ใบรับรองวิทยานิพนธ์

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมคอมพิวเตอร์)

ปริญญา

วิศวกรรมคอมพิวเตอร์ วิศวกรรมคอมพิวเตอร์

สาขา **ภาควิชา**

เรื่อง การตรวจจับโฮสต์สแปมด้วยแอนท์โคโลนีออพติไมเซชัน

Spam Host Detection Using Ant Colony Optimization

นามผู้วิจัย นายอภิชาติ ทวีศิริเวทย์

ได้พิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผู้ช่วยศาสตราจารย์อานนท์ รุ่งสว่าง, Ph.D.)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

(ผู้ช่วยศาสตราจารย์ภุชงค์ อุทโยภาศ, Ph.D.)

หัวหน้าภาควิชา

(ผู้ช่วยศาสตราจารย์ภุชงค์ อุทโยภาศ, Ph.D.)

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

(รองศาสตราจารย์กัญจนา ธีระกุล, D.Agr.)

คณบดีบัณฑิตวิทยาลัย

วันที่ ..... **เดือน** ..... **พ.ศ.** .....

วิทยานิพนธ์

เรื่อง

การตรวจจับโฮสต์สแปมด้วยแอนท์โคโลนีออพติไมเซชัน

Spam Host Detection Using Ant Colony Optimization

โดย

นายอภิชาติ ทวีศิริเวทย์

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

เพื่อความสมบูรณ์แห่งปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมคอมพิวเตอร์)

พ.ศ. 2555

อภิชาติ ทวีศิริเวทย์ 2555: การตรวจจับโฮสต์สแปมด้วยแอนท์โคโลนีออพติไมเซชัน ปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมคอมพิวเตอร์) สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก: ผู้ช่วยศาสตราจารย์ อานนท์ รุ่งสว่าง, Ph.D. 54 หน้า

การโกงผลลัพธ์การค้นหาจากเสิร์จเอ็นจินเรียกว่าสแปม ซึ่งจะทำให้ส่งผลอย่างยิ่งต่อความน่าเชื่อถือของผู้ใช้ที่มีต่อเสิร์จเอ็นจิน เมื่อความน่าเชื่อถือต่ำลงผู้ใช้จะมองหาทางเลือกอื่น โดยมีแรงจูงใจจากผลตอบแทนที่คุ้มค่ามากขึ้นเรื่อยๆ เนื่องจากต้นทุนการทำสแปมลดลงเรื่อยๆ บวกกับความนิยมของอินเตอร์เน็ตมีมากขึ้น จึงมีความจำเป็นอย่างมากที่เสิร์จเอ็นจินต้องจัดการกับปัญหาเหล่านี้เพื่อรักษาจำนวนผู้ใช้ จากปัญหาที่ได้กล่าวมาข้างต้นวิทยานิพนธ์นี้จึงได้นำเสนอการจำแนกประเภทโฮสต์สแปมในรูปแบบของกฎ เป็นการรวมกันของแนวคิดจากงานวิจัยที่ผ่านมา โดยเสนอวิธีการจำแนกประเภทด้วยกฎ ที่สร้างมาจากทิศทางการเชื่อมโยงของเพื่อนบ้านและแก้ไขปัญหาเรื่องเส้นทางด้วยอัลกอริทึมแอนท์โคโลนีออพติไมเซชัน จากนั้นอาศัยคุณลักษณะด้านเนื้อหาและเส้นการเชื่อมโยงของโฮสต์เพื่อสร้างกฎ งานวิจัยนี้เลือกชุดข้อมูล Web Spam UK 2006 ซึ่งเป็นที่นิยมใช้ในงานทางด้านนี้มาทำการวัดผลซึ่งสามารถดูจากบทที่ 4 พร้อมบทวิเคราะห์และแนวทางในการพัฒนาต่อไป

_____ _____ ____/____/_____

ลายมือชื่อนิสิต ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

Apichat Tawesiriwate 2012: Spam Host Detection Using Ant Colony Optimization. Master of Engineering (Computer Engineering), Major Field: Computer Engineering, Department of Computer Engineering. Thesis Advisor: Assistant Professor Arnon Rungsawang, Ph.D. 54 pages.

Techniques for cheating results of search engine to get higher rank than they deserve call “spam”, the effect of web spamming will reduce search reliability. If search engine has low reliability then user may pay attention to search engine. Motivation of the spammer is the return of investment and tend of cost are continue decrease but opposite tend of traffic in internet are continue increase. So from spam problem, this thesis present spam hosts detection by rules. We exploit structure of the hosts graph and combine idea from other researches. Then we use ant colony optimization for find paths. Experiments on the WEBSPAM-UK2006 dataset show that the proposed method provides higher precision in detecting spam, comparing to the decision tree model. See the result in lesson 4.

_____ _____ ____/____/_____

Student’s signature Thesis Advisor’s signature

กิตติกรรมประกาศ

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์อานนท์ รุ่งสว่าง อาจารย์ที่ปรึกษาวิทยานิพนธ์หลักที่ให้โอกาส คำแนะนำ ความรู้ การตรวจสอบแก้ไขงานวิจัย ปัจจัยที่เกี่ยวข้องและสิ่งต่างๆ อีกมากมายที่เป็นประโยชน์กับงานวิจัยและข้าพเจ้า ขอขอบพระคุณผู้ช่วยศาสตราจารย์ภุชงค์ อุทโยภาศอาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ที่ช่วยให้ความรู้และข้อคิดเห็นที่ดีต่างๆ

ขอขอบพระคุณ ดร.บัณฑิต มนัสเกษมศักดิ์ที่ให้คำปรึกษาและช่วยเหลือมาโดยตลอด โดยเริ่มตั้งแต่หาหัวข้อทำวิจัย ความรู้ต่างๆ ขั้นตอนการทำงานวิจัย แก้ไขจุดบกพร่องของงาน การตีพิมพ์ผลงาน คำปรึกษาในการเขียนวิทยานิพนธ์ และอื่นๆ อีกมากมาย ขอขอบคุณ คุณกำธร พันธุมะผลที่ให้คำปรึกษาในเรื่องทฤษฎี ขอขอบคุณคุณ Carlos Castillo ที่ตอบคำถามเกี่ยวกับข้อมูลที่ใช้ในวิทยานิพนธ์นี้

ขอขอบพระคุณอาจารย์คณะวิศวกรรมศาสตร์ภาควิชาคอมพิวเตอร์ที่สละเวลามาสอนและให้ความรู้ทุกท่าน ขอขอบคุณเจ้าหน้าที่โครงการบัณฑิตศึกษาภาควิชาวิศวกรรมคอมพิวเตอร์ที่ช่วยประสานงานและเรื่องของเอกสารให้เป็นไปด้วยดี

ขอขอบพระคุณพ่อคุณแม่ที่ได้สนับสนุนการศึกษามาโดยตลอด

อภิชาติ ทวีศิริเวทย์

กุมภาพันธ์ 2555

สารบัญ

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญภาพ

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

การตรวจจับโฮสต์สแปมด้วยแอนท์โคโลนีออพติไมเซชัน

Spam Host Detection Using Ant Colony Optimization

คำนำ

การเติบโตของจำนวนผู้ใช้อินเตอร์เน็ตที่มากขึ้นเรื่อยๆ ส่งผลต่อจำนวนเว็บไซต์ที่มากขึ้นตามไปด้วยเช่นกัน วิธีที่ได้รับความนิยมจากผู้ใช้ในการเข้าถึงข้อมูลจากเว็บเพจที่ต้องการคือผ่านระบบสืบค้นหรือเสิร์จเอ็นจิน (search engine) การทำงานของระบบสืบค้นคือผู้ใช้ป้อนคำสำคัญ (key word) ไประบบสืบค้นจะให้ผลลัพธ์เป็นเว็บเพจมาจำนวนหนึ่งซึ่งเกี่ยวข้องกับคำสำคัญ ผู้สร้างเว็บไซต์ที่ขายโฆษณาจำนวนมากอยากให้เว็บไซต์ของตนมีลำดับที่ดีกว่าคู่แข่งเพราะผู้ใช้จะให้ความสำคัญกับผลลัพธ์ลำดับต้นๆ ก่อนทำให้ผู้สร้างเว็บไซต์จำนวนหนึ่งพยายามโกงการค้นหา เรียกผู้สร้างเว็บไซต์เหล่านี้ว่าสแปมเมอร์ (spammer)

แรงจูงใจในการสร้างเว็บเพจสแปมเนื่องจากผลตอบแทนเป็นจำนวนเงินที่ดีและคุ้มค่ากับเจ้าของเว็บไซต์ ในทางกลับกันจะเป็นผลเสียต่อเสิร์จเอ็นจินเนื่องจากไปลดความน่าเชื่อถือลง เพราะว่าผู้ใช้คาดหวังว่าเว็บเพจอันดับต้นๆ ต้องมีคุณภาพที่ดีและเกี่ยวข้องกับคำสำคัญที่ผู้ใช้ต้องการ อีกทั้งตลาดเสิร์จเอ็นจินมีการแข่งขันกันสูงเมื่อความน่าเชื่อถือลดลงผู้ใช้จะเริ่มหันไปใช้บริการจากเสิร์จเอ็นจินอื่น ทำให้เสิร์จเอ็นจินต้องหาวิธีการรับมือกับปัญหาสแปม

การแบ่งประเภทเว็บเพจสแปมได้ถูกศึกษาโดย Gyougyi (Gyöngyi and Garcia-Molina, 2005a) สามารถจัดแบ่งรูปแบบออกเป็นกลุ่มใหญ่ๆ ได้คือ 1) การสแปมเนื้อหาของเว็บเพจ สแปมเมอร์จะใส่คำจำนวนมากไปในเว็บเพจ ซึ่งทำให้เสิร์จเอ็นจินคิดว่าเว็บเพจเหล่านี้เกี่ยวข้องคำสำคัญตามจำนวนคำที่มีอยู่ภายในเว็บเพจ แต่เมื่อผู้ใช้อ่านจะพบว่าบางครั้งอ่านแล้วไม่สามารถเข้าใจได้ เนื่องจากเว็บเพจเหล่านั้นถูกสร้างมาจากโปรแกรมอัตโนมัติ 2) การสแปมที่ลิงค์ (link) เนื่องจากอัลกอริทึมที่เสิร์จเอ็นจินนิยมใช้คือเพจแร็งค์ (PageRank (Page *et al*., 1998)) ซึ่งอัลกอริทึมเพจแร็งค์จะเรียงลำดับเว็บเพจตามจำนวนการถูกอ้างถึง สแปมเมอร์จะสร้างเว็บเพจขึ้นมาเป็นจำนวนมากเพื่อทำการอ้างถึงเว็บเพจที่ต้องการเพิ่มค่าคะแนนเพจแร็งค์ มีนักวิจัยหลายท่านพยายามแก้ไข

