้าให้อยู่ในรูปแบบของ key-value ที่สามารถนำไปใช้งานต่อได้ซึ่งไม่จำเป็นต้องอยู่ในรูปแบบเดียวกันกับข้อมูลรับเข้า โดยข้อมูลรับเข้าหนึ่งข้อมูลอาจถูกแปลงเป็นหลายคู่ของ key-value จากนั้นแต่ละคู่ของ key-value จะถูกส่งต่อไปในส่วนของขั้นตอน Reduce ซึ่งทำหน้าที่ในการประมวลผลลัพธ์ของแต่ละ key เดียวกันเพื่อให้ได้ผลลัพธ์ตามที่ต้องการ ซึ่งทำให้สามารถประมวลผลชุดข้อมูลขนาดใหญ่ได้ในเวลาอันสั้น [5]

บทที่ 3
การออกแบบและการพัฒนา

3.1 ภาพรวมของระบบ

ในการตรวจสอบสถานะการทำงานของระบบประมวลผลแบบคลาวด์ ผู้ดูแลระบบจะต้องทำการตรวจสอบข้อมูลหลายๆ ชนิดเช่น การใช้งานซีพียู การใช้งานหน่วยความจำ หรือการรับส่งข่อมูลผ่านเครือข่าย เป็นต้น ผู้ดูแลระบบสามารถอ่านข้อมูลเหล่านี้ได้จากเครื่องมือต่างๆ เช่น VMware vCenter เป็นต้น

ระบบที่นำเสนอแบ่งออกเป็นสองส่วนหลักๆ คือ ส่วนที่ใช้ในการเก็บข้อมูลจากระบบประมวลผลแบบคลาวด์ และส่วนที่ใช้ในการวิเคราะห์และหาความสัมพันธ์ของข้อมูลเครื่องแม่ข่ายและเครื่องคอมพิวเตอร์เสมือนในระบบประมวลผลแบบคลาวด์ดังรูปที่ 3.1

รูปที่ 3.1 ภาพแสดงการติดต่อกันในระบบ

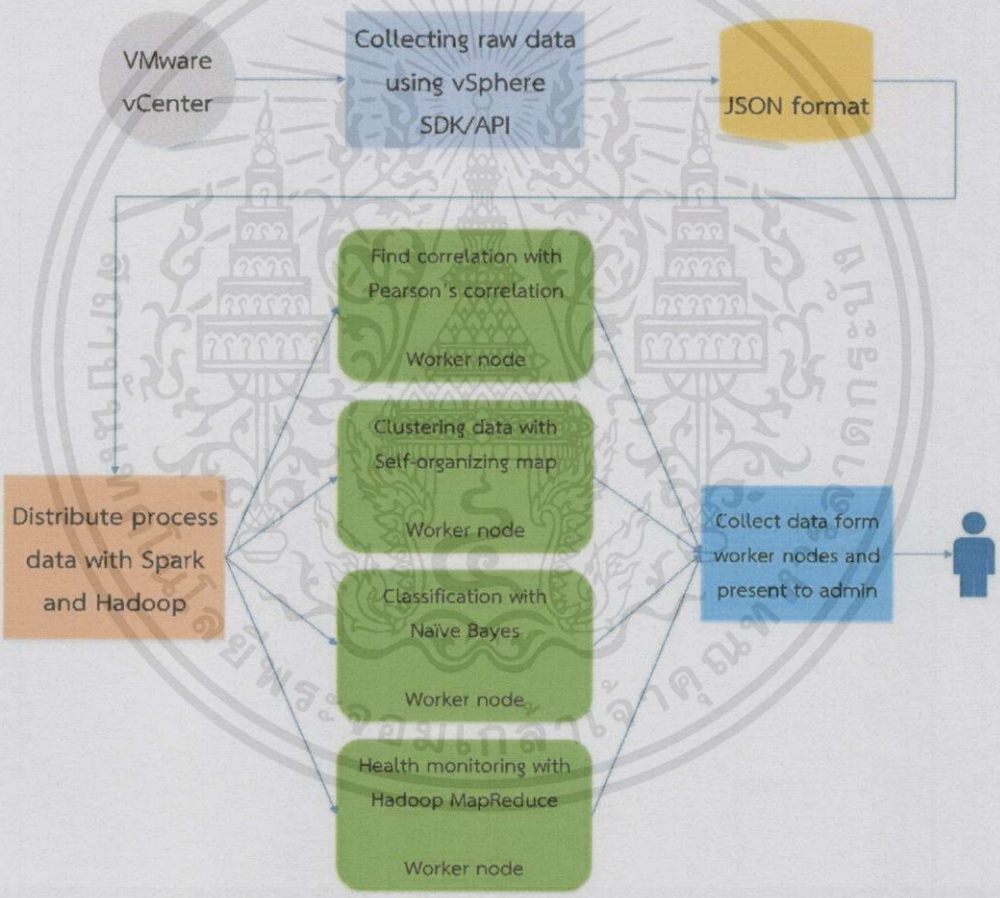
3.1.1 ส่วนการเก็บข้อมูล

ส่วนนี้ใช้ในการเก็บข้อมูลจากเครื่องคอมพิวเตอร์เสมือนในระบบประมวลผลแบบคลาวด์ โดยใช้VMware vSphere API ใน vSphere Web Services SDK [10] ซึ่งเป็นชุดคำสั่งในการติดต่อกับ VMware vCenter เพื่อทำการเก็บข้อมูลสถานะและข้อมูลการใช้ทรัพยากรต่างๆ ภายในระบบโดยใช้ภาษา Java ในการพัฒนา ซึ่งส่วนนี้ทำการเก็บข้อมูลที่ได้มาในรูปแบบของ JSON

3.1.2 ส่วนการวิเคราะห์และหาความสัมพันธ์ของข้อมูล

ส่วนนี้ทำหน้าที่วิเคราะห์ข้อมูลที่ได้มาจากส่วนแรกเพื่อหาความผิดปกติที่อาจจะเกิดขึ้นในระบบและหาความสัมพันธ์ระหว่างเครื่องคอมพิวเตอร์เสมือน โดยใช้การกระบวนการทำเหมืองข้อมูล (Data Mining), กระบวนการเรียนรู้ของเครื่อง (Machine Learning) และกระบวนการทางสถิติ ความสัมพันธ์ของเครื่องคอมพิวเตอร์แบบเสมือนที่วิเคราะห์หาได้สามารถช่วยระบุผลกระทบเนื่องจากปัญหาที่เกิดขึ้น ทำให้ผู้ดูแลระบบสามารถแก้ไขหรือบรรเทาปัญหาที่เกิดขึ้นได้รวดเร็วขึ้น

3.2 โครงสร้างในการพัฒนาระบบ



รูปที่ 3.2 โครงสร้างของระบบ

3.2.1 ส่วนการเก็บข้อมูล

3.2.1.1 Collecting Data

ผู้พัฒนาได้พัฒนาโปรแกรมจาวา (Java) ไปติดต่อกับ VMware vCenter Server เพื่อเก็บข้อมูลสถานะและข้อมูลการใช้ทรัพยากรของเครื่องคอมพิวเตอร์เสมือนในระบบประมวลผลแบบคลาวด์ โดยทำการติดต่อผ่าน VMware vSphere API ใน vSphere Web Services SDK

ข้อมูลที่ได้รับมาจาก vCenter ประกอบไปด้วยข้อมูลที่แสดงในตารางที่ 3.1

ตารางที่ 3.1 ตารางแสดงรายละเอียดของข้อมูลที่จัดเก็บ

ชื่อ	รายละเอียด
Alarm	ข้อมูลการแจ้งเตือนภายใน
Cluster compute resource	ข้อมูลรายละเอียดกลุ่มของเครื่องคอมพิวเตอร์ที่ให้บริการเครื่องคอมพิวเตอร์แบบเสมือน
Datacenter	ข้อมูลรายละเอียดของ Datacenter
Data store	ข้อมูลรายละเอียดของระบบไฟล์
Distributed virtual port group	ข้อมูลรายละเอียดของ Distributed virtual port
Event	ข้อมูลเหตุการณ์ที่เกิดขึ้น
Folder	ข้อมูลรายละเอียดกลุ่มของ object ที่มีชนิดเดียวกัน
Host	ข้อมูลรายละเอียดของคอมพิวเตอร์
Network	ข้อมูลรายละเอียดของเครือข่าย
Performance	ข้อมูลการใช้ทรัพยากรของเครื่องคอมพิวเตอร์เสมือน
Resource pool	ข้อมูลรายละเอียดของ Resource pool
Virtual application	ข้อมูลรายละเอียดของ Virtual application
Virtual machine	ข้อมูลรายละเอียดของเครื่องคอมพิวเตอร์เสมือน
VMware distributed virtual switch	ข้อมูลรายละเอียดของ distributed virtual switch

3.2.1.2 JSON Format Data

ส่วนนี้ทำหน้าที่แปลงรูปแบบของข้อมูลที่ได้จาก 3.2.1.1 ให้อยู่ในรูปแบบ JSON และนำไปจัดเก็บไว้ใน Hadoop Distributed File System (HDFS)

3.2.2 ส่วนการตรวจสอบสถานะและวิเคราะห์หาความสัมพันธ์

3.2.2.1 Distributed data processing

ส่วนนี้ทำหน้าที่ดึงข้อมูลที่จัดเก็บใน HDFS มาประมวลผลและวิเคราะห์ข้อมูลโดยใช้อัลกอริธึมที่เลือก โดยจะใช้โปรแกรมอะไร??? จะการทำงานไปยังโหนดย่อยๆ การทำงานในส่วนนี้ประกอบ 4 อย่าง

1. การหาความสัมพันธ์ทางสถิติโดยใช้อัลกอริธึม Pearson's correlation
2. การจัดกลุ่มของข้อมูลโดยใช้อัลกอริธึม Self-organizing map
3. การตรวจสอบการทำงานของเครื่องคอมพิวเตอร์เสมือนโดยใช้ Naïve Bayes
4. การค้นหาสถานะหรือการใช้ทรัพยากรของเครื่องคอมพิวเตอร์เสมือนโดยใช้ Hadoop และ MapReduce

3.2.2.2 Displaying result

ส่วนนี้นำข้อมูลผลลัพธ์ที่ได้จากการวิเคราะห์ข้อมูลใน 3.2.2.1 มาแสดงผลให้ผู้ดูแลระบบดูในรูปแบบที่ผู้ใช้งานสามารถทำความเข้าใจได้ง่ายโดยผ่านทางเว็บแอพพลิเคชั่น

บทที่ 4
การทดลองและผลการทดลอง

ผู้พัฒนาได้ออกแบบ 3 การทดลองเพื่อศึกษาและวัดความสามารถของอัลกอริธึมที่คัดเลือกมาว่าสามารถวิเคราะห์ข้อมูลที่ได้จากศูนย์ข้อมูลได้ดีเท่าไหร่

การทดลองที่ 1. การทดลองวิเคราะห์หาความสัมพันธ์โดยใช้ Pearson's Correlation Algorithm และ Self-Organizing Map (SOM) เพื่อวัดความถูกต้องของความสัมพันธ์ที่ทั้งสองอัลกอริธึมค้นพบ รวมทั้งวัดเวลาที่ใช้การประมวลผล

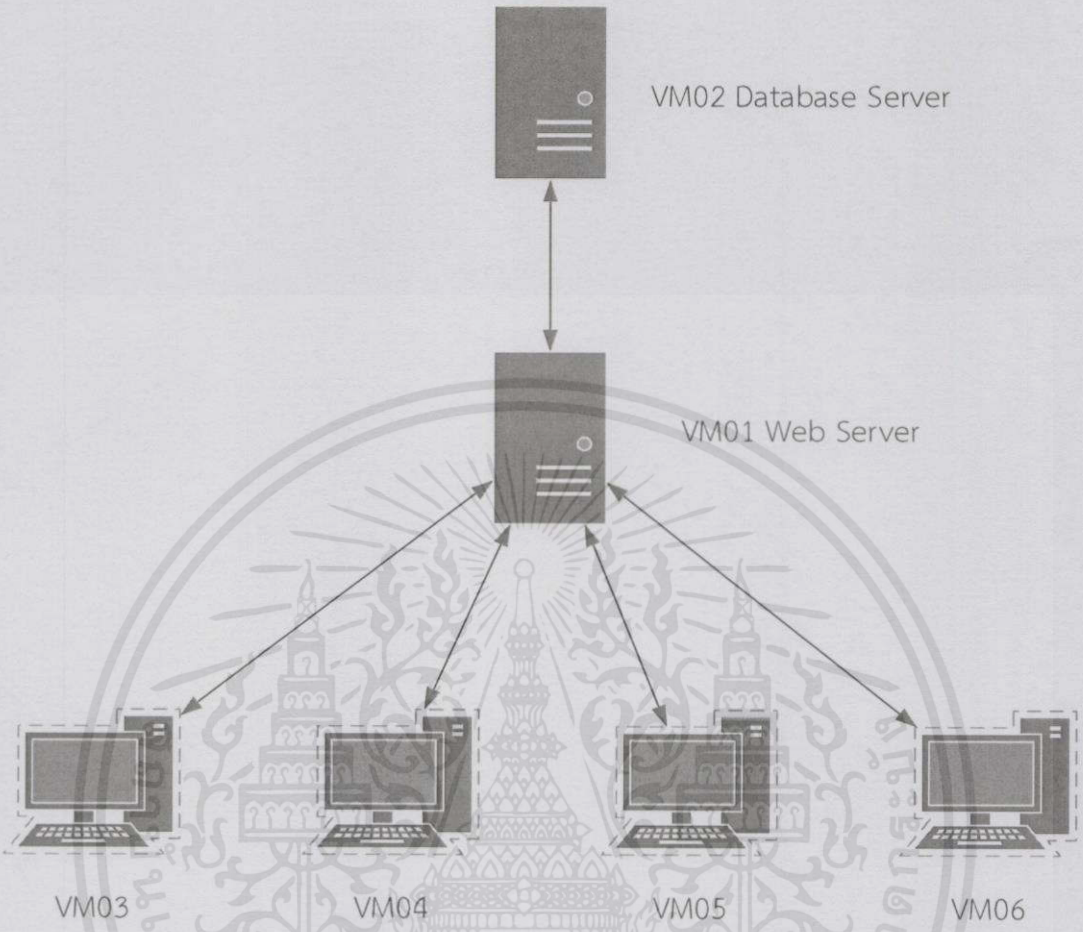
การทดลองที่ 2. ในการทดลองนี้ผู้พัฒนาทดลองใช้ Naïve Bayes ช่วยในการค้นหาเครื่องที่มีพฤติกรรมการใช้ทรัพยากรคล้ายกับการถูกโจมตีด้วย Denial of Service (DOS) และวัดความถูกต้องและความผิดพลาดในการค้นหา

การทดลองที่ 3. ในการทดลองนี้ผู้พัฒนาได้ใช้ Hadoop มาช่วยค้นหาข้อมูลเกี่ยวกับการใช้งานอุปกรณ์ เพื่อช่วยให้ผู้ใช้สามารถค้นหาข้อมูลสถิติที่สนใจเร็วขึ้น เช่น หาอุปกรณ์ที่มีการใช้ network bandwidth มากที่สุด 10 ตัวในขณะนี้

4.1 การวิเคราะห์หาความสัมพันธ์

4.1.1 ระบบที่ใช้ในการทดลอง

ระบบที่ใช้ในการทดลองประกอบไปด้วยเครื่องแม่ข่ายที่ให้บริการเว็บไซต์ (web server) , เครื่องแม่ข่ายที่ให้บริการฐานข้อมูล (database server) และเครื่องลูกข่ายจำนวน 4 เครื่อง ซึ่งเครื่องทั้งหมดนั้นเป็นเครื่องคอมพิวเตอร์แบบเสมือนโดยให้มีการเชื่อมต่อดังรูป 4.1 เครื่องคอมพิวเตอร์เสมือนทั้งหมดนี้อยู่บนเครื่องแม่ข่าย ESX 5.1



รูปที่ 4.1 ระบบที่ใช้ในการทดสอบ

4.1.2 ข้อมูลที่ใช้ในการทดลอง

โดยปกติแล้ว vCenter จะแสดงค่าข้อมูลการใช้ทรัพยกากรต่างๆ ของเครื่องคอมพิวเตอร์แบบเสมือนรวมถึงเครื่องแม่ข่ายทุก 20 วินาที เช่น การใช้งาน CPU, การใช้งานหน่วยความจำ, การใช้งานอุปกรณ์เก็บข้อมูล, การใช้งานเครือข่าย เป็นต้น แต่ในการทดลองนี้ จะพิจารณาเฉพาะข้อมูลการใช้งานเครือข่ายก่อน คือ

- Data receive rate (Network) in KBps (rate) Rolluptype (average)
- Data transmit rate (Network) in KBps (rate) Rolluptype (average)

4.1.3 การนำข้อมูลไปวิเคราะห์หาความสัมพันธ์

ในการวิเคราะห์หาความสัมพันธ์ของข้อมูลการใช้งานเครือข่ายนั้นจะใช้ข้อมูลที่เป็นการใช้ทรัพยากรที่เป็น time series โดยจะแบ่งครั้งละ 10 จุดซึ่งแต่ละจุดจะห่างกันทุกๆ 20

วินาที โดยทำการวิเคราะห์ทั้งหมด 30 ครั้ง จากนั้นจึงนำข้อมูลที่แบ่งนั้นไปหาค่า Pearson's Correlation และจัดกลุ่มของข้อมูลโดยใช้ Self-organizing map (SOM)

ในการหาคำนวณหาค่า Pearson's Correlation ผู้พัฒนาได้จับคู่ของข้อมูลระหว่าง Data receive rate ของเครื่องคอมพิวเตอร์เสมือนตัวหนึ่งและ Data transmit rate ของเครื่องคอมพิวเตอร์เสมือนของอีกตัวหนึ่ง จะทำแบบนี้กับทุกคู่เครื่องคอมพิวเตอร์เสมือนที่เป็นไปได้ทั้งหมด

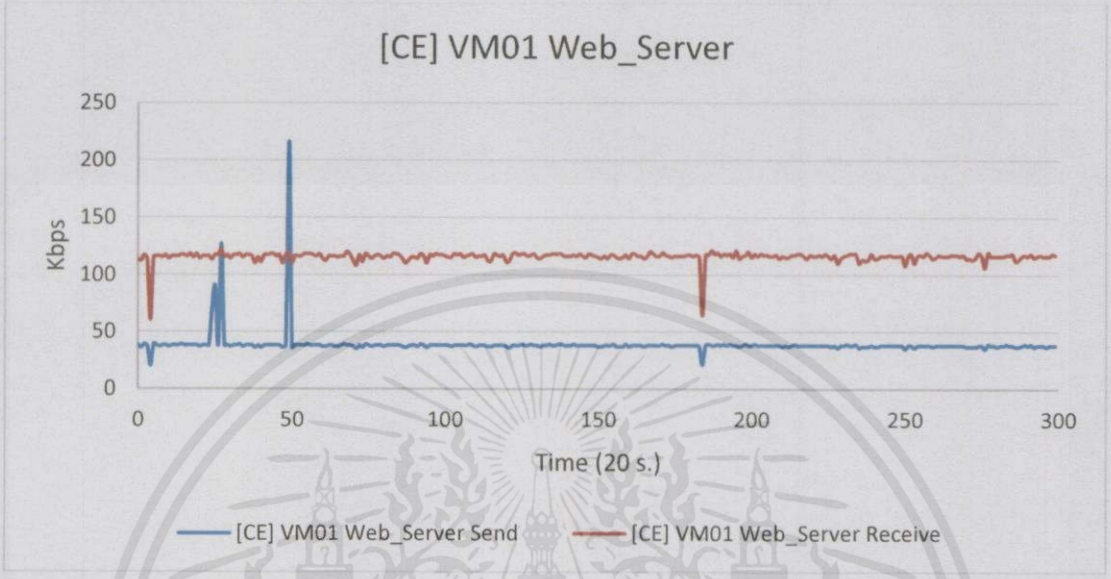
4.1.4 ผลการทดลอง

ผู้พัฒนาได้ทำการทดสอบทั้งหมด 4 แบบ ซึ่งในแต่ละแบบผู้พัฒนาได้เพิ่มจำนวนการร้องขอข้อมูล (web requests) ไปที่เครื่อง web server มากขึ้น เพื่อดูการทำงานของ Pearson's Correlation และ Self-Organizing Map ว่าสามารถค้นหาความสัมพันธ์ได้ถูกต้องหรือไม่และความถูกต้องมีสัมพันธ์กับขนาดของข้อมูลที่มาใช้ในการหาความสัมพันธ์หรือไม่ อย่างไร