ปัญหาสแปมด้วยวิธีการที่หลากหลายแตกต่างกันออกไป แต่ในงานวิจัยนี้จะแก้ไขปัญหาด้วยการตรวจจับโดยใช้วิธีการของเครื่องจักรเรียนรู้ (machine learning techniques)

ผลลัพธ์ที่ได้จากเครื่องจักรเรียนรู้ในงานวิจัยนี้คือกฎแบบถ้า-แล้ว(if-then) โดยจะคัดแยกระดับของโฮสต์ กล่าวคือผลลัพธ์จากการทำนายของกฎประกอบไปด้วยโฮสต์ปกติและโฮสต์สแปม กฎที่มีผลลัพธ์การทำนายเป็นโฮสต์ปกติจะมีพื้นฐานมาจากแนวคิดของอัลกอริทึมทรัสแร็งค์ (TrustRank (Gyöngyi *et al*., 2004)) ส่วนกฎที่มีผลลัพธ์การทำนายเป็นโฮสต์สแปมจะมีพื้นฐานมาจากแนวคิดของอัลกอริทึมแอนติทรัสแร็งค์ (Anti-TrustRank (Krishnan and Rashmi, 2006)) สองอัลกอริทึมนี้จะเสนอแนวคิดการเชื่อมโยงระหว่างเว็บเพจ ในงานวิจัยนี้จะนำเอาสองแนวคิดนี้มาใช้ในการสร้างเส้นทาง เพื่อที่จะนำเส้นทางนั้นมาสร้างเป็นกฎ ในขั้นตอนการหาเส้นทางจะอาศัยอัลกอริทึมแอนท์โคโลนีออพติไมเซชันเข้ามาใช้เพราะว่าอัลกอริทึมแอนท์โคโลนีออพติไมเซชันสามารถแก้ปัญหาที่อยู่ในรูปแบบของกราฟได้ถ้าหากเรามองโฮสต์เป็นโหนดในกราฟและมีการอ้างถึงเป็นเส้นเชื่อมระหว่างโหนด เมื่อได้กฎมาก็จะนำมาคัดแยกโฮสต์ปกติและโฮสต์สแปมออกจากกัน

ในส่วนขั้นตอนการทดลอง จะใช้ชุดข้อมูลเว็บสแปมจาก WEB Spam UK 2006 (Castillo *et al*., 2006) ซึ่งเป็นชุดข้อมูลที่ได้รับความนิยมกับงานวิจัยทางด้านนี้ จากการทดลองจะพบว่าสามารถสร้างกฎได้ดีกว่าอัลกอริทึม C4.5(Quinlan, 1993) ที่นิยมใช้ในปัจจุบัน

วัตถุประสงค์

1. ตรวจจับโฮสต์สแปมด้วยอัลกอริทึมแอนท์โคโลนีออพติไมเซชัน

2. พัฒนาอัลกอริทึมตรวจจับให้ใช้ประโยชน์จากโครงสร้างกราฟและคุณลักษณะของโฮสต์เพื่อนบ้าน

ขอบเขตงานวิจัย

1. ตรวจจับสแปมในระดับของโฮสต์

การตรวจเอกสาร

ความรู้พื้นฐานและงานวิจัยที่เกี่ยวข้อง

ในส่วนนี้จะกล่าวถึงความรู้พื้นฐานที่สแปมเมอร์ใช้สร้างเว็บเพจสแปม รวมถึงงานวิจัยที่เกี่ยวข้องกับการแก้ไขปัญหาเรื่องสแปมและความรู้เบื้องต้นเกี่ยวกับอัลกอริทึมแอนท์โคโลนีออพติไมเซชัน

ลักษณะสแปม

Gyöngyi and Garcia-Molina (2005a) ได้ศึกษาคุณลักษณะของเว็บเพจที่เป็นเว็บเพจสแปม เว็บเพจสแปมคือเว็บเพจที่พยายามโกงการจัดอันดับจากเสิร์จเอ็นจินและมักมีเนื้อหาที่ไม่ต้องการจากผู้ใช้ ตัวอย่าง จากปี 2004 ค้นคืนด้วยคำสำคัญ Pharmacy ไปยังเสิร์จเอ็นจิน หลังจากนั้นเสิร์จเอ็นจินคืนผลลัพธ์มาเป็นเว็บเพจจำนวนสิบเว็บเพจเรียงลำดับความความเกี่ยวข้อง แต่ผู้ใช้งานกลับพบว่ามีเว็บเพจที่เกี่ยวข้องเพียงแค่สามเว็บเพจเท่านั้นนอกนั้นเป็นเว็บเพจเกี่ยวกับการขายของถูกและเว็บเพจเกี่ยวกับการลดราคาสินค้า การสร้างเว็บเพจสแปมมักอาศัยเทคนิคที่เรียกว่าเสิร์จเอ็นจินออพติไมเซชัน (search engine optimization) ซึ่งเทคนิคนี้สามารถเพิ่มคะแนนให้กับเว็บเพจได้ แต่มีผู้พัฒนาเว็บไซต์กลุ่มหนึ่งใช้เทคนิคนี้เพื่อพยายามโกงจนเกินขอบเขตเพื่อให้เว็บไซต์ของตนได้อับดับที่ดีในการค้นหาเราจะเรียกเว็บไซต์เหล่านี้ว่าสแปม เทคนิคที่ใช้เว็บเพจสแปมใช้เพิ่มคะแนนจากเสิร์จเอ็นจินมีแบ่งได้เป็นสองกลุ่มใหญ่ๆคือ การสแปมที่เนื้อหาและการสแปมที่ลิงค์ของเว็บเพจ นอกจากนั้นยังมีวิธีการที่จะซ่อนเนื้อหาจากผู้ใช้อีกด้วย ดังรายละเอียดต่อไปนี้

1. การสแปมเนื้อหา (content spamming)

คือการทำสแปมโดยเพิ่มคำสำคัญจำนวนมากไปในส่วนต่างๆ ของเอกสารเพื่อให้เสิร์จเอ็นจินเข้าใจว่า เอกสารเหล่านี้เกี่ยวข้องกับคำถามของผู้สืบค้น ซึ่งการสแปมเนื้อหานั้นสามารถพิจารณาได้เป็น 2 มุมมอง ได้แก่

1.1 มุมมองการสแปมตามประเภทของแท็ก (tag)

1.1.1 การสแปมแท็กบอดี้ (body tag spamming) บอดี้เป็นส่วนเนื้อหาของเอกสาร ดังนั้นการทำสแปมจึงเป็นการเพิ่มคำสำคัญที่ต้องการจำนวนมากลงในเนื้อหา ตัวอย่างเช่น

```
<Body>Poker Poker Poker … Poker </Body>
```

1.1.2 การสแปมแท็กไตเติ้ล (titile tag spamming) เนื่องจากโดยปกติเสิร์จเอ็นจินจะให้น้ำหนักความสำคัญกับหัวข้อเป็นพิเศษ ดังนั้นการเพิ่มคำสำคัญจำนวนมากลงในแท็กไตเติ้ลจึงสามารถเพิ่มโอกาสที่จะถูกค้นคืน และถูกจัดในลำดับแรกๆ ได้เป็นอย่างมาก

1.1.3 การสแปมแท็กเมตา (meta tag spam) คุณประโยชน์หนึ่งของแท็กเมตาคือใช้ระบุถึงคำสำคัญต่างๆ ที่กล่าวในเอกสารเว็บเพจนั้น อย่างไรก็ดีสแปมเมอร์ (spammer) มักอาศัยแท็กนี้ในการเพิ่มคำที่ต้องการจำนวนมาก ทำให้เสิร์จเอ็นจินในปัจจุบันได้ลดความสำคัญของแท็กเมตาลง ตัวอย่างการเขียนแท็กเมตา เช่น

```
<Meta name="keywords" content="buy, cheap, cameras, lens, accessories, nikon, canon">
```

1.1.4 ชื่อยูอาร์แอล (url) เป็นตัวระบุที่อยู่ของเอกสารนั้นๆ โดยทั่วไปแล้วเว็บไซต์ส่วนใหญ่มักตั้งชื่อยูอาร์แอลให้สัมพันธ์กับเนื้อหาเอกสารภายนั้น ดังนั้นเสิร์จเอ็นจินจึงให้ความสำคัญกับชื่อยูอาร์แอล เว็บไซต์สแปมจะสร้างชื่อยูอาร์แอลจากคำจำนวนมากเช่น buy-canon-rebel-300d-lens-case.camerasx.com

1.1.5 การสแปมแท็กแองเคอร์ (anchor tag spamming) คุณสมบัติพิเศษของเอกสารเว็บเพจคือสามารถเชื่อมโยงไปยังเว็บเพจอื่นๆ ได้ โดยการใช้แท็กแองเคอร์ซึ่งผู้พัฒนาเว็บเพจมักระบุคำสำคัญ ที่เรียกว่า "แองเคอร์เท็กซ์" (anchor text) ไว้ภายใต้แท็กแองเคอร์ดังกล่าว เพื่อให้ผู้อ่านสนใจและคลิก (click) เพื่อไปยังเว็บเพจปลายทางนั้น ดังนั้นเสิร์จเอ็นจินจึงมักให้ความสำคัญกับแองเคอร์เท็กซ์ ซึ่งเป็นจุดอ่อนให้สแปมเมอร์ใช้ในการเพิ่มคำที่ต้องการจำนวนมาก ตัวอย่างการเขียนแท็กแองเคอร์เพื่อเชื่อมโยงไปยังเว็บเพจปลายทาง target.html ซึ่งได้ระบุแองเคอร์เท็กซ์ต่างๆ เช่น

<A href="target.html">free, great deals, cheap, inexpensive, cheap, free</A>

1.2 มุมมองสแปมตามลักษณะการกระทำ

1.2.1 การทำซ้ำ สแปมเมอร์จะเพิ่มคำจำนวนมากที่เกี่ยวข้องกับเอกสารด้วยคำสำคัญที่ต้องการ เมื่อผู้ใช้ค้นหาด้วยคำสำคัญดังกล่าว มีความเป็นไปได้สูงที่เอกสารเหล่านี้จะถูกให้ความสำคัญจากเสิร์จเอ็นจิน

1.2.2 การเพิ่มคำที่ไม่เกี่ยวกับเอกสาร สแปมเมอร์จะเพิ่มคำจากพจนานุกรมให้กับเอกสารของตนเองเพื่อให้เสิร์จเอ็นจินสามารถค้นหาได้จากคำสำคัญที่หลากหลาย ส่วนใหญ่มักจะเป็นคำเฉพาะในวงการ เนื่องจากเอกสารเหล่านี้มีน้อยใส่คำเพิ่มไปจำนวนไม่มากนักก็เพียงพอที่จะปรากฏในลำดับต้นๆ ได้

1.2.3 การแทรกคำ สแปมเมอร์จะทำการคัดลอกเอกสารมาจากที่ต่างๆ จากนั้นจะสุ่มตำแหน่งในการแทรกคำ โดยผลลัพธ์จะต่างจากการเพิ่มคำคือ สามารถอำพรางการทำสแปมได้ดีกว่าการเพิ่มคำ เอกสารขึ้นมาเพิ่มเติมคำซึ่งให้ผลดีกับเอกสารเฉพาะวงการทำให้เสิร์จเอ็นจินเข้าใจผิด

1.2.4 การรวมเอกสาร วิธีนี้ทำให้สแปมเมอร์สร้างเอกสารได้อย่างรวดเร็ว แนวคิดคือสุ่มเอาประโยคของเอกสารที่มีอยู่มารวมกัน ตัวอย่างของการรวมประโยคจากต่างเอกสาร

The objective of a search engine is to provide high-quality results by correctly identifying. An unjustifiably favorable boosting technique, i.e., methods through which one seeks relies on the identification of some common features of spam pages.
ซึ่งมาจากการรวม 3 ประโยคดังนี้

1. The objective of a search engine is to provide high-quality results by correctly identifying.

2. An unjustifiably favorable boosting technique, i.e.,

3. methods through which one seeks relies on the identification of some common features of spam pages.

2. การสแปมลิงค์ (link spamming)

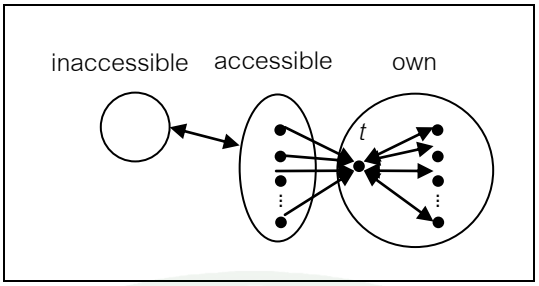
การทำสแปมส่วนนี้จะมีความสำคัญอย่างมากเนื่องจากส่งผลโดยตรงต่อค่าคะแนนเว็บเพจ แม้ว่าเสิร์จเอ็นจินของแต่ละที่จะใช้อัลกอริทึมในการคิดค่าคะแนนเว็บเพจไม่เหมือนกัน แต่อัลกอริทึมที่มีชื่อเสียงเช่นเพจแร็งค์หรือฮิตส์(HITS (Kleinberg, 1999)) ค่าคะแนนจะพิจารณาจากปริมาณและทิศทางของลิงค์ สแปมเมอร์ที่ต้องการเพิ่มค่าคะแนนเว็บเพจจึงสร้างลิงค์จำนวนหนึ่งขึ้นมาเอง โดยไม่จำเป็นต้องไปพัฒนาคุณภาพของเอกสารเพื่อที่จะให้เว็บเพจอื่นทำลิงค์ชี้เข้าหาเว็บเพจของตน

2.1 การเพิ่มคะแนนเพจแร็งค์ สแปมเมอร์จะสร้างลิงค์จากเว็บเพจจำนวนหนึ่งซึ่งชี้ไปยังเว็บเพจที่ต้องการโกงคะแนนเพจแร็งค์ สามารถแบ่งกลุ่มเว็บเพจจากการใช้งานของสแปมเมอร์ได้ออกเป็น 3 ประเภทคือ

2.1.1 กลุ่มเว็บเพจที่สแปมเมอร์ไม่มีสิทธิแก้ไข (inaccessible page) กลุ่มเว็บเพจเหล่านี้สแปมเมอร์จะไม่สามารถแทรกลิงค์ไปยังเว็บเพจอื่นได้ แต่สแปมเมอร์จะยังสามารถสร้างลิงค์ชี้ไปหาเว็บเพจเหล่านี้ได้

2.1.2 กลุ่มเว็บเพจที่สามารถทำการแก้ไขหรือเพิ่มเติมเนื้อหาได้บางส่วน (accessible page) เว็บเพจเหล่านี้จะตกเป็นเครื่องมือของสแปมเมอร์ โดยการแทรกลิงค์ชี้ไปยังเว็บเพจเป้าหมาย ตัวอย่างกลุ่มเว็บเพจนี้เช่น เว็บบอร์ด (web board) ในการแสดงความคิดเห็น (comment) สแปมเมอร์จะแทรกลิงค์ไปยังเว็บเพจที่ต้องการ

2.1.3 กลุ่มเว็บเพจที่เป็นของสแปมเมอร์ (own page) เว็บเพจเหล่านี้จะถูกควบคุมทั้งหมดโดยสแปมเมอร์ ในกลุ่มนี้จะมีเว็บเพจเป้าหมาย *t* ที่สแปมเมอร์ต้องการเพิ่มค่าคะแนน (target page)



ภาพที่ 1 ประเภทของเว็บเพจแยกตามการใช้งานของสแปมเมอร์

เมื่อสแปมเมอร์มีเว็บเพจที่ต้องการแล้วจะพยายามสร้างลิงค์จากกลุ่มเว็บเพจจำพวก ที่สามารถแก้ไขได้บางส่วนไปยังกลุ่มเว็บเพจของสแปมเมอร์ ซึ่งการเชื่อมต่อภายในกลุ่มเว็บเพจของสแปมเมอร์จะมีโครงสร้างที่แตกต่างกันออกไปตามวิธีการของสแปมเมอร์ โดยมีปัจจัยเรื่องทรัพยากรณ์ในการสร้างเว็บเพจเป็นหลัก แต่แนวคิดจะไม่แตกต่างกันมากนักคือมีเว็บเพจเป้าหมาย (target page) ที่ต้องการค่าคะแนนเพจแร็งค์สูง ถ้าต้องการค่าคะแนนเพจแร็งค์สูงก็ต้องมีลิงค์ที่มาจากเว็บเพจที่ค่าคะแนนเพจแร็งค์สูงด้วย สแปมเมอร์จะพยายามเพิ่มค่าคะแนนเพจแร็งค์ให้กับกลุ่มเว็บเพจของสแปมเมอร์ ให้มากที่สุดเพื่อที่กลุ่มเว็บเพจของสแปมเมอร์ จะสามารถไปเพิ่มคะแนนให้กับเว็บเพจเป้าหมาย แต่การเชื่อมโยงภายในกลุ่มเว็บเพจของสแปมเมอร์ จะไม่มีลิงค์ชี้ออกไปจากกลุ่มเนื่องจากเป็นการไปเพิ่มคะแนนให้กับเว็บเพจอื่นที่ไม่ใช่ของสแปมเมอร์ซึ่งสแปมเมอร์จะไม่ได้ผลตอบแทนอะไร

2.2 วิธีสแปมเพื่อเพิ่มค่าฮับและออทอริตี้ในอัลกอริทึมฮิตส์ สามารถแบ่งออกตามทิศทางของลิงค์ได้เป็นสองประเภทคือ

2.2.1 ลิงค์ชี้ออก (Outgoing links) สแปมเมอร์จะสร้างลิงค์ชี้ออกไปยังเว็บเพจที่มีชื่อเสียง เพื่อเพิ่มคะแนนจากอัลกอริทึมฮิตส์ โดยที่เว็บเหล่านี้ส่วนใหญ่จะมาจาก dmoz.org (DMOZ Open directory project) ซึ่งเป็นแหล่งจัดหมวดหมู่ของเว็บขนาดใหญ่ จากนั้นสแปมเมอร์จะเลือกหมวดหมู่ เพื่อสร้างลิงค์ชี้ไปหาเว็บเพจในหมวดหมู่นั้น

2.2.2 ลิงค์ชี้เข้า (Incoming links) มีวิธีดังต่อไปนี้

ก. เว็บเพจกับดัก (Honey pot) คือการคัดลอกเอกสารที่มีคนสนใจจากแหล่งข้อมูลต่างๆ จากนั้นจึงทำการแก้ไขเอกสาร โดยการแทรกลิงค์ชี้ไปยังเว็บเพจเป้าหมาย

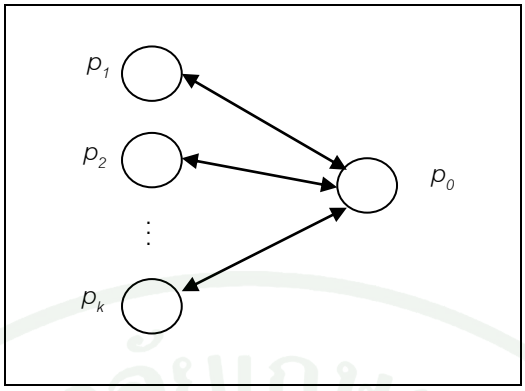
ข. การแทรกลิงค์ สแปมเมอร์จะทำการหาเว็บเพจที่มีค่าเพจแร็งค์สูงและสามารถแก้ไขหรือเพิ่มเอกสารได้ซึ่งส่วนใหญ่เป็นพวกเว็บบอร์ด (web board) วิกิพีเดีย (Wikipedia) หรือ บล็อก (blog) ที่เปิดให้คนทั่วไปได้รับความยินยอมในการเขียน เมื่อสแปมเมอร์พบเว็บเพจเหล่านี้ก็จะทำการแก้ไขหรือสร้างเว็บเพจขึ้นมาใหม่ เพื่อแทรกลิงค์ไปยังเว็บเพจของตนเอง

ค. การแลกเปลี่ยนลิงค์ กลุ่มผู้ที่ทำสแปมโครงสร้างลิงค์ มักให้เว็บเพจของตนเองแลกเปลี่ยนลิงค์กับเว็บเพจของสแปมเมอร์ที่เป็นสมาชิกกลุ่ม โดยมีเป้าหมายเพื่อคะแนนสำหรับการจัดลำดับเพิ่มขึ้น