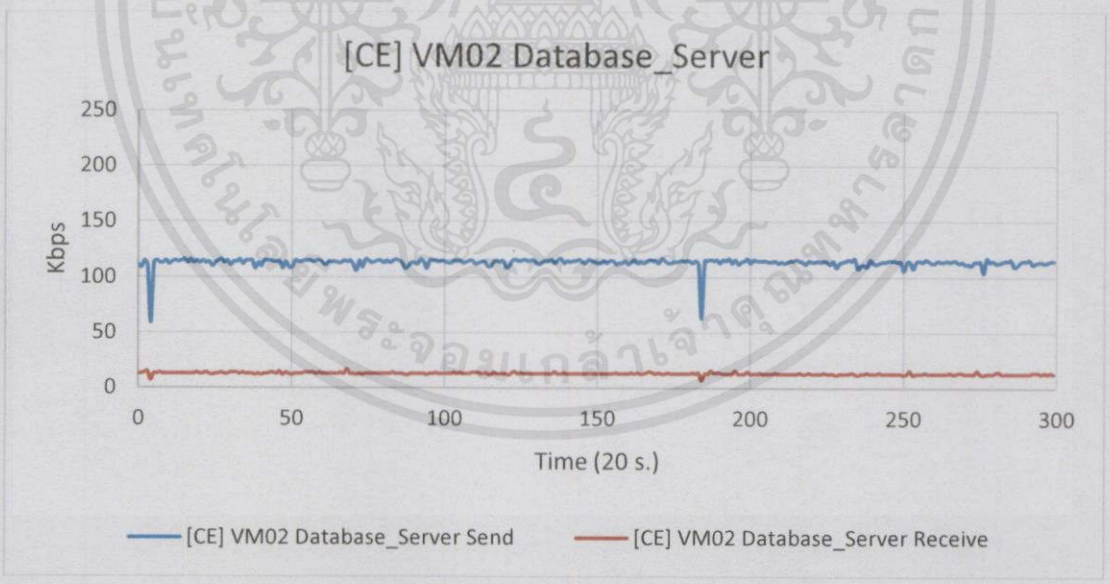
4.1.5 การทดลองแบบที่ 1

ในการทดลองนี้กำหนดให้เฉพาะเครื่อง VM03 ร้องขอข้อมูลหน้าเว็บไซต์ (web request) ไปยังเครื่อง Web server การร้องขอข้อมูลของ VM03 เป็นไปตามกราฟที่ 4.3 VM03 Send (เส้นสีฟ้า) หลังจากที่ Web server ได้รับการร้องขอข้อมูลก็จะทำการติดต่อไปยังเครื่อง database server เพื่อทำการหาค้นข้อมูลที่ต้องการ ปริมาณข้อมูลที่การรับส่งของเครื่อง web server เป็นไปตามกราฟที่ 4.1 และกราฟที่ 4.2 แสดงปริมาณข้อมูลเข้าออกของเครื่อง database server

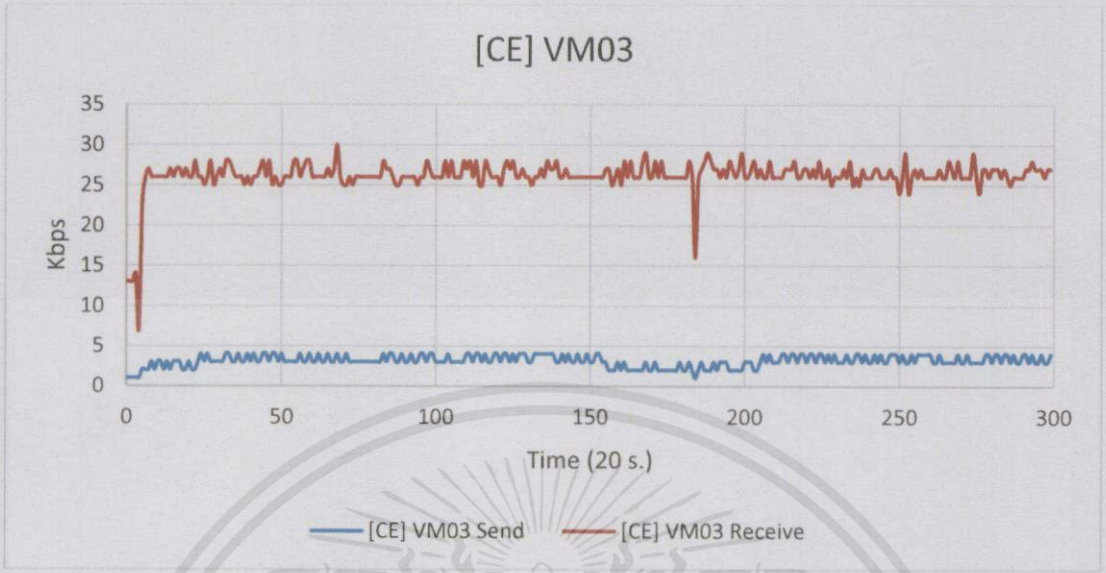
4.1.5.1 ข้อมูลการใช้งานเครือข่าย



กราฟที่ 4.1 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง Web server



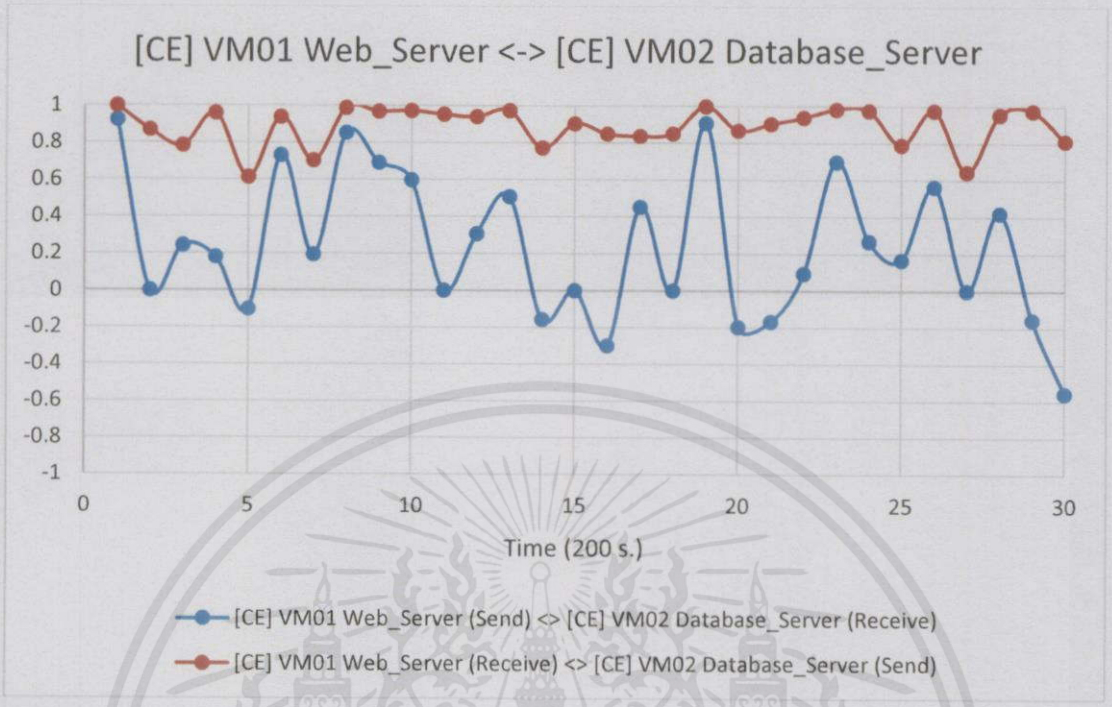
กราฟที่ 4.2 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง Database server



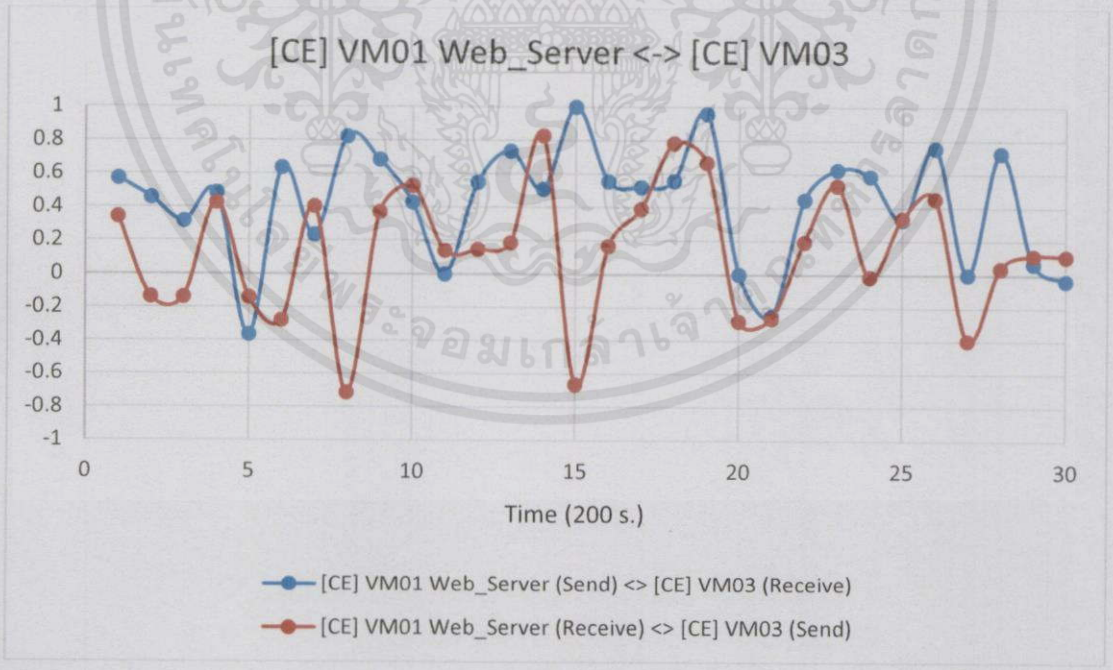
กราฟที่ 4.3 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง VM03

4.1.5.2 ผลการหาความสัมพันธ์โดยใช้ Pearson's Correlation

กราฟที่ 4.4 และ 4.5 แสดงค่า Pearson's Correlation ที่หาได้ซึ่งจะเห็นได้ว่าค่า correlation (เส้นสีฟ้า) ระหว่างข้อมูลการส่งออกของ web server และข้อมูลการรับของ database server มีการเปลี่ยนแปลงขึ้นลงอย่างมากทำให้ไม่สามารถหาแนวโน้มที่ชัดเจนได้ ทั้งนี้เนื่องมาจากปริมาณข้อมูลที่นำมาคิดมีปริมาณน้อยเกินไปเมื่อเทียบกับความเปลี่ยนแปลงของข้อมูลที่เกิดขึ้น ทำให้เราไม่สามารถหาความสัมพันธ์ระหว่าง web server และ database server ได้ ในทางกลับกันเมื่อเราใช้ปริมาณข้อมูลการรับของ web server และข้อมูลการส่งของ database server มาคิดค่า correlation ทำให้เห็นว่าสองเครื่องนี้มีแนวโน้มที่ชัดเจน นั่นคือค่า correlation ส่วนใหญ่เข้าใกล้ 1 เพราะการเปลี่ยนแปลงของปริมาณข้อมูลรับส่งเพียงเล็กน้อยก็ไม่ทำให้แนวโน้มมีการเปลี่ยนแปลงไป จากผลการทดลองนี้แสดงให้เห็นว่า web server และ database server มีความเกี่ยวเนื่องกัน



กราฟที่ 4.4 แสดงค่า Pearson's Correlation ระหว่าง Web server และ Database server



กราฟที่ 4.5 แสดงค่า Pearson's Correlation ระหว่าง Web server และ VM03

4.1.5.3 ผลการทดลองจัดกลุ่มโดยใช้ Self-organizing map (SOM)

SOM สามารถแบ่งเครื่องคอมพิวเตอร์เสมือนออกเป็น 2 กลุ่มหลัก คือ

กลุ่มที่ 1 ประกอบด้วย

- [CE] VM01 Web Server receive
- [CE] VM02 Database Server send

โดยการจัดกลุ่มให้ข้อมูลทั้งสองอย่างนี้อยู่ด้วยกันทั้งหมด 30 ครั้งหรือคิดเป็น 100% จากชุดข้อมูลทั้งหมด

กลุ่มที่ 2 ประกอบด้วย

- [CE] VM01 Web Server send
- [CE] VM03 receive

โดยการจัดกลุ่มให้ข้อมูลทั้งสองอย่างนี้อยู่ด้วยกันทั้งหมด 27 ครั้งหรือคิดเป็น 90% จากชุดข้อมูลทั้งหมด

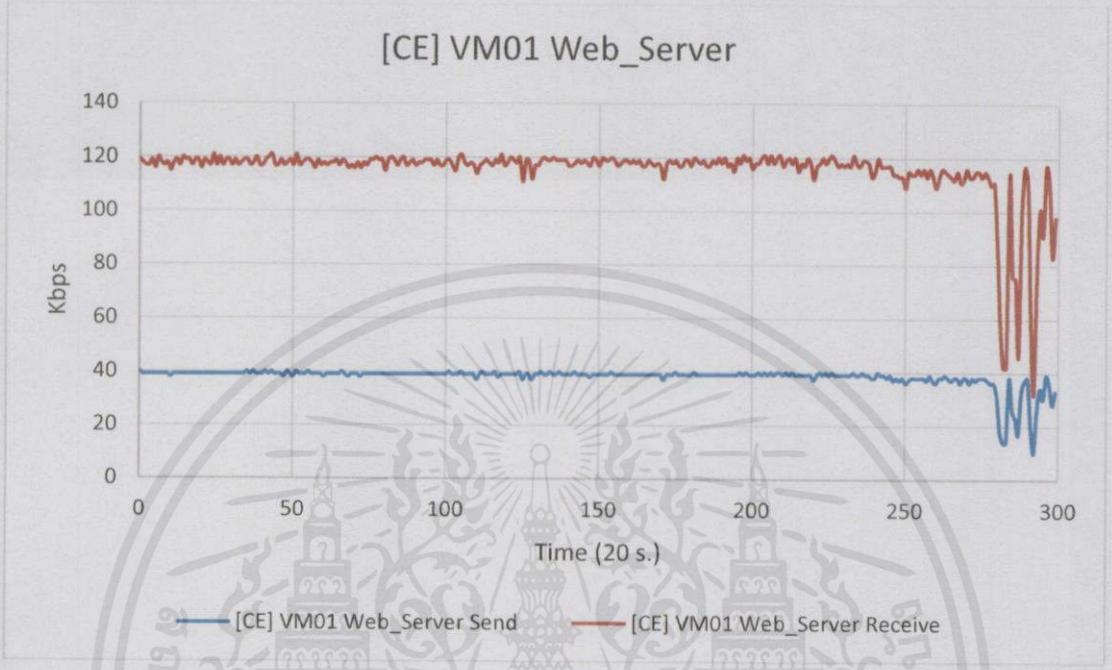
ส่วนข้อมูลอัตราการรับส่งข้อมูลของเครื่องคอมพิวเตอร์เสมือนอื่นๆ นั้น SOM จัดให้อยู่ในกลุ่มที่แตกต่างกันซึ่งแสดงให้เห็นว่าไม่มีความสัมพันธ์กับเครื่องอื่นๆ

เมื่อดูสองกลุ่มหลักที่ได้จากการใช้ SOM แล้ว ทำให้เห็นว่า SOM ยังสามารถจัดกลุ่มได้อย่างถูกต้อง เพราะ web server จะมีการติดต่อกับเครื่อง database server ทั้งสอง จึงถูกจัดให้อยู่ในกลุ่มเดียวกัน และเครื่อง VM03 เป็นเครื่องที่ติดต่อกับ web server ทั้งสองเครื่องนี้จึงอยู่ในกลุ่มเดียวกัน

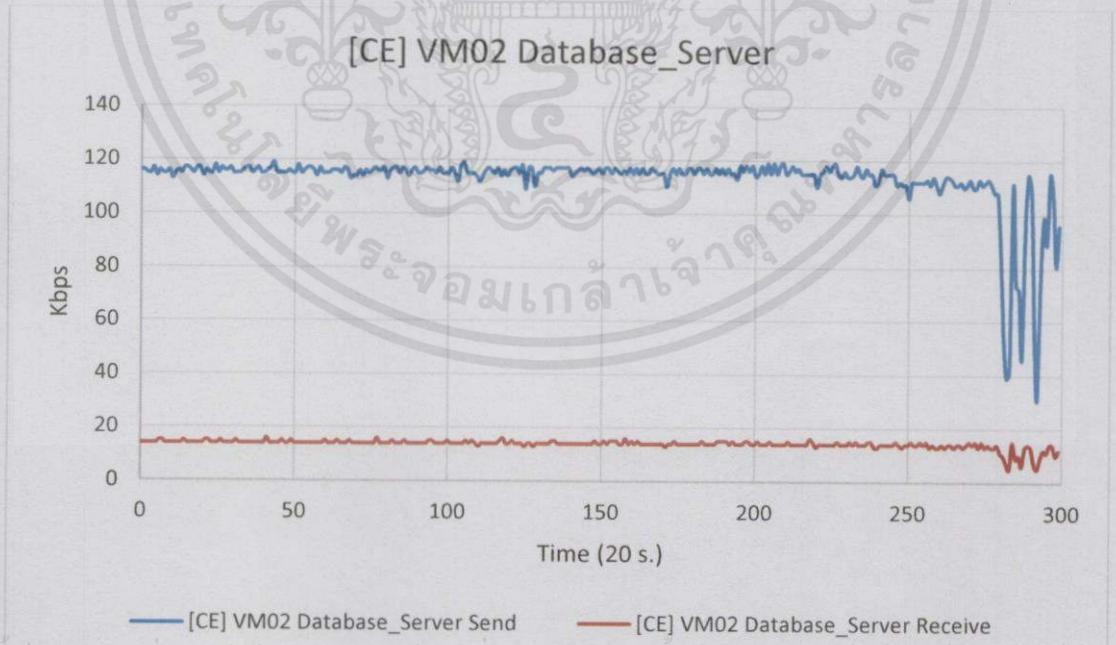
4.1.6 การทดลองแบบที่ 2

ในการทดลองนี้กำหนดให้เครื่อง VM03 และ VM04 ร้องขอข้อมูลหน้าเว็บไซต์ (web request) ไปยังเครื่อง web server โดยเครื่อง VM03 และ VM04 ได้มีการส่งข้อมูลไปหา web server ตามกราฟที่ 4.8 และ 4.9 (เส้นสีฟ้า) และกราฟที่ 4.6 (เส้นสีฟ้า) แสดงปริมาณข้อมูลรับส่งของเครื่อง web server ซึ่งจะเห็นได้ว่ามีปริมาณเพิ่มจากการทดลองแบบที่ 1 และกราฟที่ 4.7 แสดงปริมาณข้อมูลรับส่งของเครื่อง database server