ง. การสร้างสแปมฟาร์ม สแปมเมอร์จะทำการสร้างเว็บเพจขึ้นมาเองเป็นจำนวนมากเพื่อสร้างโครงสร้างลิงค์ของตนเองขึ้นมา ในสมัยก่อนจะมีต้นทุนที่สูงแต่ทุกวันนี้เป็นวิธีที่สแปมเมอร์นิยมใช้กันเนื่องจากต้นทุนในการสร้างเว็บไซต์ลดลง ซึ่งต้นทุนประกอบไปด้วย ค่าโดเมน (cost of domain) และค่าเช่าโฮสต์ (cost of web hosting)

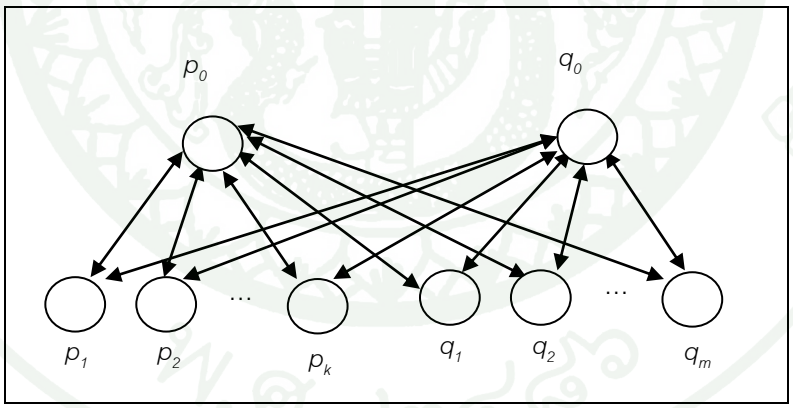
2.3 ลักษณะโครงสร้างลิงค์ Gyöngyi and Garcia-Molina, 2005b ได้ศึกษาลักษณะโครงสร้างของลิงค์สแปมในแบบต่างๆ ซึ่งจะส่งผลต่อคะแนนเพจแร็งค์ไม่เหมือนกัน แต่ทุกรูปแบบจะประกอบไปด้วยเว็บเพจส่งเสริม (boosting page) และเว็บเพจเป้าหมาย (target page) สามารถแบ่งกลุ่มโครงสร้างได้ดังนี้

2.3.1 โครงสร้างสแปมแบบหนึ่งเว็บเพจเป้าหมายหนึ่งสแปมฟาร์ม (spam farm, คือกลุ่มของเว็บเพจส่งเสริม) จากภาพที่ 2 โครงสร้างนี้สแปมเมอร์จะมีต้นทุนที่น้อยคือเป็นเจ้าของเว็บเพจเป้าหมาย p_0 เพียงเว็บเพจเดียว สแปมเมอร์จะพยายามสร้างลิงค์จากเว็บต่างๆ ชี้มาเว็บเพจ p_0 ให้มากที่สุด เว็บจำพวกที่ถูกสแปมเมอร์ใช้ตัวอย่างเช่นเว็บบอร์ดที่ขาดการตรวจสอบจากผู้ดูแล จากภาพที่ 2 เว็บเพจส่งเสริมคือเว็บเพจ p_1 ถึงเว็บเพจ p_k



ภาพที่ 2 โครงสร้างสแปมแบบหนึ่งเว็บเพจเป้าหมายหนึ่งสแปมฟาร์ม

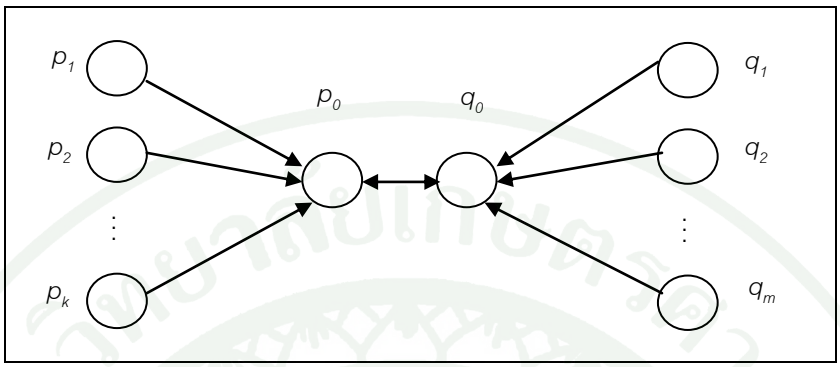
2.3.2 โครงสร้างสแปมแบบสองสแปมฟาร์มแบ่งปันเป้าหมาย จากภาพที่ 3 ประกอบไปด้วยสแปมฟาร์ม p และสแปมฟาร์ม q มีเว็บเพจเป้าหมาย p_0 และ q_0 เว็บเพจที่เหลือ p_1 ถึง p_k และ q_1 ถึง q_m เป็นเว็บเพจส่งเสริม โครงสร้างนี้เว็บเพจส่งเสริมทุกเว็บเพจจะสร้างลิงค์ชี้ไปยังเว็บเพจเป้าหมาย p_0 และ q_0 โครงสร้างแบบนี้สแปมฟาร์มที่มีเว็บเพจน้อยกว่าจะได้รับการคะแนนเพจแร็งค์สูงขึ้นจากสแปมฟาร์มที่ใหญ่กว่า



ภาพที่ 3 โครงสร้างสแปมแบบสองสแปมฟาร์มแบ่งปันเป้าหมาย

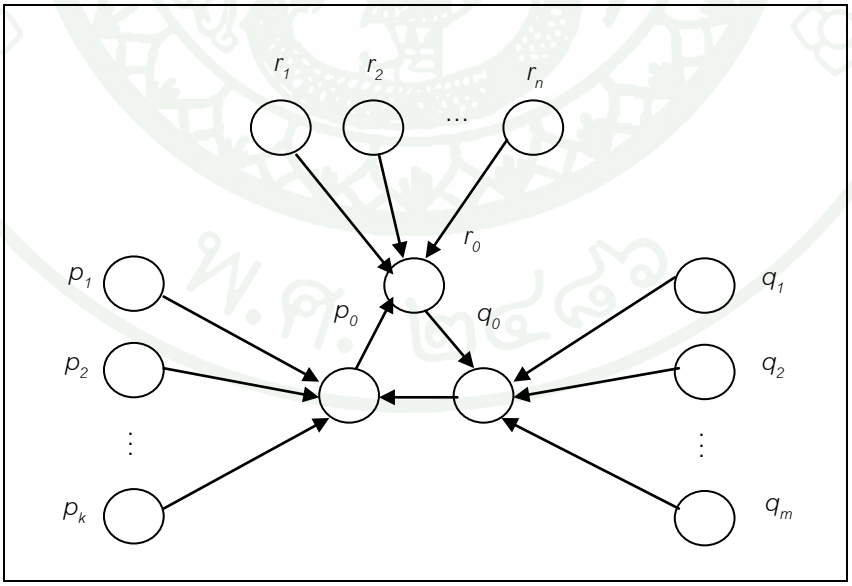
2.3.3 โครงสร้างสแปมแบบสองสแปมฟาร์มเชื่อมต่อผ่านเว็บเพจเป้าหมาย จากภาพที่ 4 ประกอบไปด้วยสแปมฟาร์ม p และสแปมฟาร์ม q มีเว็บเพจเป้าหมาย p_0 และ q_0 เว็บเพจที่เหลือ p_1 ถึง p_k และ q_1 ถึง q_m เป็นเว็บเพจส่งเสริม โครงสร้างสแปมแบบนี้จะไม่มีลิงค์เชื่อมต่อภายนอกสแปมฟาร์ม ยกเว้นที่เว็บเพจเป้าหมายเท่านั้นซึ่งจะมีลิงค์เชื่อมต่อกับเว็บเพจเป้าหมายของอีกสแปม

ฟาร์ม การเชื่อมต่อแบบนี้จะส่งผลกับเว็บเพจเป้าหมายของแต่ละสแปมฟาร์มจะมีคะแนนเพจแร็งค์สูงขึ้นเมื่อเปรียบเทียบกับไม่มีการเชื่อมต่อกันระหว่างสแปมฟาร์ม



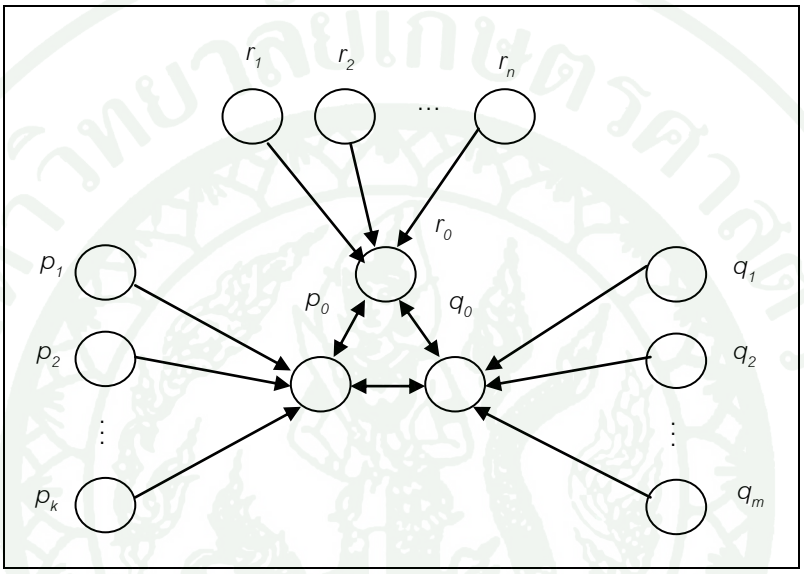
ภาพที่ 4 โครงสร้างสแปมแบบสองสแปมฟาร์มเชื่อมต่อผ่านเว็บเพจเป้าหมาย

2.3.4 โครงสร้างสแปมแบบวงแหวน จากภาพที่ 5 ประกอบไปด้วยสแปมฟาร์ม p , สแปมฟาร์ม q และสแปมฟาร์ม r มีเว็บเพจเป้าหมาย p_0, q_0 และ r_0 เว็บเพจที่เหลือ p_1 ถึง p_k, q_1 ถึง q_m และ r_1 ถึง r_n เป็นเว็บเพจส่งเสริม การเชื่อมต่อจะคล้ายกับของโครงสร้างสแปมที่ 2.2.3 เพียงแต่เพิ่มสแปมฟาร์มเข้าไป ซึ่งการเชื่อมต่อแบบนี้สามารถเพิ่มค่าคะแนนเพจแร็งค์ได้



ภาพที่ 5 โครงสร้างสแปมแบบวงแหวน

2.3.5 โครงสร้างสแปมแบบการเชื่อมต่อแบบสมบูรณ์ จากภาพที่ 6 ประกอบไปด้วย สแปมฟาร์ม p ,สแปมฟาร์ม q และสแปมฟาร์ม r มีเว็บเพจเป้าหมาย p_0, q_0 และ r_0 เว็บเพจที่เหลือ p_1 ถึง p_k, q_1 ถึง q_m และ r_1 ถึง r_n เป็นเว็บเพจส่งเสริม โครงสร้างนี้จะคล้ายกับโครงสร้างสแปมที่ 2.2.4 แต่จะแตกต่างกันที่การเชื่อมต่อของเว็บเพจเป้าหมาย ซึ่งจะเป็นการเชื่อมต่อแบบสมบูรณ์ (completely connected)



ภาพที่ 6 โครงสร้างสแปมแบบการเชื่อมต่ออย่างสมบูรณ์

3. เทคนิคการปิดบัง

บางครั้งสแปมเมอร์จะพยายามอำพรางการทำสแปมเพื่อปิดบังจากผู้ตรวจสอบที่เป็นมนุษย์ แต่วิธีการเหล่านี้จะส่งผลที่แตกต่างออกไปกับมุมมองของเสิร์จเอ็นจิน สามารถแบ่งประเภทเทคนิคการปิดบังได้ดังนี้

3.1 การซ่อนเนื้อหา (content hiding) เมื่อสแปมเมอร์เพิ่มคำหรือลิงค์ที่ต้องการไปในเว็บเพจ คำและลิงค์เหล่านั้นสามารถซ่อนจากสายตาผู้ใช้ได้ วิธีหนึ่งคือปรับสีตัวอักษรให้เหมือนกับพื้นหลัง อีกวิธีหนึ่งคือปรับขนาดให้เล็กลงจนไม่สามารถสังเกตได้ การซ่อนส่งผลให้ผู้ใช้ไม่สามารถเห็นได้จากเว็บบราวเซอร์ (web browser) ยกเว้นผู้ใช้จะทำการเรียกดูรหัสต้นฉบับ (source code) แต่ในมุมมองของเสิร์จเอ็นจินจะสามารถเห็นสิ่งที่สแปมเมอร์ซ่อนจากสายตาผู้ใช้

3.2 การพรางเอกสาร (cloaking) สแปมเมอร์จะสร้างเอกสารเว็บเพจขึ้นมาสองชุด ชุดแรกจะเป็นเอกสารเว็บเพจสำหรับผู้ใช้ธรรมดา ชุดที่สองจะเป็นเอกสารสำหรับเสิร์จเอ็นจินที่ได้ทำการสแปมเรียบร้อยไว้แล้ว สแปมเมอร์สามารถตรวจสอบการร้องขอของเอกสารว่ามาจากผู้ใช้หรือเสิร์จเอ็นจิน สามารถตรวจสอบได้จากยูเซอร์เอเจ้น (user-agent) ตัวอย่างข้อความการร้องขอเอกสารด้วยข้อตกลง (protocol) เอชทีทีพี (HTTP) บรรทัดที่สามเป็นการบ่งบอกยูเซอร์เอเจ้น ของผู้ใช้ธรรมดา

```
GET /db pages/members.html HTTP/1.0
Host: www-db.stanford.edu
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
```

ตัวอย่างยูเซอร์เอเจ้น ของเสิร์จเอ็นจินจากกูเกิล (Google)

```
GET /db pages/members.html HTTP/1.0
Host: www-db.stanford.edu
User-Agent: Googlebot/2.1 (+http://www.google.com/bot.html)
```

3.3 การเปลี่ยนหน้าเว็บเพจอัตโนมัติ (redirection) สแปมเมอร์จะสร้างเว็บเพจขึ้นมาอย่างน้อยสองเว็บเพจเมื่อผู้ใช้งานได้รับเอกสารที่มีการเปลี่ยนหน้าเว็บเพจ เว็บบราวเซอร์จะไปนำเอาหน้าเว็บเพจใหม่ตามคำสั่งเปลี่ยนหน้าเว็บเพจ ผู้ใช้ส่วนใหญ่จะไม่สามารถอ่านหน้าเว็บเพจเดิมได้ทัน เนื่องจากกระบวนการนี้เกิดขึ้นอย่างรวดเร็ว แต่จะส่งผลต่อเสิร์จเอ็นจินเพราะมีความสามารถในการอ่านเว็บเพจที่แตกต่างออกไปจากผู้ใช้ปกติ การเปลี่ยนหน้าเว็บเพจสามารถทำได้ที่เอกสาร เอชทีเอ็มแอล (HTML) ตัวอย่างเช่น

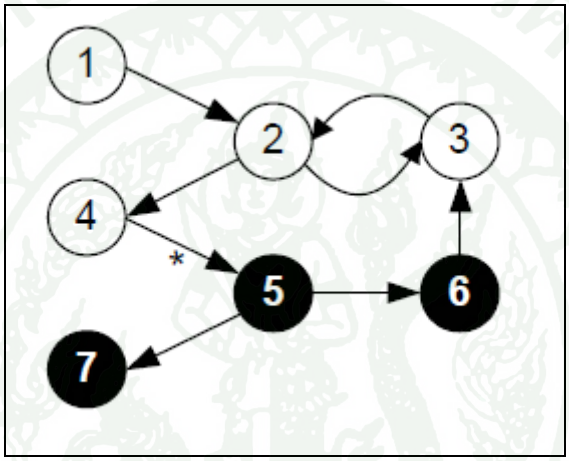
```
<meta http-equiv="refresh" content="0;url=target.html">
```

หรือจากภาษาจาวาสคริป

```
<script language="javascript"><!- -
        location.replace("target.html")- -></script>
```

ทรัสแร็งค์

เนื่องจากสแปมเมอร์ใช้จุดอ่อนของอัลกอริทึมเพจแร็งค์ จึงเกิดงานวิจัยที่ได้นำเสนออัลกอริทึมใหม่ในการจัดลำดับตามความน่าเชื่อถือคือทรัสแร็งค์ (TrustRank) อัลกอริทึมทรัสแร็งค์ (Gyöngyi *et al*., 2004) เกิดจากสมมติฐานที่ว่าไม่มีเว็บเพจที่ดีชี้ไปหาเว็บเพจสแปม แม้ว่าในความเป็นจริงจะมีเว็บเพจที่ดีชี้ไปหาเว็บเพจที่เป็นสแปมอยู่ แต่เว็บเพจเหล่านี้มีจำนวนน้อยมากเมื่อเทียบกับเว็บเพจทั้งหมด



ภาพที่ 7 ตัวอย่างเว็บกราฟใช้ในการคำนวณทรัสแร็งค์
วงกลมสีขาวแทนเว็บเพจปกติ วงกลมทีดำแทนเว็บเพจสแปม

การคัดแยกเว็บเพจสแปมกับเว็บเพจปกติออกจากกันมีวิธีที่เชื่อถือได้คือ ใช้ผู้เชี่ยวชาญทำการคัดแยกทีละเว็บเพจซึ่งจะเรียกว่าออเรเคิลฟังก์ชัน (oracle function) กำหนดให้เว็บเพจปกติมีค่าเป็นหนึ่งและเว็บเพจสแปมมีค่าเป็นศูนย์ สามารถเขียนเป็นสมการได้ดังนี้

$$O(p) = \begin{cases} 0 & \text{ถ้า } p \text{ เป็นเว็บเพจไม่สแปม} \\ 1 & \text{กรณีอื่นๆ} \end{cases} \quad (1)$$

โดยที่ p เป็นเว็บเพจที่นำมาตรวจสอบ ยกตัวอย่างเว็บกราฟในภาพที่ 7 เว็บเพจที่ 1 ถึง 4 เมื่อตรวจสอบโดยใช้ออเรเคิลฟังก์ชันจะมีค่าเป็นหนึ่ง ส่วนเว็บเพจที่ 5 ถึง 7 เมื่อตรวจสอบโดยใช้ออเรเคิลฟังก์ชันจะมีค่าเป็นศูนย์ อย่างไรก็ตามเนื่องจากเว็บเพจมีจำนวนมาก จึงไม่สามารถใช้ผู้เชี่ยวชาญ

มาตรวจสอบได้ทั้งหมดจึงจำเป็นต้องใช้วิธีการประมาณค่าด้วยค่าความน่าจะเป็น ซึ่งจะเรียกว่า คะแนนความน่าเชื่อถือ (trust score) เขียนเป็นสมการได้ดังนี้

$$T(p) = \Pr[O(p) = 1] \tag{2}$$

ตัวอย่าง สมมติมีเว็บเพจจำนวน 100 เว็บเพจแต่ละเว็บเพจมีคะแนนความน่าเชื่อถือเท่ากับ 0.7 สามารถสรุปได้ว่า มีอยู่ 70 เว็บเพจที่เมื่อตรวจสอบโดยใช้ออเรเคิลฟังก์ชันจะมีค่าเป็นหนึ่ง และมีอยู่ 30 เว็บเพจที่เมื่อตรวจสอบโดยใช้ออเรเคิลฟังก์ชันมีค่าเป็นศูนย์

สืบเนื่องจากปริมาณเว็บเพจมีจำนวนมากจึงสามารถตรวจสอบออเรเคิลฟังก์ชันได้เพียงบางเว็บเพจ และเรียกเว็บเพจเหล่านี้ว่าเว็บเพจต้นกำเนิด (seed) เขียนแทนด้วย S ให้เซตย่อย(subset)ของ S ที่มีแต่เว็บเพจปกติเป็น S^+ และเซตย่อยที่มีแต่เว็บเพจสแปมเป็น S^- เว็บเพจที่ไม่เป็นสมาชิกของเซตเว็บเพจต้นกำเนิดจะมีค่าคะแนนความน่าเชื่อถือเป็น 0 เขียนเป็นสมการได้ดังนี้

$$T_0 = \begin{cases} O(p) & \text{ถ้า } p \in S \\ 0 & \text{กรณีอื่นๆ} \end{cases} \tag{3}$$