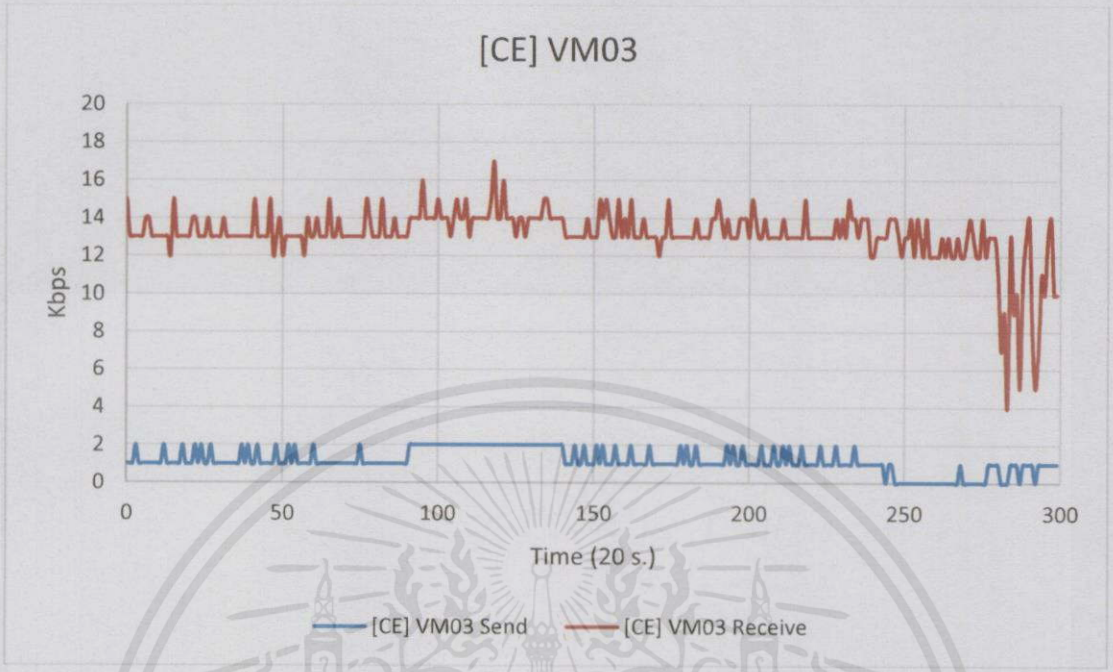
4.1.6.1 ข้อมูลการใช้งานเครือข่าย



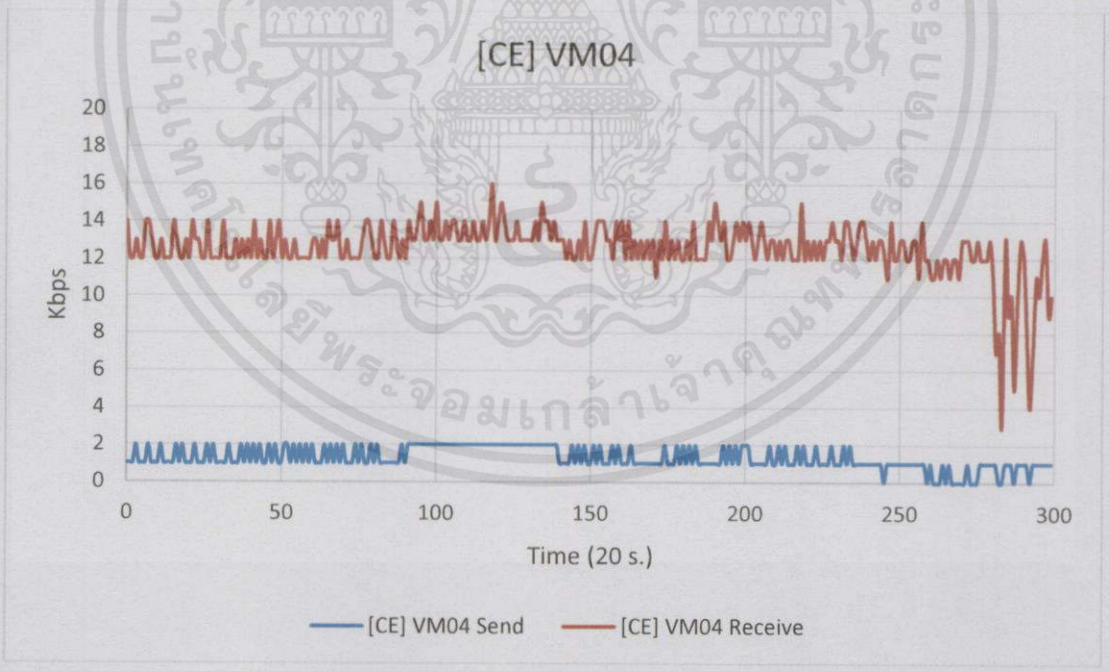
กราฟที่ 4.6 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง Web server



กราฟที่ 4.7 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง Database Server



กราฟที่ 4.8 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง VM03



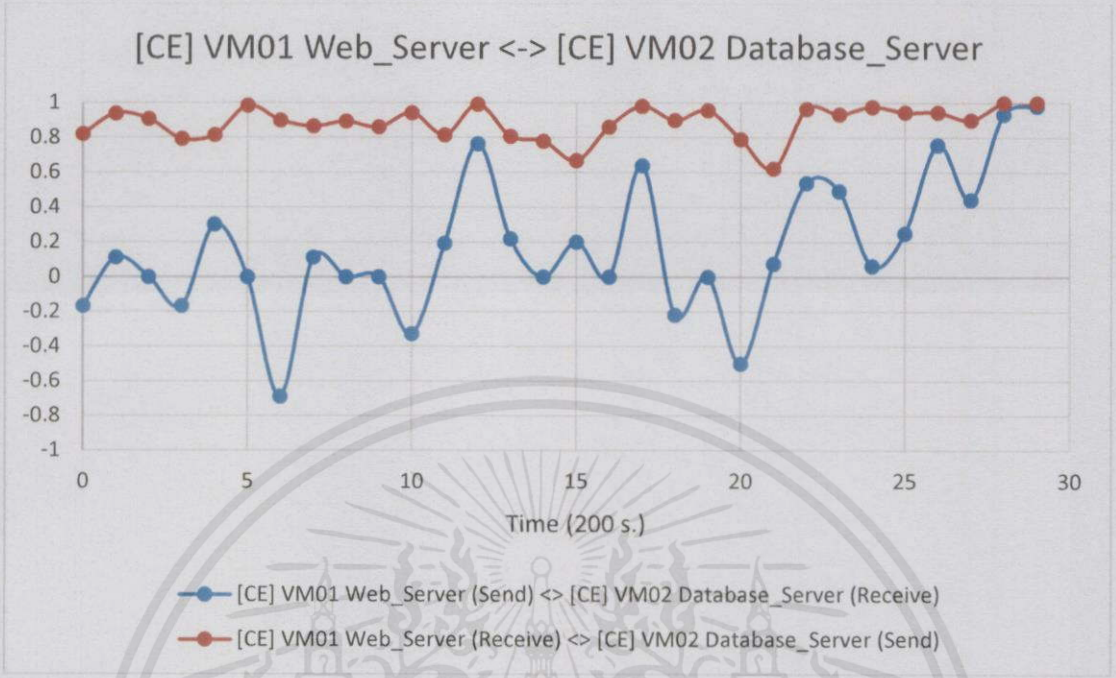
กราฟที่ 4.9 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง VM04

4.1.6.2 ผลการทดลองหาความสัมพันธ์โดยใช้ Pearson's Correlation

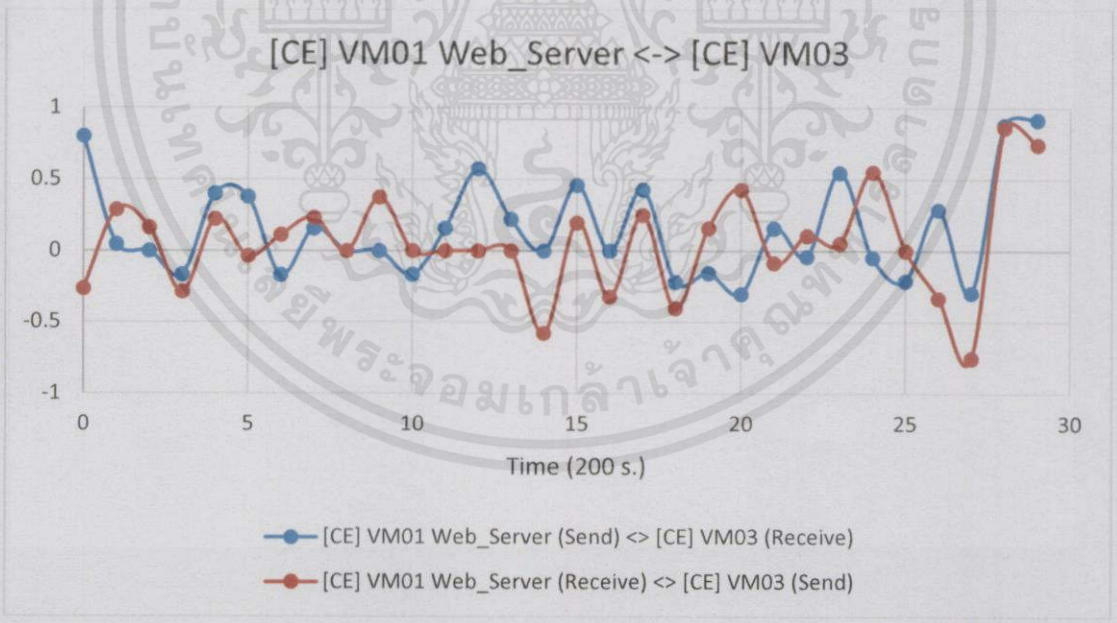
กราฟที่ 4.10, 4.11 และ 4.12 แสดงค่า correlation ที่ใช้ Pearson's Correlation ในการหาโดยใช้ข้อมูลของสามคู่ web server และ database server, web server และ VM03, web server และ VM04 จากกราฟจะเห็นได้ว่า ถ้าใช้ข้อมูลการส่งของ web server และการข้อมูลการรับของ database server (เส้นสีฟ้า ในกราฟที่ 4.10) จะไม่สามารถนำมาใช้หาความสัมพันธ์ระหว่างสองเครื่องนี้ได้ เพราะ web server มีการส่งข้อมูลไปหาทั้ง database server, VM03 และ VM04 ในขณะที่ข้อมูลการรับของ database server มาจากแค่ web server ทำให้รูปแบบการส่งข้อมูล (traffic pattern) มีความแตกต่างกันมากขึ้นเมื่อเทียบกับการทดลองแบบที่ 1

แต่เมื่อใช้ข้อมูลการรับของ web server และข้อมูลการส่งของ database server มาใช้การคิด correlation ทำให้การความเกี่ยวเนื่องกันของทั้งสองเครื่องดีขึ้น เพราะปริมาณการรับข้อมูลที่ web server ได้รับจาก VM03 และ VM04 มีปริมาณน้อยเมื่อเทียบกับการรับข้อมูลที่ web server ได้รับจาก database server ทำให้ค่า correlation ที่ได้ส่วนใหญ่มีค่าใกล้เคียง 1 ซึ่งแสดงให้เห็นว่า web server และ database server มีความสัมพันธ์กัน

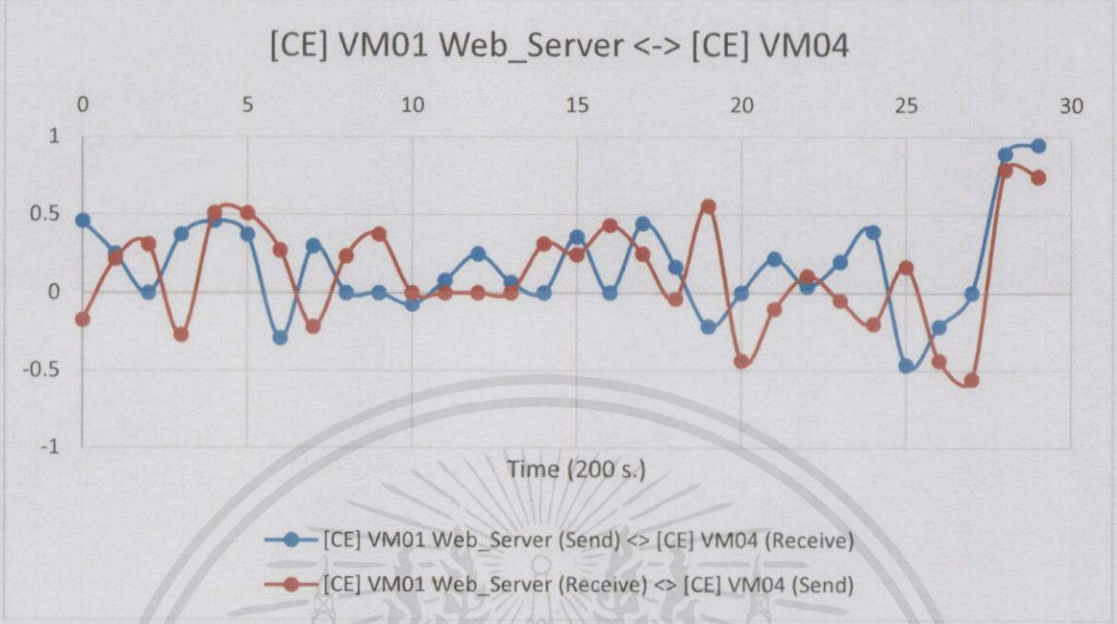
จากกราฟที่ 4.11 และ 4.12 ค่า correlation ที่หาได้ระหว่าง web server และ VM03, web server และ VM04 มีการเปลี่ยนแปลงขึ้นๆ ลงๆ และค่า correlation เข้าใกล้ 0 ซึ่งแปลว่า Pearson's Correlation ไม่สามารถหาความสัมพันธ์ระหว่างเครื่องเหล่านี้ได้ สาเหตุมาจากปริมาณข้อมูลที่รับส่งระหว่าง web server และ VM03, VM04 มีปริมาณน้อยเกินไป



กราฟที่ 4.10 แสดงค่า Pearson's Correlation ระหว่าง Web server และ Database server



กราฟที่ 4.11 แสดงค่า Pearson's Correlation ระหว่าง Web server และ VM03



กราฟที่ 4.12 แสดงค่า Pearson's Correlation ระหว่าง Web server และ VM04

4.1.6.3 ผลการทดลองจัดกลุ่มโดยใช้ Self-organizing map (SOM)

SOM จัดเครื่องคอมพิวเตอร์เสมือนออกเป็น 4 กลุ่มหลักๆ ดังต่อไปนี้

กลุ่มที่ 1 ประกอบด้วย

- [CE] VM01 Web Server receive
- [CE] VM02 Database Server send

โดยการจัดกลุ่มให้ข้อมูลทั้ง 2 อย่างนี้อยู่ด้วยกันทั้งหมด 30 ครั้งหรือคิดเป็น 100% จากชุดข้อมูลทั้งหมด

กลุ่มที่ 2 ประกอบด้วย

- [CE] [CE] VM02 Database Server receive
- [CE] VM03 receive
- [CE] VM04 receive

โดยการจัดกลุ่มให้ข้อมูลทั้ง 3 อย่างนี้อยู่ด้วยกันทั้งหมด 30 ครั้งหรือคิดเป็น 100% จากชุดข้อมูลทั้งหมด

กลุ่มที่ 3 ประกอบด้วย

- [CE] VM03 send
- [CE] VM04 send

โดยการจัดกลุ่มให้ข้อมูลทั้ง 2 อย่างนี้อยู่ด้วยกันทั้งหมด 30 ครั้งหรือคิดเป็น 100% จากชุดข้อมูลทั้งหมด

กลุ่มที่ 4 ประกอบด้วย

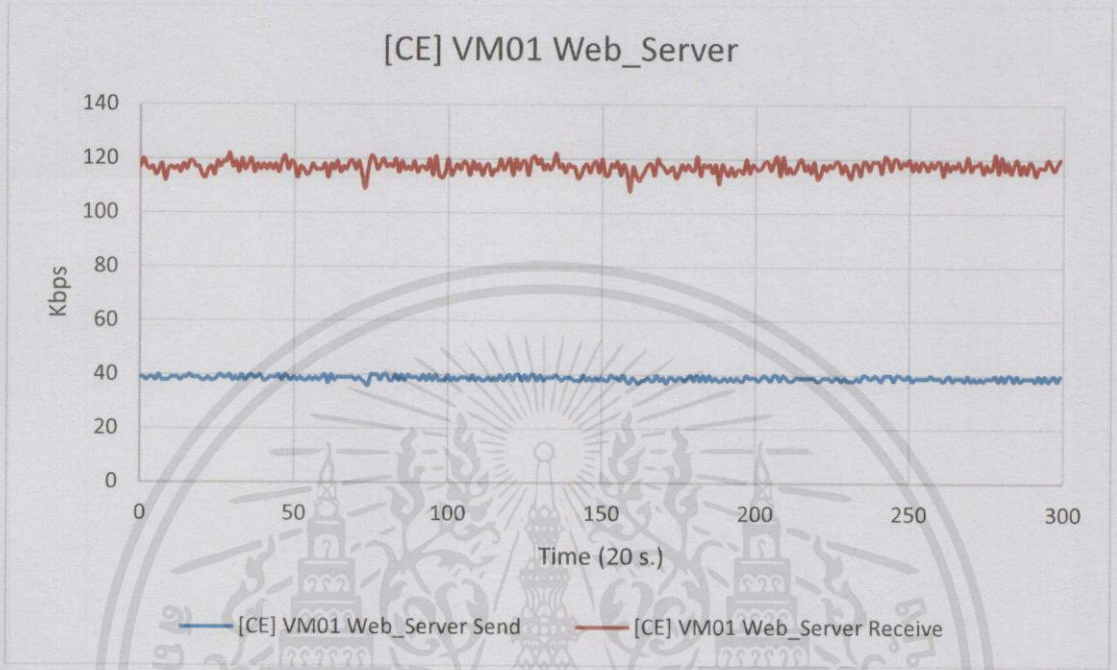
- [CE] VM01 Web_Server send

จากการทดลองนี้ ทำให้เห็นว่า SOM สามารถบอกได้ว่า Web server และ Database server นั้นมีความสัมพันธ์กันเนื่องจากการรับข้อมูลของเครื่อง web server และการส่งข้อมูลของ database server มีปริมาณใกล้เคียงกัน จึงถูกจัดให้อยู่ในกลุ่มเดียวกันคือกลุ่มที่ 1 SOM ยังสามารถจัดให้ VM03 และ VM04 อยู่ในกลุ่มเดียวกัน (กลุ่มที่ 3) เนื่องจากทั้งสองมีพฤติกรรมการส่งข้อมูลออกคล้ายกัน อย่างไรก็ดีบางกลุ่มที่ SOM จัดให้นั่นคือกลุ่มที่ 2, 4 ยังไม่ดีเท่าที่ควร SOM ควรจะจัดให้ VM03, VM04 และ Web Server อยู่ด้วยกัน ไม่ใช่อยู่กับ Database Server เหมือนในกลุ่มที่ 2