โดยที่ T_0 เป็นค่าคะแนนความน่าเชื่อถือเริ่มต้น และให้ t_0 เป็นเวกเตอร์ค่าคะแนนความน่าเชื่อถือเริ่มต้นของทุกเว็บเพจ ซึ่งก่อนจะนำไปคำนวณต้องผ่านการนอร์มัลไลเซชัน (normalization) คือการหารทุกค่าใน t_0 ด้วยผลรวมของค่าใน t_0 ตัวอย่างจากภาพที่ 7 สมมุติให้ เซต $S = \{2,4,5\}$ สามารถเขียนค่าคะแนนความน่าเชื่อถือ ให้อยู่ในรูปของเวกเตอร์ t_0 ได้ดังนี้

$$t_0 = [0, \frac{1}{2}, 0, \frac{1}{2}, 0, 0, 0]$$

การคำนวณคะแนนจะอาศัยการส่งผ่านคะแนนทางลิงค์เชื่อมโยง ค่าทรัสแร็งค์ที่นำมาคำนวณจะอยู่ในรูปแบบของเมตริกซ์ (matrix) ดังนั้นเว็บกราฟจึงต้องอยู่ในรูปแบบของเมตริกซ์ด้วยเช่นกัน ซึ่งจะเรียกเมตริกซ์นี้ว่าเมตริกซ์ส่งผ่าน (transition matrix, L) มีนิยามดังนี้

$$L(p,q) = \begin{cases} 0 & \text{ถ้าไม่มีลิงค์ชี้จากเว็บเพจ } q \text{ ไปยังเว็บเพจ } p \\ 1/\omega(q) & \text{กรณีอื่นๆ} \end{cases} \tag{4}$$

เมื่อ $\omega(q)$ คือจำนวนลิงค์ที่ชี้ออกของเว็บเพจ q การคำนวณค่าคะแนนความน่าเชื่อถือจะคำนวณแบบเดียวกันกับอัลกอริทึมคำนวณค่าคะแนนเพจแร็งค์ คือการคำนวณเป็นแบบวนรอบจึงจำเป็นต้องมีการกำหนดค่ายับยั้ง (damping factor, α_β) และจำนวนรอบในการคำนวณเท่ากับ M_B สามารถคำนวณค่าความน่าเชื่อถือได้จากสมการต่อไปนี้

$$t_{i+1} = \alpha_\beta L t_i + (1-\alpha_\beta)t_0 \quad (5)$$

เมื่อ t_i เป็นค่าคะแนนความน่าเชื่อถือรอบที่ i เมื่อคำนวณคะแนนความน่าเชื่อถือถึงจำนวน M_B รอบแล้ว จำเป็นที่จะต้องตัดสินใจว่าเว็บเพจใดเป็นเว็บเพจปกติและเว็บเพจใดเป็นเว็บเพจสแปม โดยจะใช้ค่าที่น้อยที่สุดที่สามารถยอมรับได้ (threshold) ถ้าค่าคะแนนความน่าเชื่อถือของเว็บเพจใดมากกว่าค่านี้ก็แสดงว่าเว็บเพจนั้นเป็นเว็บเพจปกติ ถ้าน้อยกว่าจะเป็นเว็บเพจสแปม

การคัดแยกโดยอาศัยคุณลักษณะ

นักวิจัยจำนวนหนึ่งใช้เครื่องจักรเรียนรู้เพื่อคัดแยกเว็บเพจสแปมออกจากเว็บเพจปกติ แต่ก่อนที่เครื่องจักรเรียนรู้จะสามารถสร้างแบบจำลอง (model) ออกมาได้นั้นจำเป็นที่ต้องอาศัยคุณลักษณะ(feature, attribute)ของข้อมูล เพื่อให้เครื่องจักรทำการเรียนรู้ ในการสกัดคุณลักษณะจะใช้ข้อแตกต่างระหว่างเว็บเพจสแปมกับเว็บเพจปกติ ดังที่ได้นำเสนอไปข้างต้นคือเทคนิคที่สแปมเมอร์นำมาใช้คือการสร้างคำจำนวนมากไปในเว็บเพจ ทำให้เกิดงานวิจัย (Ntoulas *et al*., 2006) ได้นำเสนอการคัดแยกประเภทเว็บเพจสแปมกับเว็บเพจธรรมดาออกจากกัน โดยการสกัดคุณลักษณะจากจำนวนคำของแต่ละเว็บเพจ ซึ่งเว็บเพจเป็นภาษาอังกฤษ ในกลุ่มคุณลักษณะเนื้อหาประกอบไปด้วย รายละเอียดของคุณลักษณะมีดังนี้

1. กลุ่มคุณลักษณะเนื้อหา

1.1 คุณลักษณะจำนวนคำในเว็บเพจ เว็บเพจที่มีการสแปมโดยใช้วิธีด้านคำ แน่นอนว่าสแปมเมอร์ต้องใส่คำเพิ่มลงไปจนบางครั้งเว็บเพจมีคำเป็นจำนวนมาก จนกระทั่งเป็นการยากที่จะอ่านได้ทั้งหมดโดยผู้ใช้ทั่วไป ซึ่งถ้าเป็นเว็บเพจทั่วไปส่วนใหญ่ถ้ายาวเกินไปจะถูกจัดทำขึ้นเป็นอีกเว็บเพจหนึ่ง ทำให้เกิดข้อแตกต่างระหว่างจำนวนคำในเว็บเพจปกติกับจำนวนคำในเว็บเพจสแปม

1.2 คุณลักษณะจำนวนคำในแท็กหัวเรื่องของเอกสาร (tag title, <Title>…</Title>) หัวเรื่องของเอกสารเป็นส่วนที่เสิร์จเอ็นจินให้ความสำคัญกับคำในแท็กหัวเรื่องเป็นพิเศษ บางครั้งสแปมเมอร์จะใส่คำจำนวนมากในส่วนของหัวเรื่องเอกสาร ทำให้เกิดข้อแตกต่างด้านจำนวนคำในแท็กหัวเรื่องเว็บเพจระหว่างเว็บเพจปกติกับเว็บเพจสแปม

1.3 คุณลักษณะความยาวเฉลี่ยของคำ รวมไปถึงคำที่ประกอบมาจากคำอื่นๆ เช่น “freepicturedownload” ซึ่งบางเว็บเพจสแปมจะมีคำในลักษณะนี้ แต่ถ้าเป็นเว็บเพจปกติคำจะถูกคั่นด้วยการเว้นวรรค ทำให้เกิดข้อแตกต่างในด้านนี้

1.4 คุณลักษณะปริมาณแองเคอร์เท็กซ์ เนื่องจากแองเคอร์เท็กซ์เป็นส่วนสำคัญที่เสิร์จเอ็นจินให้ความสำคัญเป็นพิเศษ ซึ่งบางครั้งสแปมเมอร์จะสร้างแองเคอร์เท็กซ์มากมายในเว็บเพจเพื่อผลของค่าคะแนนเพจแร็งค์ ซึ่งจะเป็นข้อแตกต่างกับเว็บเพจปกติ

1.5 คุณลักษณะอัตราส่วนของคำที่มองเห็น บางครั้งถ้าผู้ใช้เข้าถึงเอกสารผ่านเว็บบราวเซอร์จะไม่สามารถมองเห็นบางส่วนของเว็บที่สแปมเมอร์ซ่อนเอาไว้ แต่ถ้าเป็นเว็บเพจปกติจะไม่นิยมซ่อนคำหรือถ้าซ่อนก็จะซ่อนไม่มากนัก การตรวจสอบคำที่มองไม่เห็นสามารถตรวจสอบได้จากแท็กที่ไม่แสดงผลโดยบราวเซอร์เช่น แท็กเมตาเป็นต้น

1.6 คุณลักษณะอัตราการบีบอัด สแปมเมอร์สร้างเอกสารส่วนใหญ่แล้วจะใช้โปรแกรมสร้างขึ้นมา ซึ่งโปรแกรมเหล่านี้จะอาศัยคำจากพจนานุกรมมาประกอบกันเป็นเอกสาร ซึ่งอัลกอริทึมการบีบอัดจะสามารถทำงานได้ดีก็ต่อเมื่อมีคำอยู่ในพจนานุกรม คุณลักษณะนี้จะคล้ายกับคุณลักษณะจำนวนคำ

ตัวแปรต่อไปนี้ใช้ในคุณลักษณะข้อ 1.7 ถึง 1.8

ให้ F เป็นเซตของคำส่วนใหญ่ที่เจอในฐานข้อมูล

ให้ Q เป็นเซตของคำสำคัญส่วนใหญ่ที่ใช้ในการค้นหา

ให้ P เป็นเซตของคำในเว็บเพจ

1.7 คุณลักษณะอัตราส่วนของคำที่เป็นที่นิยม เนื่องจากสแปมเมอร์จะสร้างเว็บเพจโดยสุ่มคำจากพจนานุกรม ทำให้แต่ละคำมีโอกาสสุ่มขึ้นมาเท่าๆ กันจากนั้นจึงนำคำเหล่านั้นมา

ประกอบเป็นเว็บเพจ โดยจะแตกต่างจากเว็บเพจปกติตรงที่ ถ้าเป็นเว็บเพจปกติแต่ละคำจะมีความถี่ในการถูกใช้ไม่เท่ากัน วิธีการสกัดคุณลักษณะนี้คือหาคำส่วนใหญ่ที่ถูกใช้บ่อยในฐานข้อมูล โดยที่ฐานข้อมูลคือเอกสารภาษาอังกฤษจำนวนหนึ่ง จากนั้นจึงนำเอาคำที่ถูกใช้บ่อยในฐานข้อมูลมาเปรียบเทียบกับเซตของคำในเว็บเพจ สามารถคำนวณค่าความแม่นยำและค่าเรียกคืนในฐานข้อมูลได้ดังนี้

$$CorpusPrecision = \frac{|P \cap F|}{|P|} \tag{6}$$

$$CorpusRecall = \frac{|P \cap F|}{|F|} \tag{7}$$

1.8 คุณลักษณะอัตราส่วนของคำสำคัญ คุณลักษณะนี้จะคล้ายกับคุณลักษณะก่อนข้อ 1.7 แต่จะเปลี่ยนจากคำที่ถูกใช้บ่อยในฐานข้อมูลเป็นคำที่ถูกใช้บ่อยในการค้นหา ซึ่งสแปมเมอร์จะใส่คำพวกนี้ไปในเว็บเพจจำนวนมากกว่าเว็บเพจปกติ สามารถคำนวณค่าความแม่นยำและค่าเรียกคืนของคำสำคัญได้ดังนี้

$$QueryPrecision = \frac{|P \cap Q|}{|P|} \tag{8}$$

$$QueryRecall = \frac{|P \cap Q|}{|Q|} \tag{9}$$

1.9 คุณลักษณะจำนวนรูปแบบของคำ จำนวนคำในรูปแบบจะเรียกว่าแกรม (gram) ซึ่งจะพิจารณาแค่สามแกรมหรือไตรแกรม (tri-grams) เท่านั้นตัวอย่างเช่นคำว่า “mike lab ku” จะนับว่าในเอกสารนี้มีคำว่า mike ตามด้วย lab ตามด้วย ku กี่ครั้งมาเป็นคุณลักษณะ เนื่องจากสแปมเมอร์จะสร้างเว็บเพจโดยสุ่มคำจากพจนานุกรม ทำให้โอกาสที่จะเกิดไตรแกรมนั้นจะน้อย