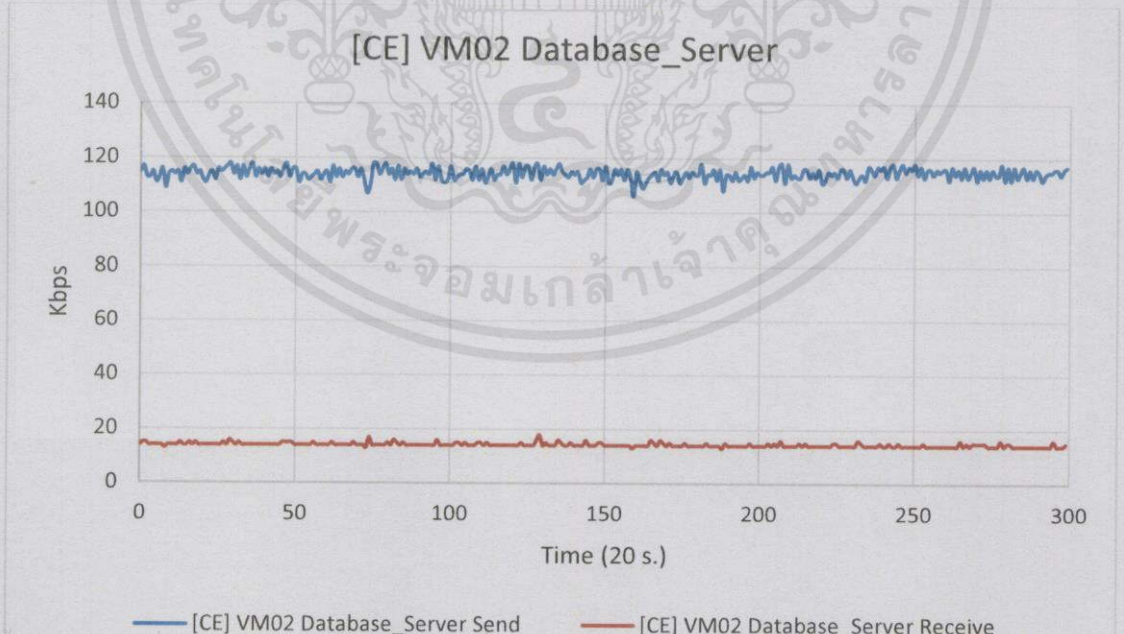
4.1.7 การทดลองแบบที่ 3

ในการทดลองนี้กำหนดให้เครื่อง VM03, VM04 และ VM05 ร้องขอข้อมูลหน้าเว็บไซท์ไปยังเครื่อง Web server ซึ่งปริมาณการรับส่งข้อมูลของ VM03, VM04 และ VM05 เป็นไปตามกราฟที่ 4.15, 4.16, และ 4.17 ส่วนปริมาณรับส่งข้อมูลของเครื่อง Web Server และ Database Server เป็นไปตามกราฟที่ 4.13 และ 4.14

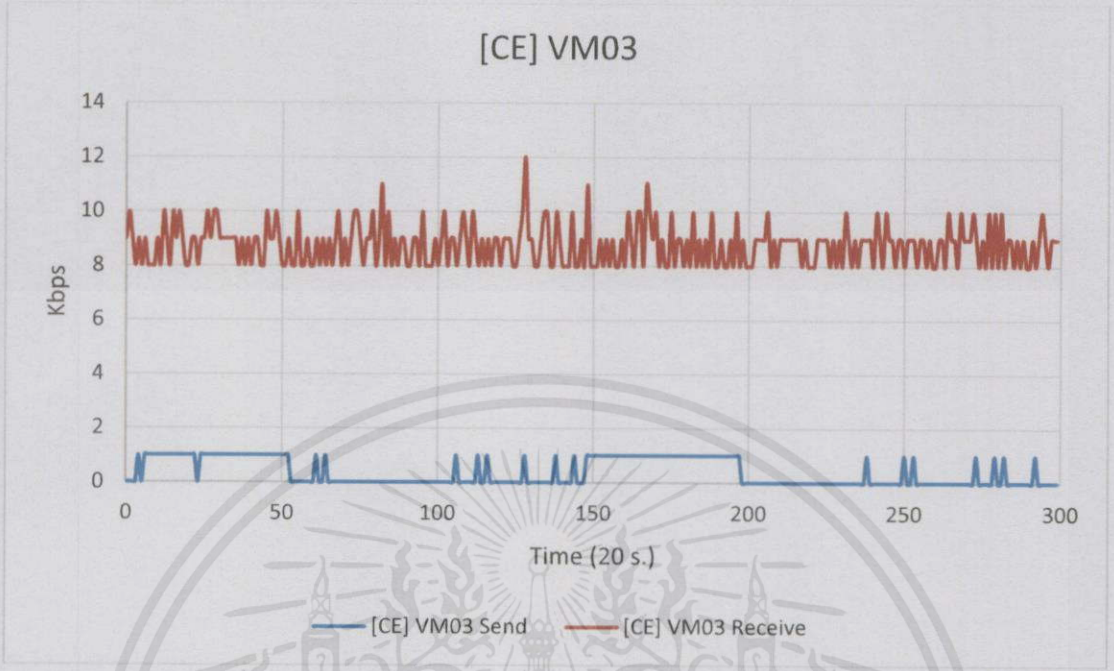
4.1.7.1 ข้อมูลการใช้งานเครือข่าย



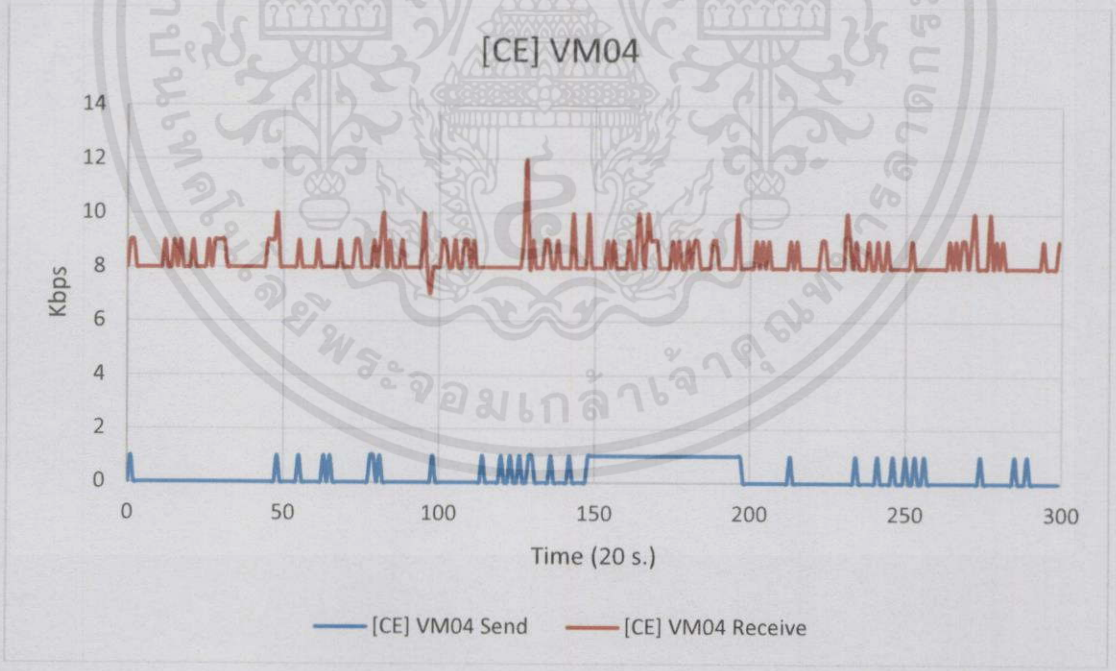
กราฟที่ 4.13 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง Web Server



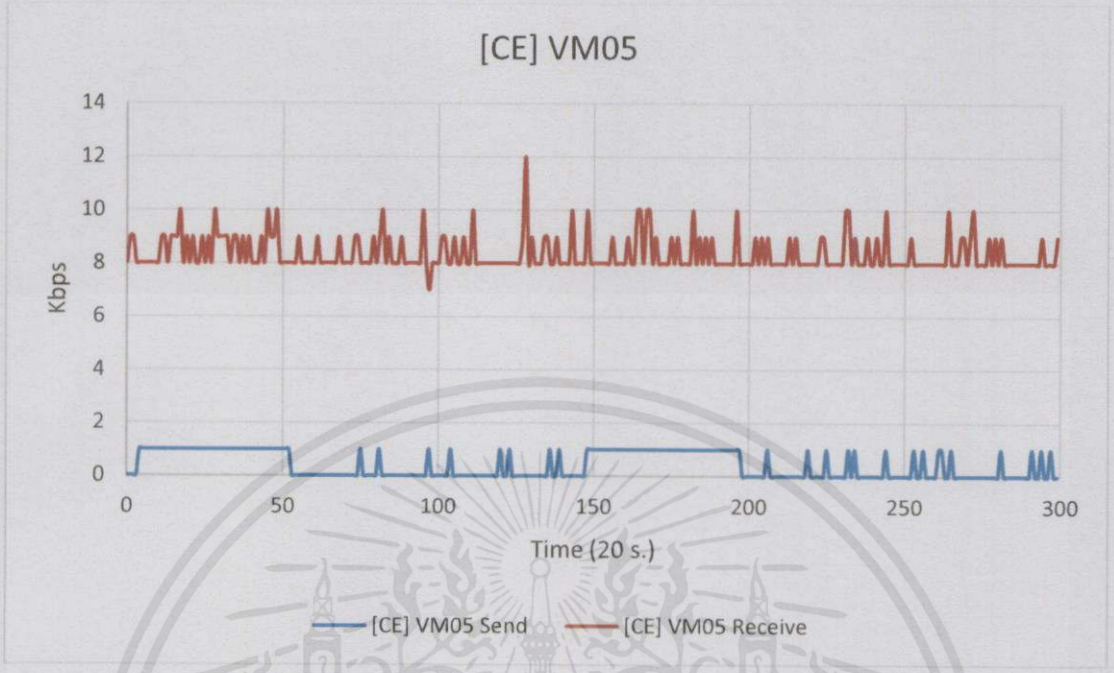
กราฟที่ 4.14 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง Database Server



กราฟที่ 4.15 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง VM03



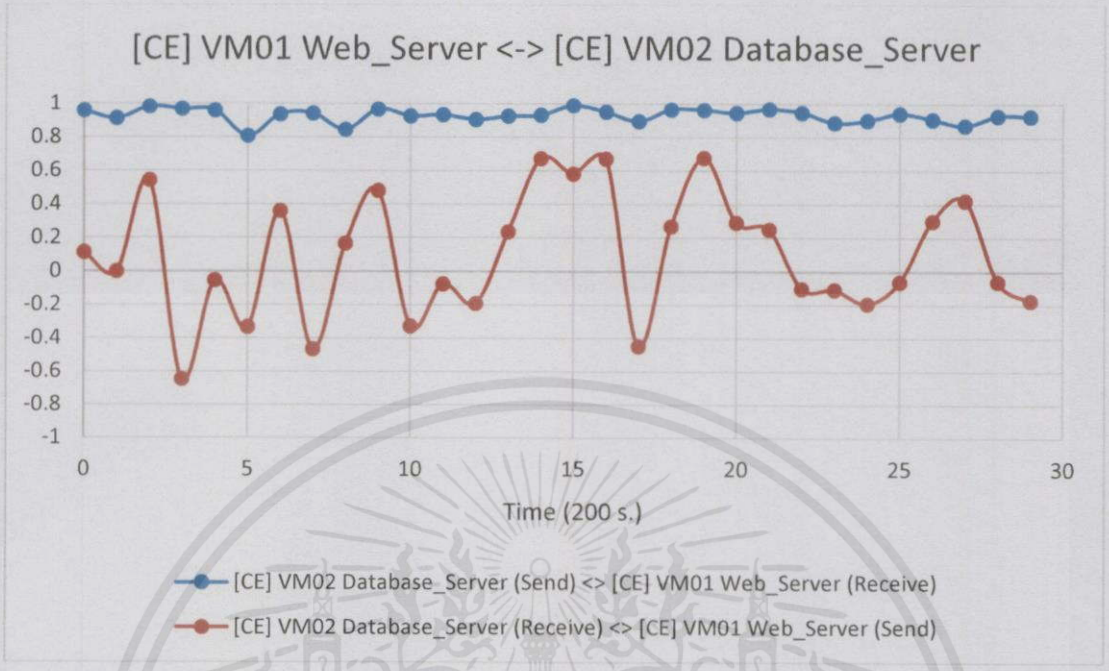
กราฟที่ 4.16 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง VM04



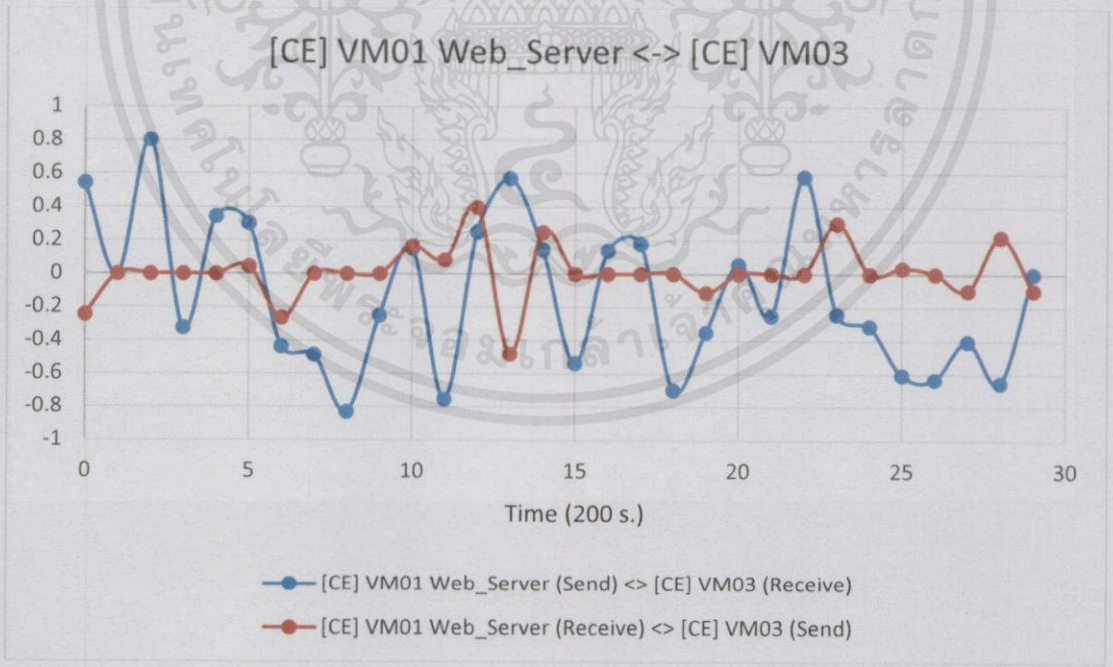
กราฟที่ 4.17 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง VM05

4.1.7.2 ผลการทดลองหาความสัมพันธ์โดยใช้ Pearson's Correlation

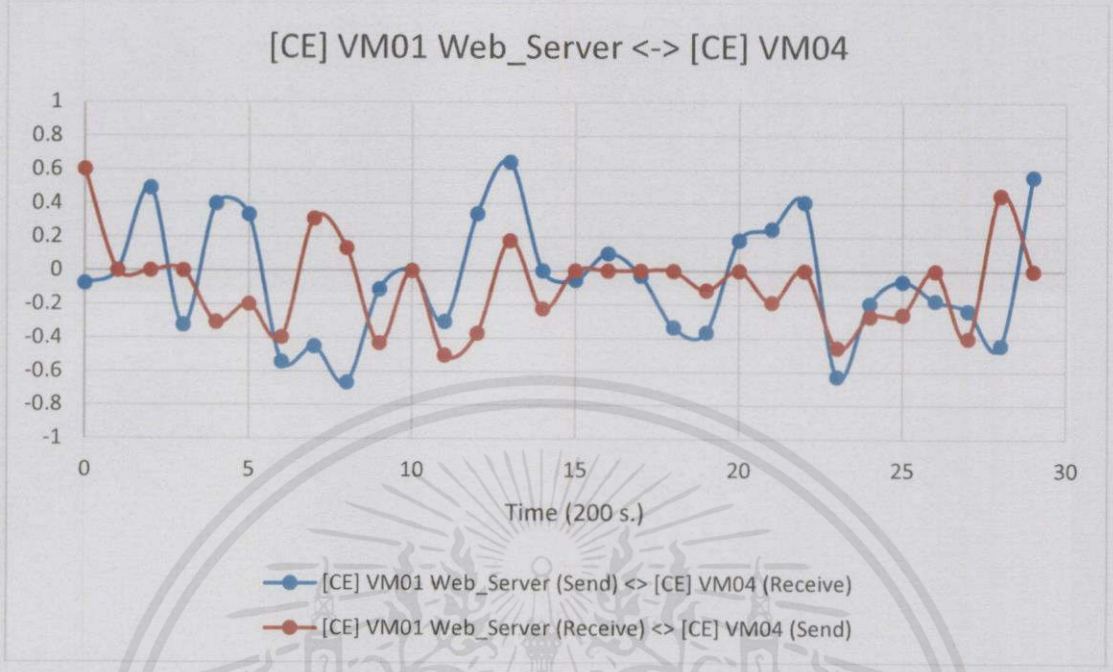
กราฟที่ 4.18, 4.19, 4.20 และ 4.21 แสดงค่า correlation ที่หาได้จากเครื่องคอมพิวเตอร์เสมือนโดยใช้อัลกอริธึม Pearson's Correlation ผลที่ได้จะคล้ายกับการทดลองแบบที่ 2 ที่มีแค่ 2 เครื่อง (VM03, VM04) ส่งหา web server นั่นคืออัลกอริธึม Pearson's Correlation สามารถหาความเกี่ยวเนื่องระหว่าง web server และ database server ได้โดยใช้ข้อมูลการส่งของ database server และข้อมูลการรับของ web server มาใช้การคิดคำนวณตามแสดงในกราฟที่ 4.18 (เส้นสีฟ้า) จะเห็นได้ว่าค่า correlation ส่วนใหญ่เข้าใกล้ 1 ส่วนความสัมพันธ์ระหว่าง Web Server และ VM03, VM04, VM05 อัลกอริธึม Pearson's Correlation ยังไม่สามารถค้นพบความสัมพันธ์ได้อย่างถูกต้องเนื่องจากปริมาณข้อมูลที่น้อย นอกจากนี้ได้วัดเวลาที่ใช้การคิดค่า correlation ของแต่ละคู่ซึ่งใช้เวลาประมาณ 100 milliseconds



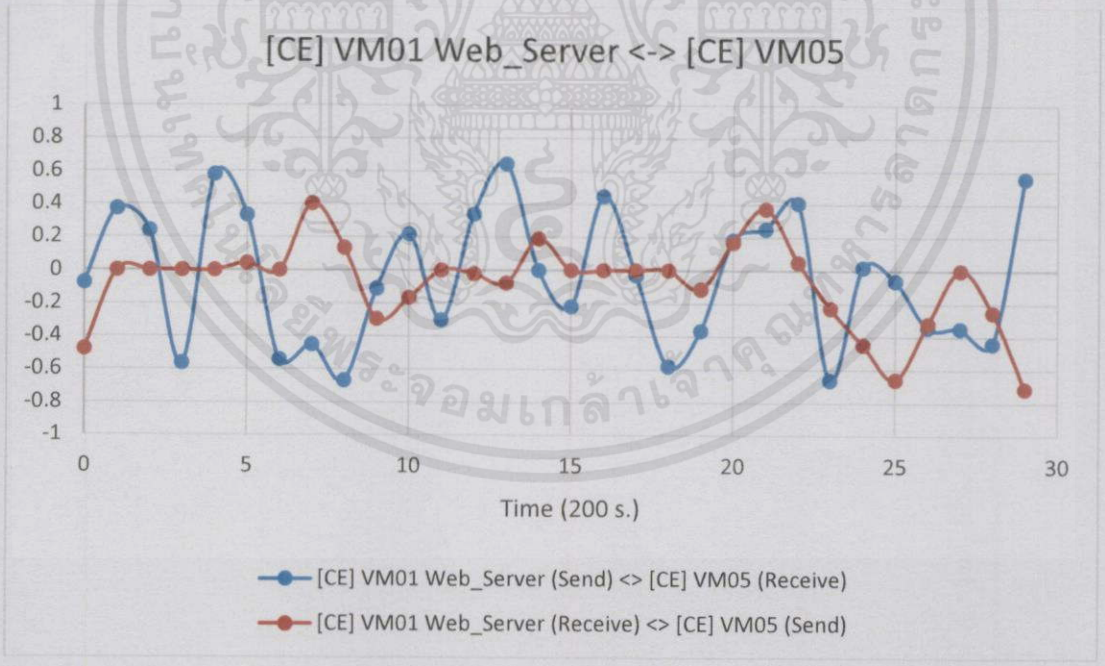
กราฟที่ 4.18 แสดงค่า Pearson's Correlation ระหว่าง Web Server และ Database Server



กราฟที่ 4.19 แสดงค่า Pearson's Correlation ระหว่าง Web Server และ VM03



กราฟที่ 4.20 แสดงค่า Pearson's Correlation ระหว่าง Web Server และ VM04



กราฟที่ 4.21 แสดงค่า Pearson's Correlation ระหว่าง Web Server และ VM05

4.1.7.3 ผลการทดลองจัดกลุ่มโดยใช้ Self-organizing map (SOM)

SOM จัดเครื่องคอมพิวเตอร์เสมือนออกเป็น 2 กลุ่มหลักๆ ดังต่อไปนี้

กลุ่มที่ 1 ประกอบด้วย

- [CE] VM01 Web Server receive
- [CE] VM02 Database Server send

โดยการจัดกลุ่มให้ข้อมูลทั้ง 2 อย่างนี้อยู่ด้วยกันทั้งหมด 30 ครั้งหรือคิดเป็น 100%จากชุดข้อมูลทั้งหมด

กลุ่มที่ 2 ประกอบด้วย

- [CE] VM03 receive
- [CE] VM04 receive
- [CE] VM05 receive
- [CE] VM02 Database Server receive

โดยการจัดกลุ่มให้ข้อมูลทั้ง 4 อย่างนี้อยู่ด้วยกันทั้งหมด 30 ครั้งหรือคิดเป็น 100% จากชุดข้อมูลทั้งหมด

ส่วนข้อมูลชุดอื่นๆ นั้นถูกจัดอยู่ในกลุ่มต่างๆ กัน ซึ่งประกอบไปด้วยชุดข้อมูลดังนี้

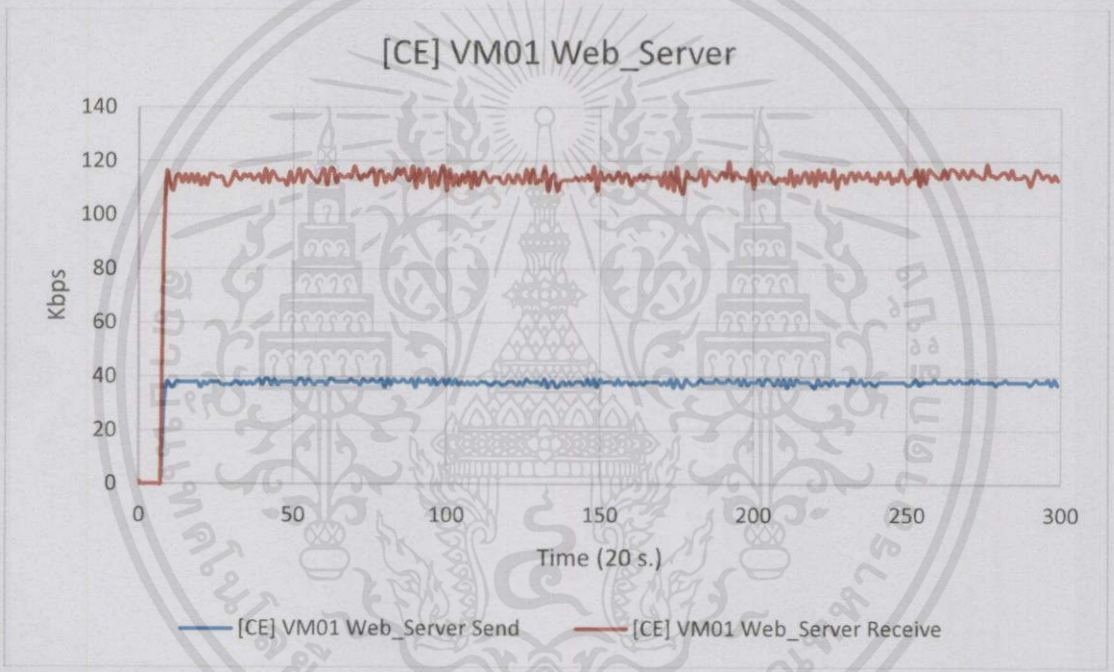
- [CE] VM01 Web Server send
- [CE] VM04 send
- [CE] VM05 send
- [CE] VM03 send

จากการทดลองนี้พบว่า SOM สามารถบอกได้ว่า web server และ database server นั้นมีความสัมพันธ์กันเนื่องจากการรับข้อมูลของเครื่อง web server และการส่งข้อมูลของ database server ถูกจัดให้อยู่ในกลุ่มเดียวกัน (กลุ่มที่ 1) อีกทั้งยังมีการจัดกลุ่มอัตราการรับข้อมูลของ VM03,VM04 และ VM05 อยู่ในกลุ่มที่ 2 ซึ่งอาจบอกได้ว่าเครื่องทั้งสองนั้นมีพฤติกรรมการรับส่งข้อมูลที่ใกล้เคียงกัน แต่จะเห็นได้ว่ามีการจัดข้อมูลอัตราการรับข้อมูลของ database server อยู่ในกลุ่มนี้ด้วยเนื่องจาก SOM นั้นจะพิจารณาข้อมูลที่มีค่าใกล้เคียงกันอยู่กลุ่มเดียวกันซึ่งในความเป็นจริงแล้ว database server นั้นไม่มีความเกี่ยวข้องโดยตรงกับเครื่อง VM03, VM04 และ VM05 ในกลุ่มที่ 2 เลย และ SOM นั้นไม่สามารถบอกได้ว่าเครื่อง VM03, VM04 และ VM05 มีการรับหรือส่งข้อมูลไปยังเครื่อง web server หรือไม่เนื่องจากไม่มีการจัดกลุ่มข้อมูลให้อยู่ในกลุ่มเดียวกัน ทั้งนี้มีสาเหตุมาจากปริมาณข้อมูลรับส่งที่น้อยเกินไป

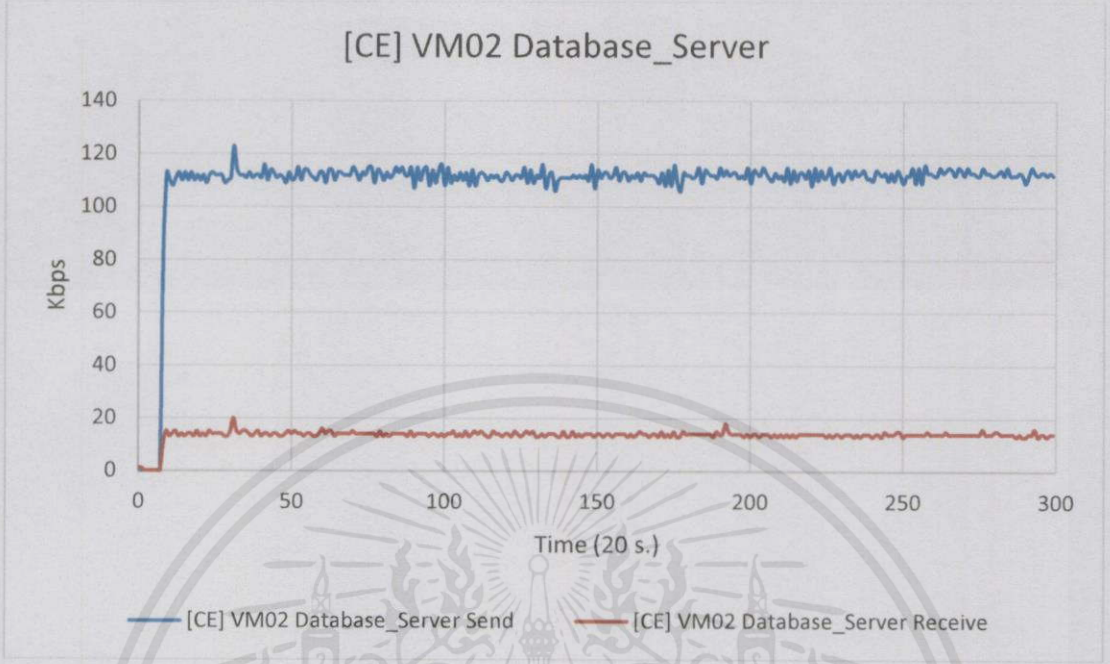
4.1.8 การทดลองแบบที่ 4

ในการทดลองนี้กำหนดให้เครื่อง VM03, VM04, VM05 และ VM06 ร้องขอข้อมูลหน้าเว็บไซท์ไปยังเครื่อง Web server โดยข้อมูลการรับส่งข้อมูลเป็นไปตามกราฟที่ 4.24, 4.25, 4.26 และ 4.27 ตามลำดับ กราฟที่ 4.22 และ 4.23 แสดงการรับส่งข้อมูลของ web server และ database server ตามลำดับ

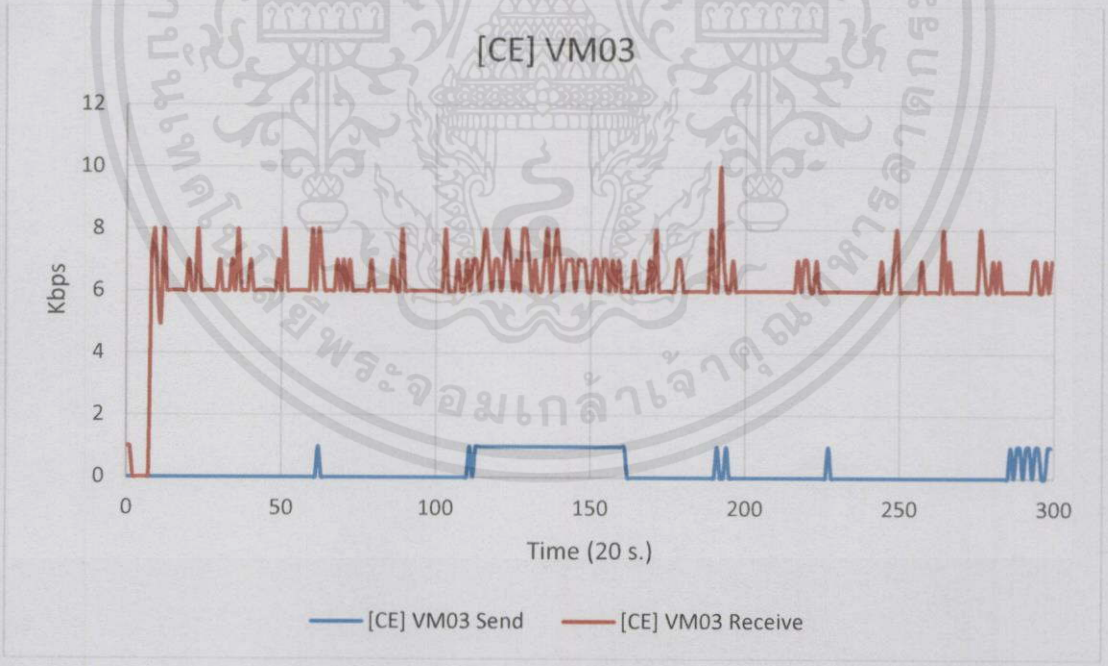
4.1.8.1 ข้อมูลการใช้งานเครือข่าย



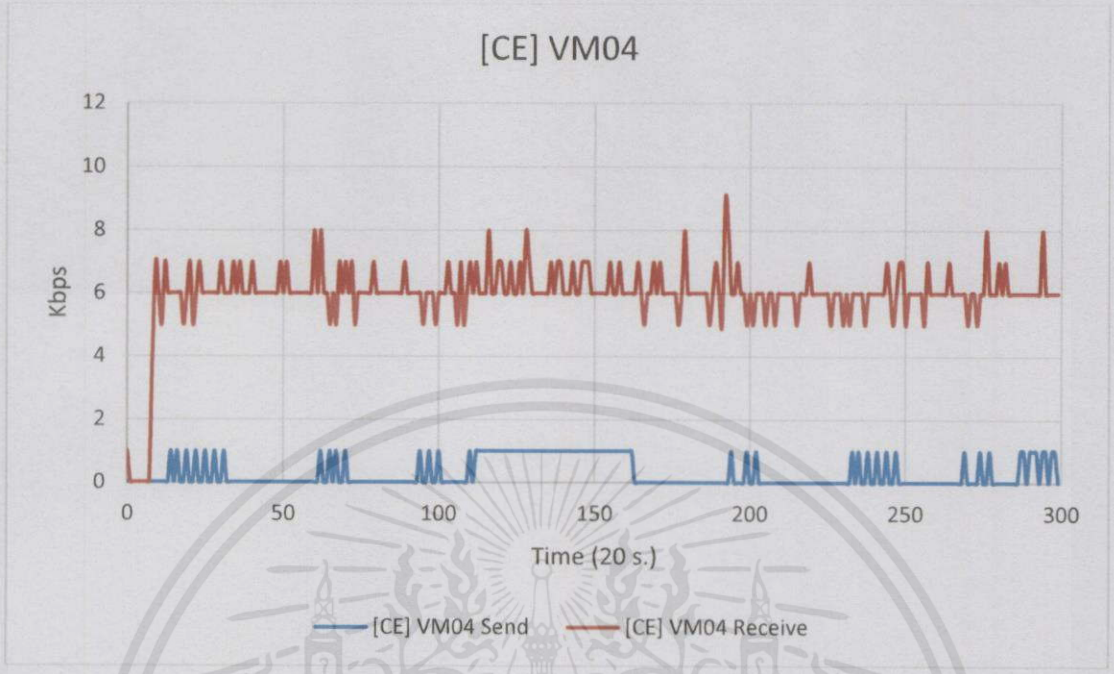
กราฟที่ 4.22 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง Web Server



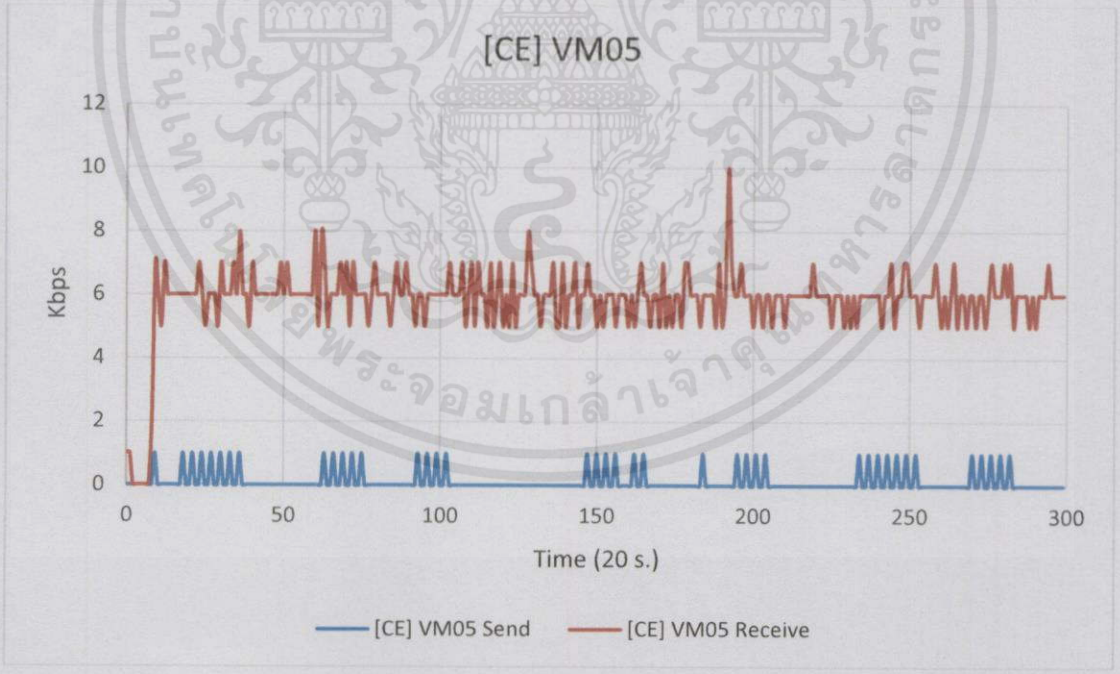
กราฟที่ 4.23 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง Database Server



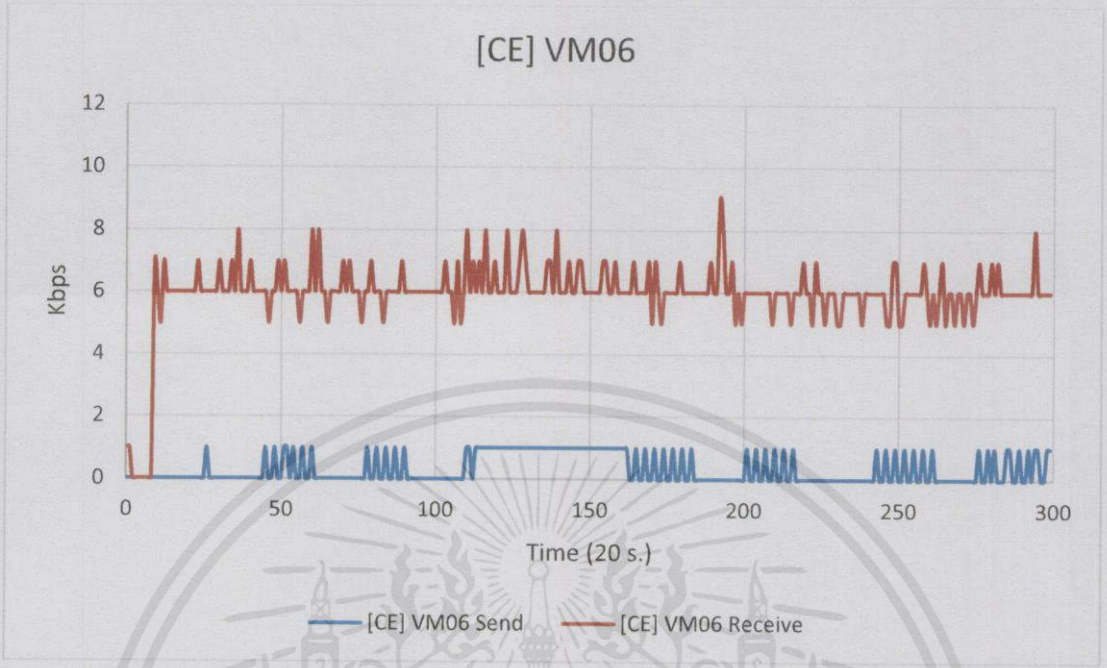
กราฟที่ 4.24 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง VM03



กราฟที่ 4.25 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง VM04



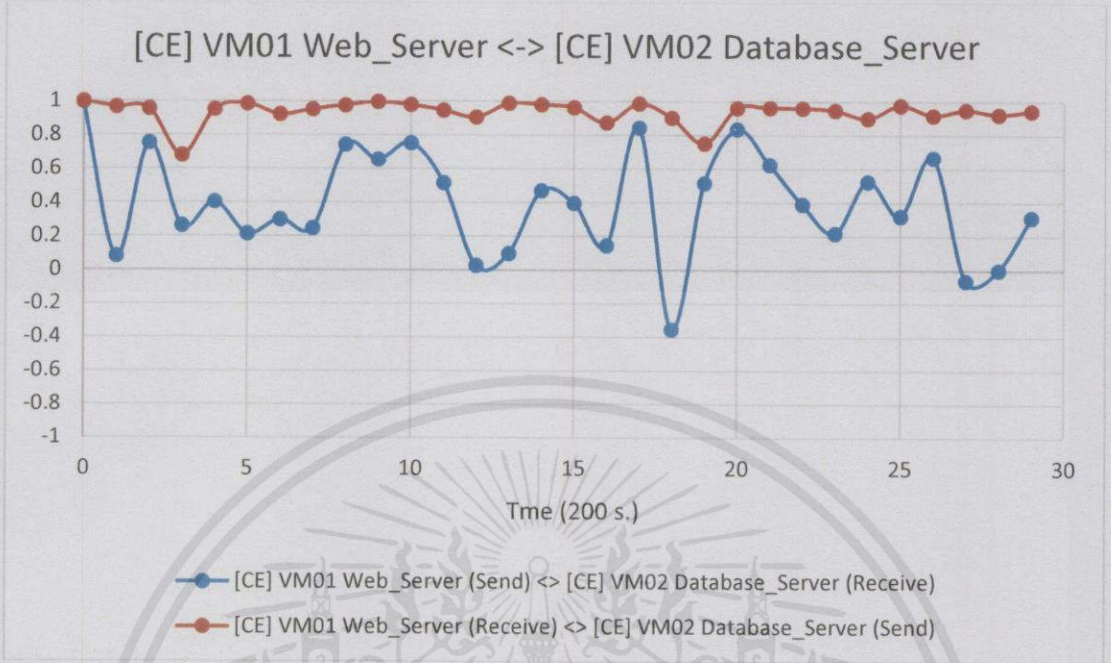
กราฟที่ 4.26 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง VM05