1.10 คุณลักษณะเอ็นโทรปี (Entropy) ให้ T เป็นเซตไตรแกรมในเว็บเพจ แต่ละสมาชิกในเซตประกอบไปด้วยไตรแกรม w_i และความถี่ p_i ของไตรแกรมนั้น เราสามารถแสดงเซต T ได้ดังนี้

$$T = \{(w_1, p_1), (w_2, p_2), ..., (w_k, p_k)\} \tag{10}$$

และสามารถหาค่าคุณลักษณะเอ็นโทรปีได้จากสมการ

$$entropy = -\sum_{w_i \in T} p_i \log(p_i) \quad (11)$$

จากการทดลองให้ผลลัพธ์เป็นที่น่าพอใจในระดับหนึ่งแต่ยังไม่สามารถคัดแยกเว็บเพจสแปมที่เกิดจากลิงค์ได้ จึงได้เกิดงานวิจัยอีกงานขึ้นมาได้ทำการสกัดคุณลักษณะของลิงค์ (Becchetti et al., 2008) ซึ่งทำในระดับของโฮสต์สามารถแบ่งได้ตามวิธีการคำนวณออกได้ดังนี้

2. กลุ่มคุณลักษณะลิงค์

2.1 กลุ่มคุณลักษณะจำนวนลิงค์ พิจารณาจากสองเว็บเพจในโฮสต์คือโฮมเพจ (home page) และเว็บเพจที่มีค่าคะแนนเพจแร็งค์สูงสุด จากนั้นจึงนับจำนวนลิงค์ชี้เข้าและลิงค์ชี้ออกของโฮมเพจหรือเว็บเพจที่มีค่าคะแนนเพจแร็งค์สูงสุด ออกมาสร้างเป็นคุณลักษณะเป็นต้น

2.2 กลุ่มคุณลักษณะเพจแร็งค์ นำค่าเพจแร็งค์ของโฮมเพจและเว็บเพจที่มีค่าคะแนนเพจแร็งค์สูงสุดมาเป็นสกัดคุณลักษณะ แต่ค่าเพจแร็งค์เพียงอย่างเดียวยังไม่มีประสิทธิภาพพอที่จะแบ่งแยกระหว่างโฮสต์สแปมกับโฮสต์ปกติได้ จึงนำค่าเพจแร็งค์มาคำนวณร่วมกับจำนวนลิงค์เช่น อัตราส่วนระหว่างค่าเพจแร็งค์หารด้วยจำนวนลิงค์ชี้ที่เข้าเป็นต้น

2.3 กลุ่มคุณลักษณะทรัสแร็งค์ เลือกค่าทรัสแร็งค์จากโฮมเพจหรือเว็บเพจที่มีค่าคะแนนเพจแร็งค์สูงสุดมาเป็นคุณลักษณะ ตัวอย่างคุณลักษณะเช่น ค่าทรัสแร็งค์หารด้วยค่าเพจแร็งค์

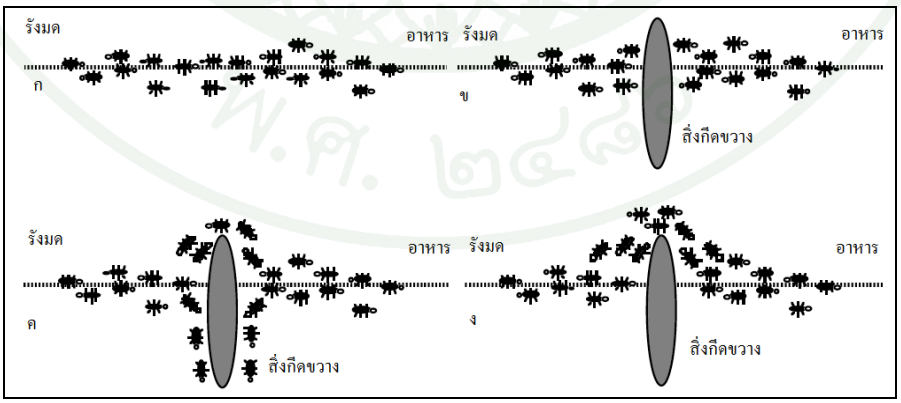
2.4 กลุ่มคุณลักษณะทรังเคทเพจแร็งค์ (truncated PageRank) เป็นการนำเอางานวิจัยของ (Becchetti *et al*., 2006b) ที่เสนอวิธีแก้ปัญหาจุดอ่อนของการคำนวณเพจแร็งค์มาเป็นคุณลักษณะ

2.5 กลุ่มคุณลักษณะเว็บสนับสนุน เนื่องจากเป็นการยากที่สร้างเว็บเพจเพียงเว็บเดียวเพื่อที่จะได้ลำดับที่ดีจากการจัดลำดับของเสิร์จเอ็นจิน จึงจำเป็นต้องดูเว็บเพื่อนบ้านรอบข้างด้วย ซึ่งจำนวนเว็บเพื่อนบ้านสามารถนำมาสกัดเป็นคุณลักษณะได้

เมื่อสแปมสามารถเกิดทั้งจากเนื้อหาและจากลิงค์จึงมีงานวิจัยซึ่งนำเอาคุณลักษณะทั้งสองมารวมกัน (Castillo *et al*., 2007) สกัดคุณลักษณะในระดับของโฮสต์แต่คุณลักษณะจำพวกจำนวนคำจากงานวิจัยข้างต้นได้สกัดในระดับเว็บเพจจึงต้องมีการแปลงให้อยู่ในระดับของโฮสต์ก่อนเพื่อที่จะใช้ได้ แต่กลับพบว่าประสิทธิภาพในการคัดแยกลดลง จึงนำเสนอวิธีการปรับผลลัพธ์จากการคัดแยกให้มีความถูกต้องมากยิ่งขึ้น โดยที่หากมองว่าเว็บอยู่ในรูปแบบปัญหาของกราฟมีโฮสต์เป็นโหนดมีลิงค์เป็นเส้นเชื่อม โดยมีสมมติฐานที่ว่าเว็บโฮสต์ที่มีประเภทเหมือนกันจะอยู่เป็นกลุ่มกันโดยนำเอาอัลกอริทึมต่างๆ ที่มีอยู่เช่น METIS มาใช้ในจัดกลุ่มโฮสต์กราฟ การปรับผลลัพธ์ที่ได้จากการคัดแยกด้วยอัลกอริทึม C4.5 ในกลุ่มเดียวกันให้มีคลาสเหมือนกันโดยคลาสเป็นไปตามคลาสที่มีจำนวนมากกว่าในกลุ่ม พบว่าผลลัพธ์ที่ได้สามารถคัดแยกโฮสต์สแปมได้ดียิ่งขึ้น

แอนท์โคโลนีออพติไมเซชัน

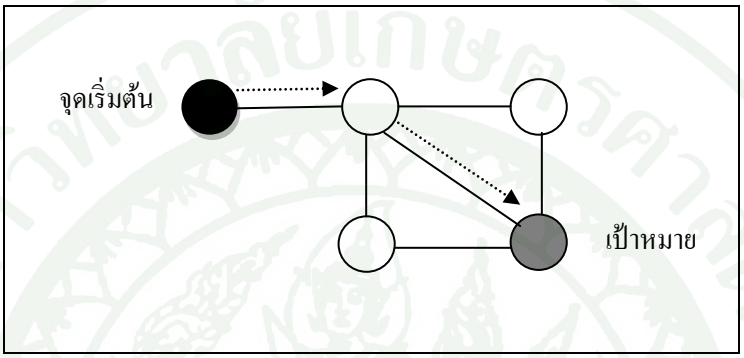
อัลกอริทึมแอนท์โคโลนีออพติไมเซชัน (Dorigo and Gambardella, 1997) หนึ่งในอัลกอริทึมกลุ่มของสวอร์มอินเทลลิเจนซ์ (swarm intelligent) เป็นอัลกอริทึมเลียนแบบพฤติกรรมในการหาอาหารของมดตามธรรมชาติ ตัวอย่างดังภาพที่ 8 (ก) กลุ่มมดเดินจากรังไปหาอาหารเพื่อในกลับมายังรัง (ข) เมื่อเพิ่มสิ่งกีดขวางทางเดินของกลุ่มมด (ค) เมื่อกลุ่มมดพบสิ่งกีดขวางกลุ่มมดจะหาทางเดินทางอื่นโดยใช้การสุ่ม (ง) เมื่อเวลาผ่านไปซักพักกลุ่มมดจะสามารถหาเส้นทางที่สั้นที่สุดได้ ซึ่งกลุ่มมดจะอาศัยสารเคมีที่ทิ้งไว้ที่เรียกว่าฟีโรโมน (pheromone) ในการหาเส้นทาง นักวิจัยจึงอาศัยพฤติกรรมเหล่านี้มาสร้างเป็นอัลกอริทึม



ภาพที่ 8 การหาอาหารของมดในธรรมชาติ

ที่มา: Dorigo and Gambardella (1997b)

ซึ่งอัลกอริทึมจะสร้างมดเทียม (artificial ant) ขึ้นมาเพื่อเลียนแบบการหาเส้นทาง โดยที่เส้นทางที่มดเทียมจะสามารถเดินได้นั้นต้องอยู่ในรูปแบบของกราฟ มดจะต้องเดินตามเส้นเชื่อมโยงระหว่างโหนดเท่านั้น โดยที่มดจะหาเส้นทางจากจุดเริ่มต้นไปยังเป้าหมาย ตัวอย่างดังภาพที่ 9 แสดงตัวอย่างกราฟที่มดเทียมสามารถเดินได้ วงกลมแทนโหนด เส้นทึบแทนเส้นเชื่อมโยง เส้นประแสดงเส้นทางที่สั้นที่สุดจากจุดเริ่มต้นไปยังเป้าหมาย (ตัวอย่างนี้ให้แต่ละเส้นเชื่อมโยงมีความยาวเท่ากัน)