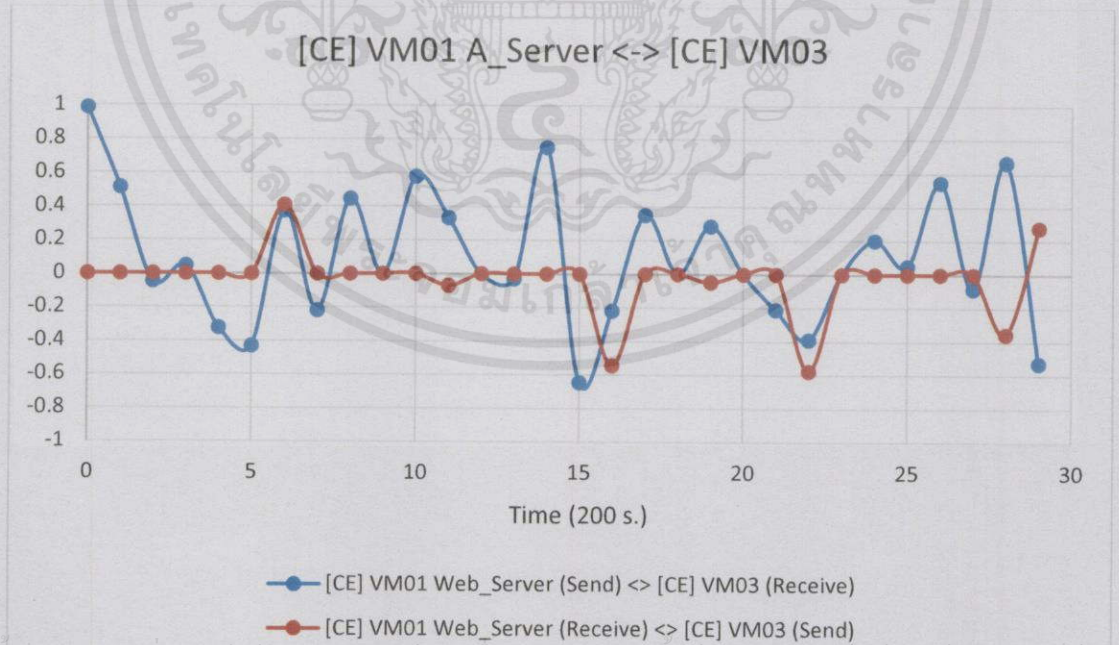
กราฟที่ 4.27 แสดงข้อมูลการใช้งานเครือข่ายของเครื่อง VM06

4.1.8.2 ผลการหาความสัมพันธ์โดยใช้ Pearson's Correlation

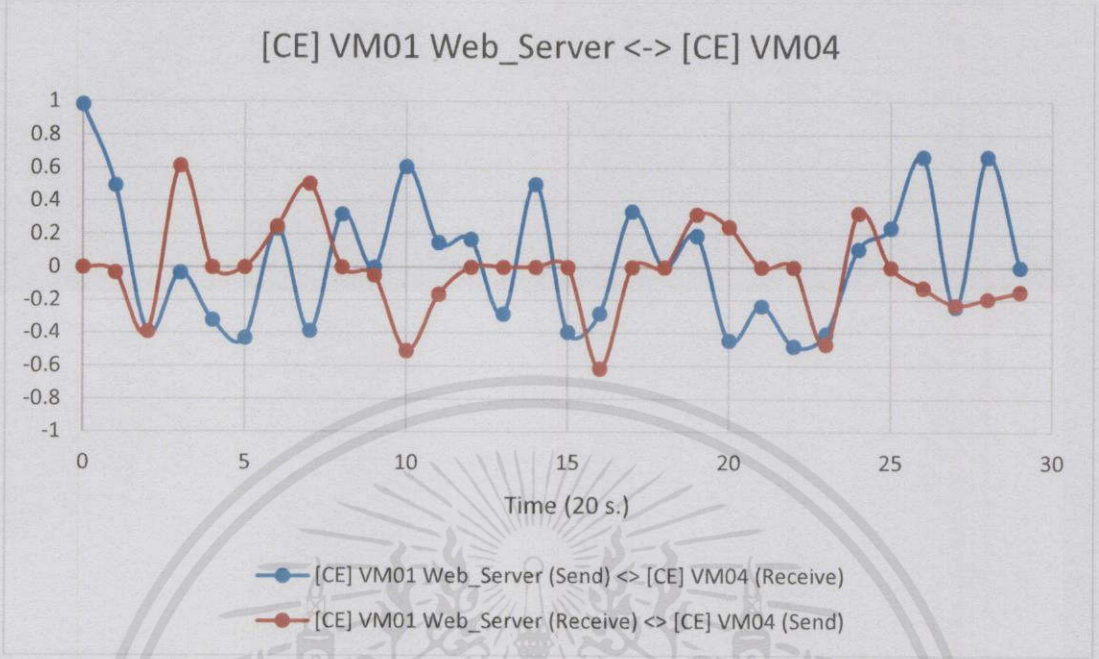
กราฟที่ 4.28, 4.29, 4.30, 4.31 และ 4.32 แสดงค่า correlation ที่หาได้จากการใช้อัลกอริธึม Pearson's Correlation ผลการทดลองที่ได้จะคล้ายกับการทดลองแบบที่ 3 ที่ใช้ 3 เครื่อง (VM03, VM04 ,VM05) ส่งหา web server นั่นคือ Pearson's Correlation สามารถหาความเกี่ยวเนื่องระหว่าง web server และ database server ได้โดยใช้ข้อมูลการส่งของ database server และข้อมูลการรับของ web server มาใช้การคิดคำนวณตาม แสดงในกราฟที่ 4.28 (เส้นสีแดง) จะเห็นได้ว่าค่า correlation ส่วนใหญ่เข้าใกล้ 1 ส่วนความสัมพันธ์ระหว่าง Web server กับ VM03, VM04, VM05, และ VM6 Pearson's Algorithm ยังไม่สามารถค้นพบได้อย่างถูกต้องเนื่องจากปริมาณข้อมูลที่น้อย



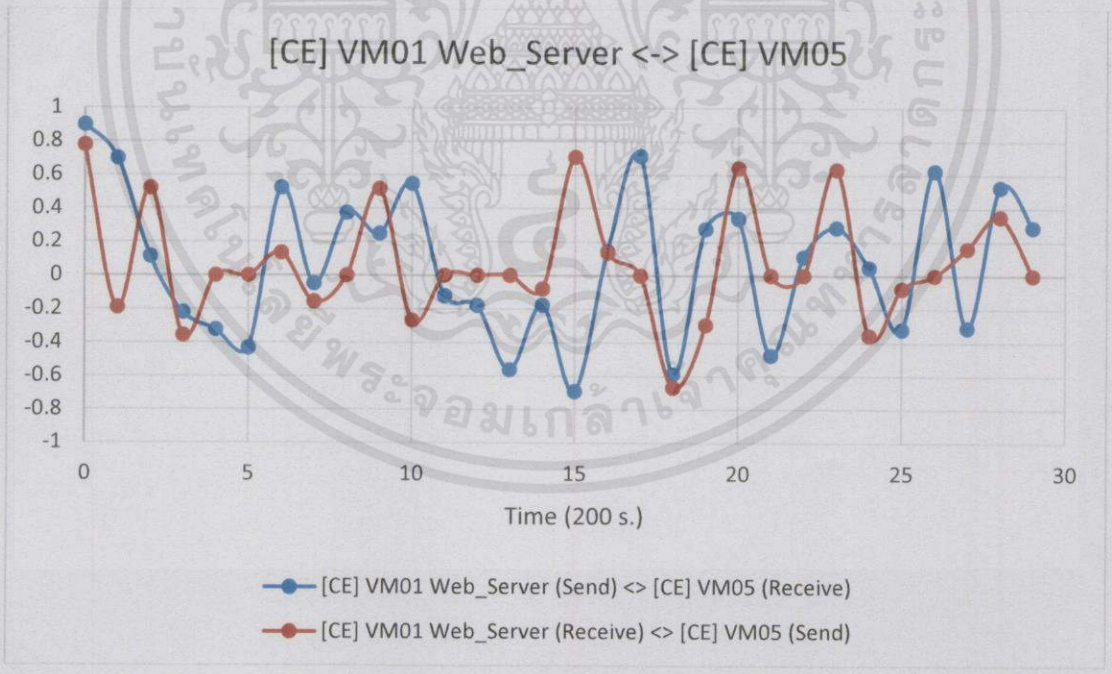
กราฟที่ 4.28 แสดงค่า Pearson's Correlation ระหว่าง Web Server และ Database Server



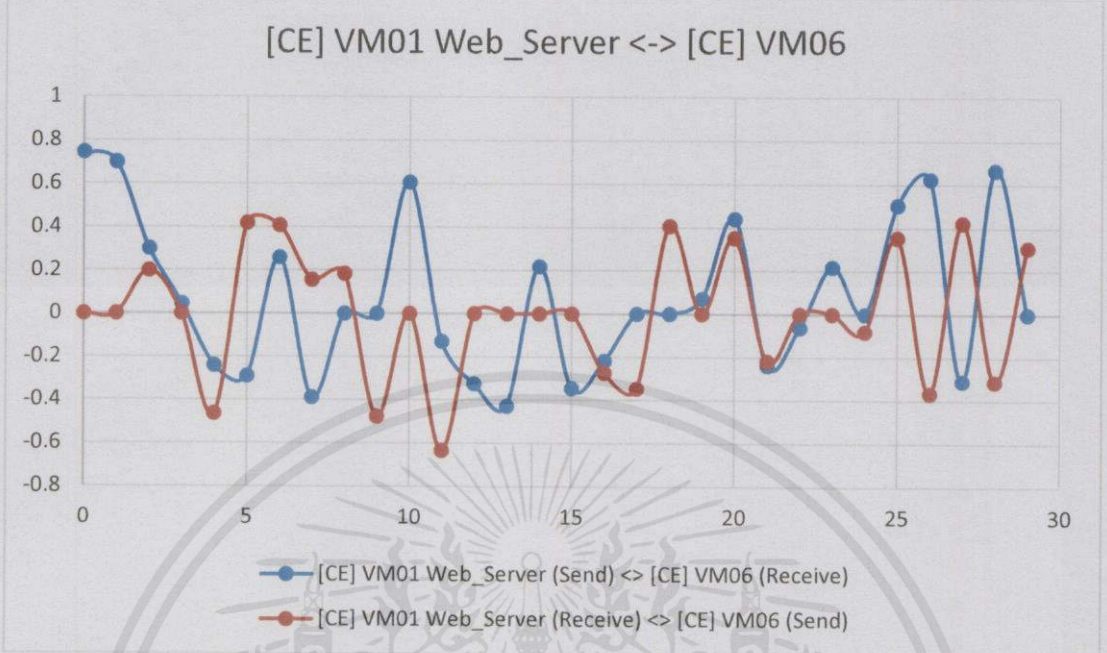
กราฟที่ 4.29 แสดงค่า Pearson's Correlation ระหว่าง Web Server และ VM03



กราฟที่ 4.30 แสดงค่า Pearson's Correlation ระหว่าง Web Server และ VM04



กราฟที่ 4.31 แสดงค่า Pearson's Correlation ระหว่าง Web Server และ VM05



กราฟที่ 4.32 แสดงค่า Pearson's Correlation ระหว่าง Web Server และ VM05

4.1.8.3 ผลการทดลองจัดกลุ่มโดยใช้ Self-organizing map (SOM)

SOM จัดกลุ่มคอมพิวเตอร์เสมือนออกเป็น 2 กลุ่มหลักๆ ดังต่อไปนี้

กลุ่มที่ 1 ประกอบด้วย

- [CE] VM01 Web Server receive
- [CE] VM02 Database Server send

โดยการจัดกลุ่มให้ข้อมูลทั้ง 2 อย่างนี้อยู่ด้วยกันทั้งหมด 30 ครั้งหรือคิดเป็น 100% จากชุดข้อมูลทั้งหมด

กลุ่มที่ 2 ประกอบด้วย

- [CE] VM03 receive
- [CE] VM04 receive
- [CE] VM05 receive
- [CE] VM06 receive

โดยการจัดกลุ่มให้ข้อมูลทั้ง 4 อย่างนี้อยู่ด้วยกันทั้งหมด 30 ครั้งหรือคิดเป็น 100% จากชุดข้อมูลทั้งหมด

ส่วนข้อมูลชุดอื่นๆ นั้นถูกจัดอยู่ในกลุ่มที่แตกต่างกันและมีข้อมูลบางชุดที่ถูกจัดอยู่ในกลุ่มเดียวกันบ้างในบางครั้ง ซึ่งประกอบไปด้วยชุดข้อมูลดังนี้

- [CE] VM01 Web Server send

- [CE] VM02 Database Server receive
- [CE] VM03 send
- [CE] VM04 send
- [CE] VM05 send
- [CE] VM06 send

จากการทดลองพบว่า SOM นั้นสามารถบอกได้ว่า web server และ database server นั้นมีความสัมพันธ์กันเนื่องจากการรับข้อมูลของเครื่อง web server และการส่งข้อมูลของ database server ถูกจัดให้อยู่ในกลุ่มเดียวกัน อีกทั้งยังมีการจัดกลุ่มอัตราการรับข้อมูลของ VM03, VM04, VM05 และ VM06 อยู่ในกลุ่มที่ 2 ซึ่งอาจบอกได้ว่าเครื่องทั้งสองนั้นมีพฤติกรรมการส่งข้อมูลที่ใกล้เคียงกัน และจะเห็นได้ว่าในการทดลองนี้ SOM ไม่ได้รวม database server ไปอยู่ในกลุ่มที่ 2 เหมือนในการทดลองแบบที่ 2 และ 3 เพราะปริมาณข้อมูลการรับของ VM03, VM04, VM05, VM06 มีการคล้ายกันมากกว่าคล้ายกับข้อมูลการรับของ database Server อย่างไรก็ดี SOM นั้นไม่สามารถบอกได้ว่าเครื่อง VM03,VM04,VM05 และ VM06 มีการรับหรือส่งข้อมูลไปยังเครื่อง web server หรือไม่ เนื่องจากไม่มีการจัดกลุ่มข้อมูลให้อยู่ในกลุ่มเดียวกัน

4.2 การตรวจสอบการทำงานของเครื่องคอมพิวเตอร์โดยใช้ Naïve Bayes

4.2.1 ข้อมูลที่ใช้ในการทดลอง

การทดลองนี้ใช้ข้อมูลเกี่ยวกับการใช้ทรัพยากรต่างๆ ของเครื่องคอมพิวเตอร์เสมือนซึ่งจะถูกแปลงข้อมูลให้อยู่ในสถานะต่างๆ ดังต่อไปนี้

1. cpu_value เป็นค่าเปอร์เซ็นต์ของการใช้งานหน่วยประมวลผล ซึ่งจะถูกแปลงเป็น 4 ระดับ คือ
    1. cpu_low คือสถานะการใช้งานในช่วง 0% - 25%
    2. cpu_normal คือสถานะการใช้งานในช่วง 26% - 50%
    3. cpu_high คือสถานะการใช้งานในช่วง 51% - 75%
    4. cpu_risk คือสถานะการใช้งานในช่วง 76% - 100%
2. mem_value เป็นค่าเปอร์เซ็นต์ของการใช้งานหน่วยความจำ ซึ่งจะถูกแปลงเป็น 4 ระดับคือ
    1. mem_low คือสถานะการใช้งานในช่วง 0% - 25%

2. mem_normal คือสถานะการใช้งานในช่วง 26% - 50%
3. mem_high คือสถานะการใช้งานในช่วง 51% - 75%
4. mem_risk คือสถานะการใช้งานในช่วง 76% - 100%

3. tx_value เป็นค่าการส่งข้อมูลเครือข่ายในหน่วยกิโลไบต์ต่อวินาที (KBps) ซึ่งนำมาคำนวณโดยใช้ logarithm scale ฐานสิบ จากนั้นแบ่งค่าที่ได้ออกเป็น 4 ระดับคือ
    1. tx_low คือสถานะการส่งข้อมูลในช่วง 0 - 1
    2. tx_normal คือสถานะการส่งข้อมูลในช่วง 1.1 - 2.5
    3. tx_high คือสถานะการส่งข้อมูลในช่วง 2.6 - 3
    4. tx_veryHigh คือสถานะการส่งข้อมูลที่มากกว่า 3

4. rx_value เป็นค่าการรับข้อมูลเครือข่ายในหน่วยกิโลไบต์ต่อวินาที (KBps) ซึ่งนำมาคำนวณโดยใช้ logarithm scale ฐานสิบ จากนั้นแบ่งค่าที่ได้ออกเป็น 4 ระดับคือ
    1. rx_low คือสถานะการรับข้อมูลในช่วง 0 - 1
    2. rx_normal คือสถานะการรับข้อมูลในช่วง 1.1 - 2.5
    3. rx_high คือสถานะการรับข้อมูลในช่วง 2.6 - 3
    4. rx_veryHigh คือสถานะการรับข้อมูลที่มากกว่า 3

5. cpu_change เป็นอัตราการเปลี่ยนแปลงของปริมาณหน่วยประมวลผลที่ใช้ในช่วงเวลา 1 นาทีโดยคิดจากปริมาณการใช้งาน ณ เวลาที่สนใจลบกับปริมาณการใช้งานเมื่อ 1 นาทีที่แล้ว (x) จากนั้นนำค่าที่คำนวณได้มาแบ่งออกเป็น 3 ระดับได้แก่
    1. cpu_idle คือ $x = 0$
    2. cpu_up คือ $x > 0$
    3. cpu_down คือ $x < 0$

6. mem_change เป็นอัตราการเปลี่ยนแปลงของปริมาณหน่วยความจำที่ใช้ในช่วงเวลา 1 นาทีโดยคิดจากปริมาณการใช้งาน ณ เวลาที่สนใจลบกับปริมาณการใช้งานเมื่อ 1 นาทีที่แล้ว (x) จากนั้นนำค่าที่คำนวณได้มาแบ่งออกเป็น 3 ระดับได้แก่
    1. mem_idle คือ $x = 0$
    2. mem_up คือ $x > 0$
    3. mem_down คือ $x < 0$

7. tx_change เป็นอัตราการเปลี่ยนแปลงของปริมาณการส่งข้อมูลในช่วงเวลา 1 นาทีโดยคิดจากปริมาณการใช้งาน ณ เวลาที่สนใจลบกับปริมาณการใช้งานเมื่อ 1 นาทีที่แล้ว (x) จากนั้นนำค่าที่คำนวณได้มาแบ่งออกเป็น 3 ระดับได้แก่
    1. tx_idle คือ x = 0
    2. tx_up คือ x > 0
    3. tx_down คือ x < 0

8. rx_change เป็นอัตราการเปลี่ยนแปลงของปริมาณการรับข้อมูลในช่วงเวลา 1 นาทีโดยคิดจากปริมาณการใช้งาน ณ เวลาที่สนใจลบกับปริมาณการใช้งานเมื่อ 1 นาทีที่แล้ว (x) จากนั้นนำค่าที่คำนวณได้มาแบ่งออกเป็น 3 ระดับได้แก่
    1. rx_idle คือ x = 0
    2. rx_up คือ x > 0
    3. rx_down คือ x < 0

ข้างล่างเป็นตัวอย่างการข้อมูลการใช้ทรัพยากรที่ถูกแปลงตามที่กล่าวมาข้างต้น

1. Cpu = 24%
2. Memory = 51%
3. Transfer Rate = 20 KBps -> log(20) = 1.3010
4. Receive Rate = 25 KBps -> log(25) = 1.3979
5. Cpu Change = +10
6. Memory Change = -5
7. Transfer Change = +10
8. Receive Change = -10

เมื่อทำการแปลงข้อมูลให้อยู่ในรูปแบบที่ต้องการแล้วจะได้เป็นชุดข้อมูลดังนี้

[cpu_low, mem_high, tx_normal, rx_normal, cpu_up, mem_down, tx_up, rx_down]

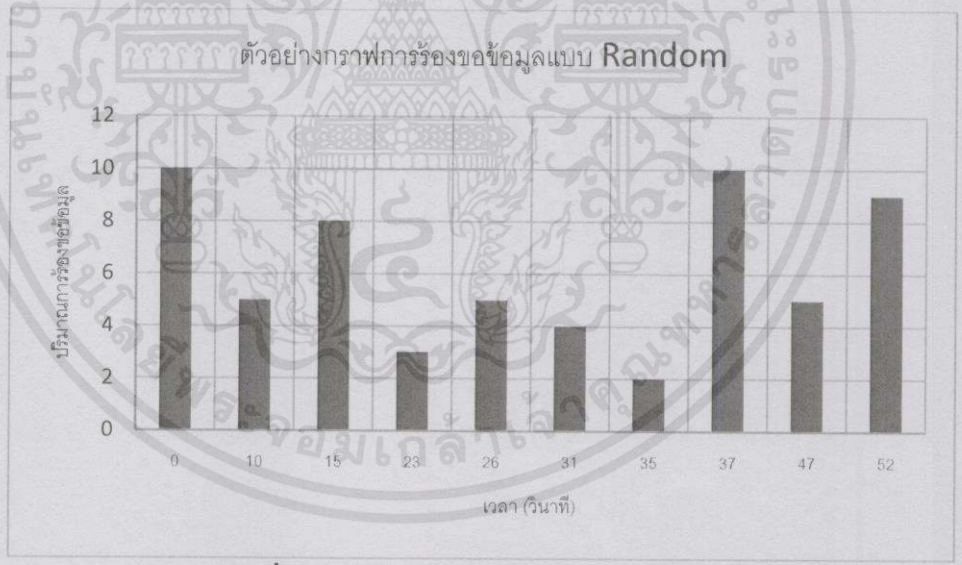
Naïve Bayes จะใช้ข้อมูลการใช้ทรัพยากรในข้างต้นในการจำแนกครื่องคอมพิวเตอร์ออกได้เป็น 2 คลาสคือ

1. Normal แสดงว่าสถานะการทำงานของเครื่องเป็นปกติ
2. Warning แสดงว่าเครื่องอาจจะมีการทำงานที่ผิดปกติ ในกรณีนี้คือถูกโจมตีแบบ DOS (SYN flood)

4.2.2 สถานการณ์ที่ใช้ในศึกษาพฤติกรรมการใช้ทรัพยากร

ในขั้นตอนการเก็บข้อมูลเพื่อนำให้ Naïve Bayes ใช้การเรียนรู้และทดสอบ ผู้พัฒนาได้เก็บข้อมูลการใช้ทรัพยากรของเครื่องคอมพิวเตอร์เสมือนในสถานการณ์ 4 ประเภทดังนี้

1. Idle เป็นช่วงเวลาที่เครื่องคอมพิวเตอร์เสมือนที่ให้บริการเว็ปไซต์และให้บริการฐานข้อมูลไม่มีการร้องขอข้อมูลจากผู้ใช้
2. Random เป็นช่วงเวลาที่ผู้ใช้จะทำส่งการร้องขอข้อมูลตามกราฟที่ 4.33 โดยแกน x คือ เวลา (วินาที) แกน y คือปริมาณการร้องขอข้อมูลพร้อมๆ กัน ณ เวลานั้นๆ ซึ่งปริมาณการร้องขอ ณ เวลาใดๆ ได้มาจากการสุ่มค่า และเวลาที่ใช้ในการส่งการร้องขอข้อมูลก็มาจากการสุ่มค่าเช่นกัน



กราฟที่ 4.33 แสดงการร้องขอข้อมูลแบบ random

จากกราฟที่ 4.33 ผู้ใช้จะส่งการร้องขอข้อมูลเว็บไซต์ 10 ครั้งพร้อมๆ กัน จากนั้น จากนั้นผู้ใช้จะไม่ส่งการร้องขอข้อมูลไป 10 วินาที เมื่อเวลาผ่านไป 10 วินาทีก็จะเริ่มทำการสุ่มค่าใหม่ขึ้นมาซึ่งเป็น 5 นั่นหมายถึงผู้ใช้จะส่งการร้องขอข้อมูล 5 ครั้งพร้อมๆ กัน จากนั้นก็จะทำการสุ่มตัวเลขถัดไปซึ่งเป็นเลข 5 ซึ่งแปลว่าที่เวลา 15 วินาที

ผู้ใช้ก็จะทำการร้องขอข้อมูลพร้อมๆ กันอีกเป็นจำนวนที่ได้จากการสุ่มค่าครั้งต่อไป ซึ่งทำแบบนี้ไปเรื่อยๆ

3. Heavy Load คือผู้ใช้ได้ทำการร้องขอข้อมูลหน้าเว็ปไซต์จำนวนมากจนกระทั้งเครื่องคอมพิวเตอร์เสมือนที่ให้บริการเว็ปเซิร์ฟเวอร์มีการใช้งานหน่วยความจำและหน่วยประมลผลมากกว่า 80% ของทั้งหมด
4. DOS Attack (SYN flood) คือสถานการณ์ที่ครื่องคอมพิวเตอร์เสมือนที่ให้บริการเซิร์ฟเวอร์ถูกโจมตีแบบ DOS นั่นคือมีการร้องขอข้อมูลจำนวนมากโดยไม่รอเซิร์ฟเวอร์ตอบกลับทำให้เกิดการเชื่อมต่อกับจำนวนมากจนกระทั่งเครื่องที่ให้บริการเว็บไซต์ไม่สามารถให้บริการต่อไปได้

4.2.3 Data Training & Testing

เมื่อทำการเก็บข้อมูลตามที่กล่าวใน 4.2.2 แล้วนำข้อมูลทั้งหมดมาแบ่งเป็นชุดข้อมูลสำหรับการสอน (Training Set) และชุดข้อมูลสำหรับตรวจสอบ (Test Set) โดยใช้โปรแกรม Weka เข้ามาช่วยโดย 60 เปอร์เซ็นต์ของข้อมูลทั้งหมดจะเป็นชุดข้อมูลสำหรับการสอนและอีก 40 เปอร์เซ็นต์เป็นชุดข้อมูลสำหรับตรวจสอบ

ตารางที่ 4.1 แสดงลักษณะข้อมูลของชุดข้อมูลที่ใช้สอนและตรวจสอบ

ลักษณะข้อมูล	จำนวนทั้งหมด	จำนวนที่ใช้สอน	จำนวนที่ใช้ตรวจสอบ
Idle	14	10	4
Random	39	22	17
Heavy Load	42	29	13
DOS Attack	57	31	26
Total	152	92	60

4.2.3.1 ผลการทดลองของ Naïve Bayes

จากการทดลองการจำแนกของ Naïve Bayes ผู้พัฒนาได้นำผลที่ได้มาคำนวณหา 2 ค่า คือ Accuracy และ Precision

$$\text{Accuracy} = \frac{\text{จำนวน True positives} + \text{จำนวน True negative}}{\text{จำนวน True positives} + \text{จำนวน False positives} + \text{จำนวน False negatives} + \text{จำนวน True negatives}}$$

$$\text{Precision} = \frac{\text{จำนวน True positives}}{\text{จำนวน True positives} + \text{จำนวน False positives}}$$

หลังจากนำชุดข้อมูลการสอน (Training Set) ที่ได้แบ่งโดยโปรแกรม Weka ไปสอนให้กับ Naïve Bayes จากนั้นนำชุดข้อมูลตรวจสอบ (Test Set) มาทำการจำแนกกลุ่ม ได้ผลลัพธ์ดังตารางที่ 4.2

ตารางที่ 4.2 แสดงผลการจำแนกกลุ่มของชุดข้อมูลตรวจสอบ

True positive = 32	False positive = 3
False negative = 1	True negative = 25

Accuracy = ((32 + 25) / (32 + 3 + 1 + 25)) * 100% = 93.44%

Precision = (32 / (32 + 3)) * 100% = 91.43%

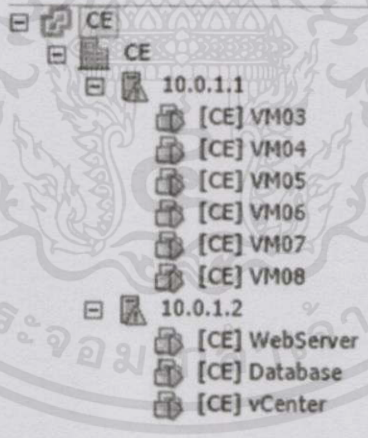
จากการทดลองมี 3 กรณีเป็น False positive ซึ่งหมายถึงเครื่องคอมพิวเตอร์มีรูปแบบการใช้ทรัพยากรคล้ายกับการถูกโจมตีแบบ DOS แต่จริงๆ แล้วไม่ได้ถูกโจมตี

1. [cpu_low, mem_high, tx_low, rx_normal, cpu_down, mem_up, tx_down, rx_down] – ในกรณีเครื่องคอมพิวเตอร์อยู่ในสถานะ Heavy Load
2. [cpu_high, mem_high, tx_low, rx_normal, cpu_down, mem_up, tx_down, rx_down] – ในกรณีเครื่องคอมพิวเตอร์อยู่ในสถานะ Heavy Load
3. [cpu_risk, mem_high, tx_normal, rx_normal, cpu_up, mem_up, tx_up, rx_up] – ในกรณีเครื่องคอมพิวเตอร์อยู่ในสถานะ Heavy Load

จากการทดลองมีอยู่ 1 กรณีที่เป็น False negative นั่นคือเวลาที่เครื่องคอมพิวเตอร์เสมือนกำลังถูกการโจมตีแบบ DOS อยู่ ซึ่งจะมีรูปแบบการใช้งานทรัพยากรดังต่อไปนี้ [cpu_normal, mem_high, tx_low, rx_normal, cpu_up, mem_idle, tx_idle, rx_idle]

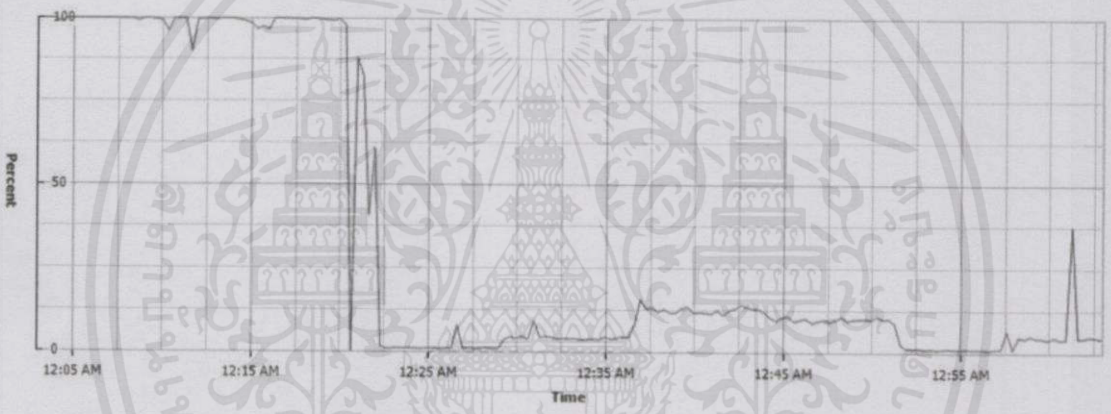
4.2.4 Naïve Bayes Evaluation

ผู้พัฒนาได้นำ Naïve Bayes ที่ผ่านการสอนและตรวจสอบในขั้นต้นมาใช้จำแนกลักษณะการทำงานของเครื่องคอมพิวเตอร์เสมือนที่กำลังถูกโจมแบบ DOS ซึ่งผู้พัฒนาได้สร้างระบบจำลองที่ประกอบด้วยเครื่องคอมพิวเตอร์แม่ข่ายจำนวน 2 เครื่องได้แก่เครื่องที่มี IP 10.0.1.1 และ 10.0.1.2 ทั้งสองเครื่องนี้มี Intel® Xeon™ CPU 3.00 GHz, Memory 12 GB, Storage 197.50 GB และผู้พัฒนาได้สร้างเครื่องคอมพิวเตอร์เสมือนจำนวน 9 เครื่องบนเครื่องแม่ข่ายตามรูปที่ 4.1 จากนั้นได้ใช้โปรแกรม Low Orbit Ion Cannon ที่ติดตั้งอยู่บนภายนอกเพื่อทำการโจมตีแบบ DOS (SYN Flood) ไปที่เครื่องคอมพิวเตอร์เสมือน [CE] WebServer โดยเครื่องคอมพิวเตอร์เสมือนเครื่องนี้มี virtual CPU 1 Core, Memory 1024 MB และ Storage 21.12 GB

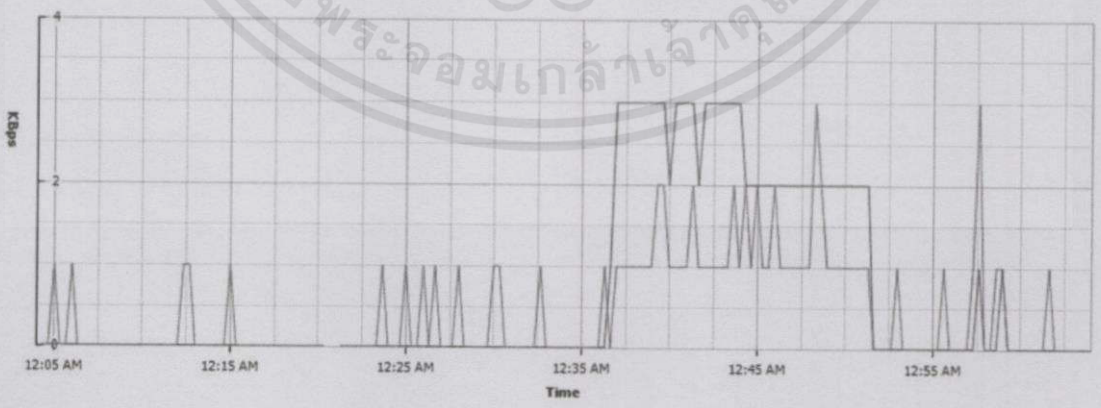


รูปที่ 4.2 แสดงโครงสร้างภายในศูนย์ข้อมูล

กราฟที่ 4.34 แสดงปริมาณการใช้หน่วยความจำของเครื่อง [CE] WebServer ก่อนทำการโจมตี



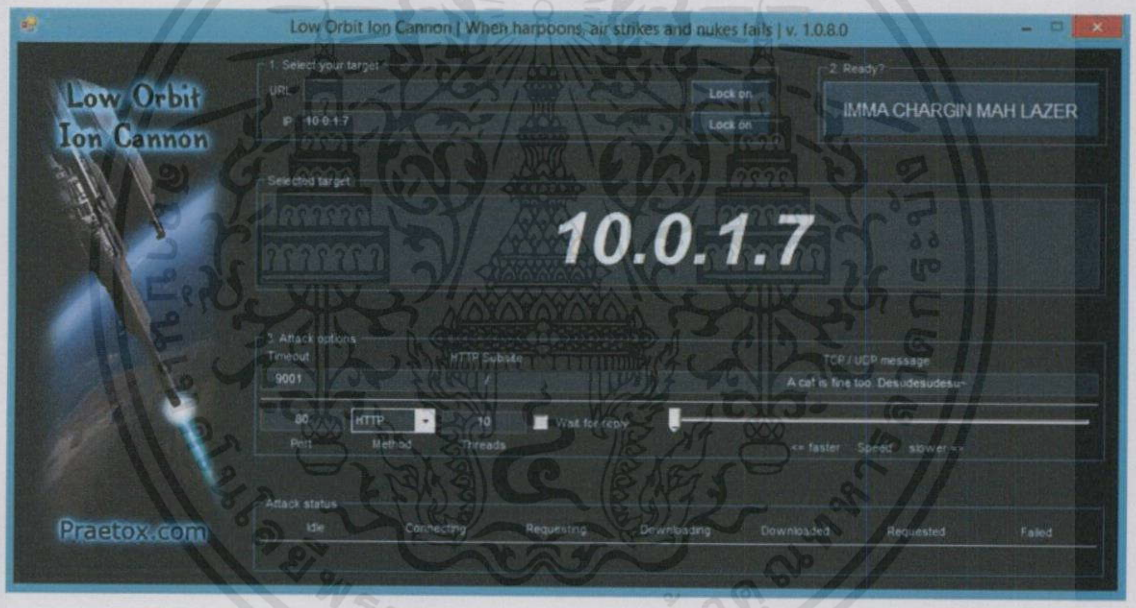
กราฟที่ 4.35 แสดงปริมาณการใช้งานหน่วยประมวลผลของเครื่อง [CE] WebServer ก่อนทำการโจมตี



กราฟที่ 4.36 แสดงปริมาณการรับ-ส่งข้อมูลผ่านเครือข่ายของเครื่อง [CE] WebServer ก่อนทำการโจมตี

กราฟที่ 4.34 ,4.35 และ 4.36 แสดงการใช้งานทรัพยากรของเครื่อง x ก่อนที่จะถูกโจมตี ซึ่งจะเห็นได้ว่าปริมาณการใช้งานหน่วยความจำและหน่วยประมวลผลส่วนใหญ่อยู่ในช่วง 4%-10% และปริมาณการรับ-ส่งข้อมูลผ่านเครือข่ายอยู่ในช่วง 0-4 กิโลไบต์ต่อวินาที (KBps) เมื่อนำ Naïve Bayes มาตรวจสอบ ก็ไม่พบว่าเครื่องมีการทำงานที่ผิดปกติ

จากนั้นทำการทดลองโจมตีแบบ DOS ด้วยโปรแกรม Low Orbit Ion Cannon ซึ่งเป็นโปรแกรมการโจมตีทางเครือข่ายที่ได้รับความนิยมสูงซึ่งได้มีการตั้งค่าโปรแกรมดังรูปที่ 4.3 ซึ่งทำการตั้งค่าการโจมตีแบบ HTTP ไปที่พอร์ท 80 โดยทำการเชื่อมต่อเพื่อร้องขอข้อมูลพร้อมๆ กันจำนวน 10 การเชื่อมต่อโดยไม่รอการตอบกลับจากเครื่องเซิร์ฟเวอร์ไปยังเครื่องคอมพิวเตอร์เสมือน [CE] WebServer



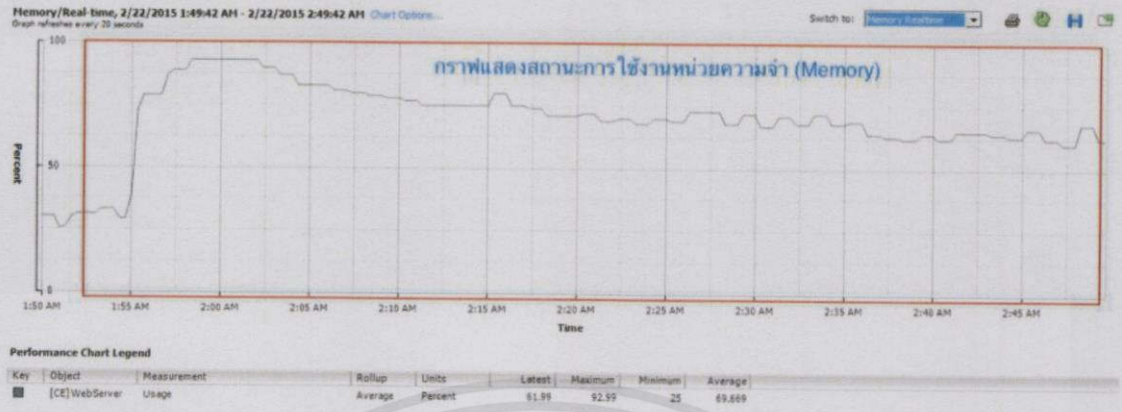
รูปที่ 4.3 แสดงการตั้งค่าโปรแกรม Low Orbit Ion Cannon

หลังจากทำการโจมตีไป 90 วินาที โปรแกรม Low Orbit Ion Cannon ได้ส่งการร้องขอข้อมูล 3135 ครั้ง ด้วย 10 การเชื่อมต่อพร้อมๆ กันและมีจำนวนการร้องขอข้อมูลที่ไม่ประสบความสำเร็จ 445 ครั้งตามรูปที่ 4.4.

Attack status

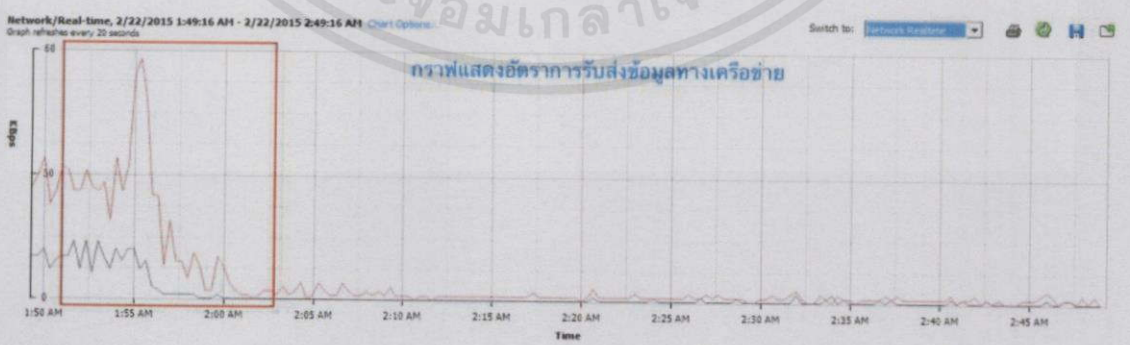
Idle	Connecting	Requesting	Downloading	Downloaded	Requested	Failed
0	10	0	0	3135	3135	445

รูปที่ 4.4 แสดงจำนวนการร้องขอทั้งหมด



กราฟที่ 4.37 ปริมาณการช้งานหน่วยความจำภายของเครื่อง [CE] WebServer หลังการถูกโจมตีทางเครือข่าย

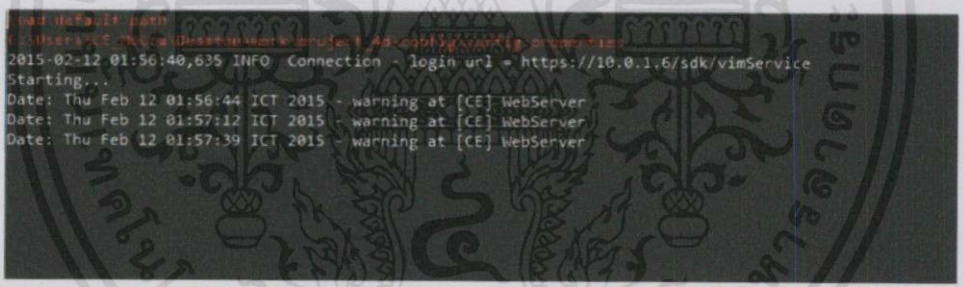
กราฟที่ 4.38 ปริมาณการใช้งานหน่วยประมวลผลของเครื่อง [CE] WebServer ภายหลังการถูกโจมตีทางเครือข่าย



กราฟที่ 4.39 ปริมาณการรับ-ส่งข้อมูลทางเครือข่ายของเครื่อง [CE] WebServer ภายหลังถูกโจมตีทางเครือข่าย

เมื่อมีสถานะการทำงานที่ผิดปกติเกิดขึ้นจากการโจมตีทางเครือข่ายด้วย DOS Attack (Syn flood) โดยมีลักษณะการใช้ทรัพยากรเป็นไปตามกราฟที่ 4.37 ซึ่งเป็นกราฟแสดงถึงการใช้งานหน่วยความจำของเครื่องคอมพิวเตอร์เสมือน [CE] WebServer ซึ่งการใช้งานหน่วยความจำจะสูงขึ้น เนื่องจากเครื่องคอมพิวเตอร์เสมือนต้องทำการจองหน่วยความจำไว้สำหรับการเปิดรับการเชื่อมต่อ แต่หลังจากเปิดรับการเชื่อมต่อจนหน่วยความจำเต็มทำให้ไม่สามารถที่จะรับการเชื่อมต่ออื่นๆ ได้อีก, กราฟที่ 4.38 และ 4.39 ซึ่งเป็นกราฟแสดงการใช้งานหน่วยประมวลผลและอัตราการรับส่งข้อมูลผ่านเครือข่ายของเครื่องคอมพิวเตอร์เสมือน [CE] WebServer ซึ่งการใช้งานหน่วยประมวลผลและอัตราการรับส่งข้อมูลมีลักษณะที่คล้ายคลึงกันคือในช่วงแรกจะมีการใช้งานที่สูงมากเนื่องจากเครื่องคอมพิวเตอร์เสมือนต้องการที่จะรับการเชื่อมต่อ ประมวลผล และตอบกลับโดยเร็วที่สุดเท่าที่จะเป็นไปได้ แต่เมื่อหน่วยความจำเต็มทำให้ไม่สามารถรับการเชื่อมต่อได้จึงทำให้ไม่สามารถที่จะประมวลผลและตอบกลับการร้องขอได้ทำให้การใช้งานหน่วยความจำและอัตราการรับส่งข้อมูลลดลง

โดยเมื่อนำข้อมูลทรัพยากรที่ได้ไปให้ Naïve Bayes จำแนกกลุ่ม Naïve Bayes สามารถตรวจจับพฤติกรรมการโจมตีได้ภายใน 40-80 วินาทีหลังที่ได้ทำการโจมตีดังรูปที่ 4.5



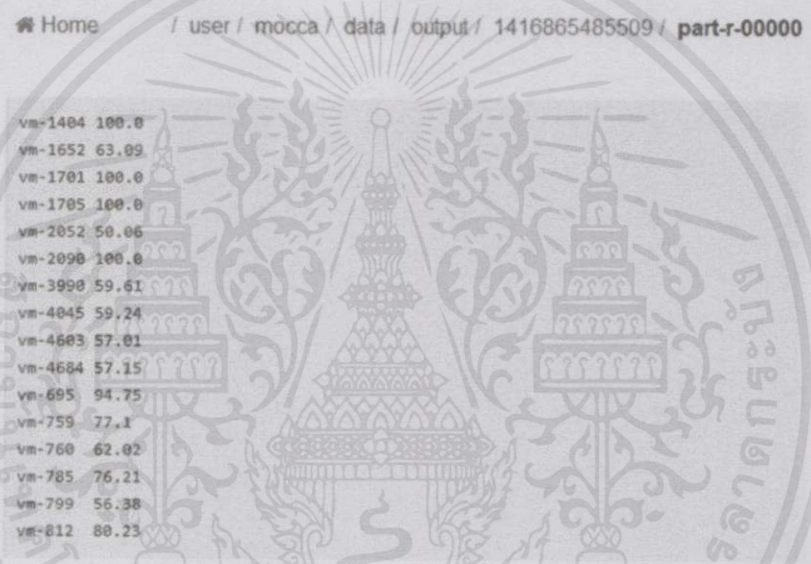
รูปที่ 4.5 ผลการจำแนกเครื่อง [CE] WebServer โดยใช้ Naïve Bayes หลังจากเครื่องถูกโจมตี

4.2.5 สรุปผลการทดลอง

อัลกอริทึม Naïve Bayes สามารถนำมาตรวจสอบการโจมตีทางเครือข่ายได้ อย่างไรก็ดีความแม่นยำในการตรวจสอบขึ้นอยู่กับชุดข้อมูลที่นำมาใช้สอน ถ้าชุดข้อมูลมีลักษณะการใช้ทรัพยากรใกล้เคียงกับเครื่องที่ต้องการจำแนกกลุ่มก็จะทำให้ความแม่นยำในการจำแนกมากขึ้น ซึ่งการใช้งานทรัพยากรก็ขึ้นอยู่กับโปรแกรมและปริมาณงานที่เครื่องคอมพิวเตอร์ได้รับ

4.3 ส่วนการตรวจสอบสถานะของระบบโดยใช้ Hadoop

ผู้พัฒนาได้นำ MapReduce Framework ใน Apache Hadoop มาใช้ค้นหาเครื่องคอมพิวเตอร์เสมือนที่มีปริมาณการใช้งานทรัพยากรที่สนใจ ตัวอย่างเช่น หาเครื่องคอมพิวเตอร์เสมือนที่ใช้ปริมาณหน่วยประมวลผลเกิน 50 เปอร์เซ็นต์ในช่วงเวลา 2 ชั่วโมงที่ผ่านมาจากเครื่องคอมพิวเตอร์เสมือนทั้งหมด 817 เครื่อง จากการทดลองพบว่าโปรแกรมสามารถหาเครื่องคอมพิวเตอร์เสมือนได้อย่างถูกต้องตามรูปที่ 4.6 ภายในเวลา 2-5 วินาที



รูปที่ 4.6 Apache Hadoop แสดงเครื่องคอมพิวเตอร์เสมือนที่ใช้ปริมาณหน่วยประมวลผลเกิน 50 เปอร์เซ็นต์ในเวลา 2 ชั่วโมงที่ผ่านมา

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

จากการผลศึกษาในโครงงานนี้ ผู้ดูแลระบบคลาวด์สามารถนำ Naive Bayes มาประยุกต์ใช้จำแนกเครื่องคอมพิวเตอร์เสมือนที่อาจจะมีการทำงานผิดปกติ เช่น ถูกโจมตีแบบ DOS ซึ่งความแม่นยำในการจำแนกขึ้นอยู่กับชุดข้อมูลที่ใช้การสอน หรือใช้ Apache Hadoop มาช่วยในการหาเครื่องที่อาจจะมีการใช้งานทรัพยากรที่สูดผิดปกติ หลังจากนั้นผู้ดูแลระบบสามารถระบุเครื่องคอมพิวเตอร์เสมือนอื่นๆ ที่อาจจะได้รับผลกระทบเนื่องจากเครื่องที่ทำงานผิดปกติ โดยดูจากความสัมพันธ์ระหว่างเครื่องคอมพิวเตอร์เสมือนที่คำนวณได้จาก Pearson's Correlation Algorithm และ Self-Organizing Map ซึ่งทั้งสองอัลกอริธึมนี้ใช้แค่ข้อมูลการใช้งานทรัพยากรซึ่งสังเกตได้จากภายนอกมาวิเคราะห์ ทำให้ผู้ดูแลระบบไม่ต้องเข้าถึงเครื่องคอมพิวเตอร์เสมือนซึ่งเป็นของลูกค้า และเนื่องจากข้อมูลในศูนย์ข้อมูลมีเป็นจำนวนมาก ผู้พัฒนาจึงได้นำอัลกอรีธึมมาประยุกต์ใช้กับการประมวลผลแบบกระจายซึ่งใช้ Apache Spark เพื่อให้การประมวลผลข้อมูลเป็นได้อย่างรวดเร็วขึ้น

5.2 ปัญหาอุปสรรคและแนวทางการแก้ไขปัญหา

1. ในการวิเคราะห์หาความสัมพันธ์ของข้อมูลในปัจจุบันใช้เพียงค่าการใช้งานเครือข่ายซึ่งอาจทำให้ไม่สามารถหาความสัมพันธ์ได้ทั้งหมด ซึ่งในอนาคตสามารถใช้การใช้งานหน่วยประมวลผลหรือหน่วยความจำเข้ามาช่วยในการหาความสัมพันธ์เพื่อให้เกิดความแม่นยำยิ่งขึ้น
2. อัลกอริธึม Pearson's Correlation นั้นใช้ค่าเฉลี่ยของชุดข้อมูลเพื่อนำมาเปรียบเทียบความแตกต่างซึ่งถ้าข้อมูลมีค่าเฉลี่ยน้อยจะทำให้ค่าสหสัมพันธ์มีการเปลี่ยนแปลงได้ง่าย ในการนำไปใช้งานจึงควรมีการตรวจสอบชุดข้อมูลในเบื้องต้นว่าข้อมูลเหล่านั้นอาจมีความสัมพันธ์กันหรือไม่ เช่น การตรวจสอบเฉพาะเครื่องคอมพิวเตอร์แบบเสมือนที่อยู่ใน Network เดียวกันก่อน เนื่องจากเครื่องคอมพิวเตอร์แบบเสมือนที่อยู่ในกลุ่มเดียวกันนั้นมีความเป็นไปได้ที่จะมีความสัมพันธ์ซึ่งกันซึ่งจะทำให้ค่าสหสัมพันธ์มีความถูกต้องยิ่งขึ้น
3. การจัดกลุ่มข้อมูลโดยใช้อัลกอริธึม Self-Organizing Map (SOM) นั้นจะมีความแม่นยำน้อยลงเวลาที่ปริมาณข้อมูลที่นำมาใช้จำแนกมีความแตกต่างกันมาก เช่น เวลาเครื่องแม่ข่ายได้รับการร้องขอข้อมูลจากเครื่องลูกข่ายจำนวนหลายเครื่องจะทำให้ SOM ไม่สามารถจัดกลุ่มให้เครื่องแม่ข่ายและเครื่องลูกข่ายอยู่ในกลุ่มเดียวกันได้ ในการนำไปใช้งานจึงควรมีการ

4. ตรวจสอบชุดข้อมูลในเบื้องต้นว่าข้อมูลเหล่านั้นอาจมีความเป็นไปได้ที่จะมีความเกี่ยวเนื่องหรืออาจอยู่ในกลุ่มเดียวกัน เช่น การจัดกลุ่มเครื่องคอมพิวเตอร์แบบเสมือนที่อยู่ใน Network เดียวกันว่าเครื่องใดมีลักษณะการทำงานที่คล้ายคลึงกันบ้าง
5. Naïve Bayes เป็นอัลกอริธึมที่ต้องการผู้สอนโดยความแม่นยำในการทำงานนั้นขึ้นอยู่กับชุดข้อมูลที่นำมาสอน ซึ่งถ้าข้อมูลนี้มีพฤติกรรมการใช้งานทรัพยากรใกล้เคียงกับเครื่องที่ต้องการจำแนกก็จะให้มีความแม่นยำในการจำแนกมากขึ้น ฉะนั้นในการใช้งานอัลกอริธึมจึงควรมีการสอนข้อมูลที่ใกล้เคียงการใช้งานจริงเรื่อยๆ
6. อัลกอริธึมและไลบรารีที่นำมาใช้งานนั้นมีความซับซ้อนผู้จัดทำจึงต้องใช้เวลาในการศึกษาและทดสอบการทำงานเพื่อนำมาปรับใช้ในการวิเคราะห์ข้อมูล ซึ่งภายอนาคตนั้นผู้จัดทำอาจทำการศึกษาผ่านสื่ออิเล็กทรอนิกส์ต่างๆเพื่อช่วยลดระยะเวลาในการศึกษาและทดสอบการทำงาน

5.3 แนวทางการพัฒนาต่อ

1. ปรับปรุงให้ระบบสามารถตรวจสอบข้อมูลรายละเอียดเบื้องต้นภายในศูนย์ข้อมูลก่อนที่จะทำข้อมูลไปใช้ในการวิเคราะห์ความสัมพันธ์ เพื่อลดจำนวนเครื่องที่จะต้องนำมาใช้ในการตรวจสอบ
2. ปรับปรุงให้ระบบสามารถรับรูปแบบข้อมูลที่ไม่ใช่ของ VMware เช่น XEN, KVM เป็นต้น
3. ปรับปรุงให้ระบบใช้งานได้สะดวกยิ่งขึ้นโดยอาจให้ผู้ใช้งานสามารถเรียกใช้ผ่าน API ได้

บรรณานุกรม

[1] Roger Wattenhofer. (2014, Aug). Principles of Distributed Computing [Online]. Available: http://dcg.ethz.ch/lectures/podc_allstars/lecture/podc.pdf

[2] Cisco System. (2014, July). Virtualization & Cloud Computing [Online]. Available: http://www.cisco.com/web/TH/about/articles/virtualisation.html

[3] ICT and Services. (2012, May). ความหมายของ Cloud Computing [Online]. Available: http://ictandservices.blogspot.com/2012/05/cloud-computing.html

[4] IBM. (2012, Oct) What is the Hadoop Distributed File System (HDFS) –United States [Online]. Available: http://www-01.ibm.com/software/data/infosphere/hadoop/hdfs/

[5] IBM. (2012, Oct). What is Map Reduce [Online]. Available: http://www-01.ibm.com/software/data/infosphere/hadoop/mapreduce/

[6] James Joyce. (2003, Jun). Bayes's Theorem [Online]. Available: http://plato.stanford.edu/entries/bayes-theorem/

[7] Apache.org. (2014, Oct). Welcome to Apache™ Hadoop®! [Online]. Available: http://hadoop.apache.org/

[8] VMware. (2014, Aug). vCenter Server [Online]. Available: http://www.vmware.com/products/vcenter-server/

[9] Dattatrey Sindol. (2014, Nov). SQL Server Big Data Tips [Online]. Available: http://www.mssqltips.com/sql-server-tip-category/208/big-data/

[10] VMware. (2014, Aug). vSphere 5.5 Documentation Center [Online]. Available: http://pubs.vmware.com/vsphere-55/index.jspXiufeng Liu. 2014. Survey of real-time processing systems for big data. IDEAS '14 Proceedings of the 18th International Database Engineering & Applications Symposium Pages 356-361

[11] Li Ruan, Jinbin Peng, Limin Xiao. 2013. CloudDVMM: Distributed Virtual Machine Monitor for Cloud Computing. Green Computing and

Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing Page 1853 – 1858

[12] Shyam M. Guthikonda. (2005, December) Kohonen Self Organizing Maps [Online]. Available: http://www.shy.am/wp-content/uploads/2009/01/kohonen-self-organizing-maps-shyam-guthikonda.pdf

[13] James Aspnes. (2014, June). Notes on Theory of Distributed Systems [Online]. Available: http://cs-www.cs.yale.edu/homes/aspnes/classes/465/notes.pdf

[14] Vangjee (2014, Nov) Computing Pearson's Correlation using Hadoop's Map/Reduce (M/R) Available: http://vangjee.wordpress.com/2012/02/29/computing-pearson-correlation-using-hadoops-mapreduce-mr-paradigm/

[15] ฉัตรศิริ ปิยะพิมลสิทธิ์. (2014, Nov) การวัดความสัมพันธ์ : Pearson's Sample Correlation Coefficient. Available: http://www.watpon.com/Elearning/pearson.pdf

[16] Mitchell, T. *Machine Learning*. McGraw-Hill. 1997.

[17] DELL Software. (2015, Mar) Naïve bayes [Online] Available : http://www.statsoft.com/textbook/naive-bayes-classifier

[18] Jeffrey D. Ullman. "Data mining" in Mining of Massive Datasets , Stanford University , 2014, ch. 1 ,pp. 1-19

[19] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2006.

[20] Deborah J. Rumsey. (2014, Nov). How to Interpret a Correlation Coefficient r [Online]. Available: http://www.dummies.com/how-to/content/how-to-interpret-a-correlation-coefficient-r.html