ภาพที่ 9 ตัวอย่างกราฟที่มดเทียมใช้เดิน

ในขั้นตอนการหาเส้นทาง วิธีที่มดเทียมใช้ตัดสินใจเดินไปยังโหนดอื่นคือการสุ่มด้วยค่าความน่าจะเป็น ซึ่งแต่ละเส้นทางจะมีค่าความน่าจะเป็นไม่เท่ากัน เนื่องจากวิธีการตัดสินใจใช้การสุ่มจึงจำเป็นต้องมีมดเทียมหลายตัวช่วยในการหาเส้นทางเพื่อลดผลกระทบที่เกิดจากการสุ่มลง และเดินด้วยจำนวนหลายรอบ ค่าความน่าจะเป็นที่มดเทียมใช้นั้นจะประกอบไปด้วยค่าฮิวริสติก (heuristic) และค่าฟีโรโมน โดยมีนิยามดังนี้ ถ้าให้โหนดที่มดกำลังอยู่คือโหนด i และโหนดที่สามารถเดินต่อไปได้คือโหนด j ค่าความน่าจะเป็นที่มดเทียมจะเดินจากโหนด i ไปโหนด j คือ

$$P_{ij} = \frac{N_{ij}T_{ij}(t)}{\sum N_{ij}T_{ij}(t)} \tag{12}$$

N_{ij} คือค่าฮิวริสติก ฮิวริสติกเป็นตัวช่วยในการค้นหาเส้นทาง การกำหนดค่าฮิวริสติกจะขึ้นอยู่กับปัญหา แต่สำหรับปัญหาการหาเส้นทางนิยมที่จะใช้คือส่วนกลับของความยาวเส้นเชื่อมโยง ซึ่งหมายถึงมดเทียมจะมีโอกาสเลือกเส้นเชื่อมโยงระหว่างโหนดที่สั้นมากกว่าเส้นเชื่อมโยงที่ยาว ซึ่งค่านี้จะไม่เปลี่ยนแปลงตามจำนวนรอบการเดิน t

$T_{ij}(t)$ คือค่าฟีโรโมน ค่านี้มีขึ้นเพื่อเลียนแบบร่องรอยที่มดตามธรรมชาติทิ้งไว้ เมื่อมดเทียมเดินเสร็จแต่ละรอบ t (หนึ่งรอบคือช่วงที่มดทุกตัวเริ่มเดินจนถึงมดทุกตัวสิ้นสุดการเดิน) ค่าฟีโรโมน

ในแต่ละเส้นเชื่อมโยงจะถูกปรับค่า ซึ่งการปรับค่าฟีโรโมนสำหรับการหาเส้นทางนั้น จะปรับค่าให้กับเส้นทางที่ได้จากมดเทียมตามระยะทางจากจุดเริ่มต้นไปยังเป้าหมาย โดยจะปรับค่าฟีโรโมนให้กับเส้นทางที่สั้นมากกว่าเส้นทางที่ยาว

รายละเอียดอื่นๆ ในแต่ละรอบมดเทียมแต่ละตัวจะไม่เดินซ้ำไปยังโหนดที่ได้เคยผ่านมาก่อนหน้านี้ การทำงานของอัลกอริทึมจะสิ้นสุดลงเมื่อมดเทียมทำงานครบตามจำนวนรอบที่ได้กำหนดไว้ นอกจากนั้นการคำนวณค่าฮิวริสติกและค่าฟีโรโมนจะอยู่กับลักษณะของปัญหา ซึ่งสามารถปรับเปลี่ยนตามลักษณะของปัญหาได้

อุปกรณ์และวิธีการ

อุปกรณ์

1. อุปกรณ์คอมพิวเตอร์ (hardware)
   1.1 หน่วยประมวลผลกลาง (CPU) Intel Xeon E5620 2.4 GHz
   1.2 หน่วยความจำ (RAM) 16 GB
   1.3 ฮาร์ดดิสก์ (hard disk)

2. ซอฟต์แวร์ (software)
   2.1 Weka 3.7
   2.2 JDK 1.6 (complier)
   2.3 Eclipse (editor)
   2.4 ระบบปฏิบัติการ Linux
   2.5 ระบบปฏิบัติการ Windows 7

วิธีการ

ในส่วนนี้จะบรรยายถึงขั้นตอนของวิธีการซึ่งอาศัยแนวคิดและสมมติฐานบางส่วนของงานวิจัยที่ผ่านมาในการประยุกต์ใช้เพื่อสร้างเป็นวิธีการคัดแยกโฮสต์สแปมและโฮสต์ปกติออกจากกัน ซึ่งผลลัพธ์ที่ได้จะอยู่ในรูปแบบของกฎกล่าวคือประกอบด้วยเงื่อนไขและคำตอบหรือคลาส (class) โดยที่ผลการทำนายของกฎจะมีอยู่สองประเภทคือโฮสต์สแปมและโฮสต์ปกติ ในการคัดแยกประเภท (classification) ข้อมูลโดยทั่วไปแล้วสามารถแบ่งออกเป็น 2 กลุ่มคือการเรียนรู้แบบมีผู้สอน (supervised learning) ซึ่งการเรียนรู้แบบนี้ข้อมูลที่ใช้สอน (training data) จะต้องมีคลาสด้วยเสมอ และอีกกลุ่มการเรียนรู้แบบไม่มีผู้สอน (unsupervised learning) ซึ่งในงานวิจัยนี้จะอยู่ในกลุ่มการเรียนรู้แบบมีผู้สอน นอกจากนี้จะยังใช้ประโยชน์จากโฮสต์เพื่อนบ้านและโฮสต์ที่ไม่มีคำตอบหรือคลาสเพื่อสร้างเป็นกฎอีกด้วย กฎจะถูกสร้างจากเส้นทางโดยอัลกอริทึมแอนท์โคโลนีออพติไมเซชัน สามารถแบ่งออกได้เป็นสามระยะได้แก่ระยะเริ่มต้น ระยะปรับตัว ระยะบรรจบ

ภาพรวมการทำงาน

อัลกอริทึมแอนท์โคโลนีออพติไมเซชันจะสร้างมดเทียม (artificial ant) ขึ้นมา มดเทียมเหล่านี้จะให้เส้นทาง ซึ่งเส้นทางจะประกอบไปด้วยโฮสต์ต่าง โดยที่งานวิจัยนี้ต้องการโฮสต์บนเส้นทางที่มีคลาสเดียวกัน ซึ่งวิธีการหาเส้นทางจะกล่าวในลำดับถัดไป จากนั้นจึงนำเส้นทางเหล่านี้มาสร้างเป็นกฎเพื่อบอกความแตกต่างระหว่างโฮสต์สแปมและโฮสต์ปกติ

ภาพที่ 10 ภาพรวมการทำงานของการเรียนรู้

สร้างแบบจำลองโฮสต์กราฟ

ในการเตรียมพร้อมให้อัลกอริทึมแอนท์โคโลนีออพติไมเซชันทำงานได้ จำเป็นที่ต้องสร้างมุมมองของปัญหาให้อยู่ในรูปแบบของกราฟ ซึ่งโฮสต์และลิงค์การเชื่อมโยงระหว่างโฮสต์สามารถจัดให้อยู่ในรูปแบบของกราฟได้ดังนี้คือ ให้ $G_h = (V, E)$ เป็นโฮสต์กราฟเมื่อ V และ E แสดงถึงเซตของโฮสต์และลิงค์เชื่อมโยงระหว่างโฮสต์ตามลำดับ ระหว่างโฮสต์จะมีลิงค์เชื่อมโยง $e(h_i, h_j) \in E$ ถ้ามีเว็บเพจ u เป็นสมาชิกของโฮสต์ $h_i \in V$ มีลิงค์เชื่อมโยงไปยังเว็บเพจ v ซึ่งเป็นสมาชิกของโฮสต์ $h_j \in V$ และ $i \neq j$ เนื่องจากโฮสต์กราฟเป็นการมองจากกลุ่มของเว็บเพจซึ่งต่างจากเว็บกราฟทำให้อาจจะมีจำนวนลิงค์มากมายดังนั้นให้ $|e(h_i, h_j)|$ เป็นจำนวนลิงค์ออกจากโฮสต์ h_i ไปยังโฮสต์ h_j แต่จะถ้าเป็นลิงค์จากเว็บเพจภายในโฮสต์เดียวกัน จะถูกตัดออก $|e(h_i, h_j)| = 0$ เมื่อ $i = j$

การเรียนรู้ด้วยอัลกอริทึมแอนท์โคโลนีออพติไมเซชัน

หลักจากได้รับโฮสต์กราฟจากนั้น เราจะให้มดเทียมเริ่มต้นเดินทางโฮสต์ h_i ที่มีคลาสมดเทียมจะทำการเลือกเส้นทาง $e(h_i, h_j)$ เพื่อเดินทางไปยังโฮสต์ถัดไปจากโฮสต์ h_i ไปยังโฮสต์ h_j โดยใช้การสุ่ม (randomization) ด้วยค่าความน่าจะเป็น (probability) ซึ่งคำนวณมากจากค่าของฮิวริสติก (heuristic value) และค่าของฟีโรโมน (pheromone value) ถ้าเราให้ P_{ij} เป็นค่าความน่าจะเป็นที่

มดเทียมจะเลือกเส้นทางจากลิงค์ $e(h_i, h_j)$ เราสามารถเขียนสมการความน่าจะเป็นที่มดสุ่มเลือกเส้นทางจากโฮสต์ h_i ไปยังโฮสต์ h_j ดังนี้

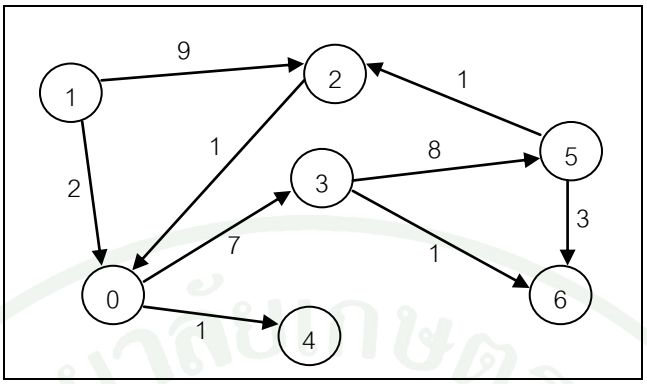
$$P_{ij} = \begin{cases} x_j \cdot \dfrac{\eta_{ij}\tau_{ij}(t)}{\sum_{h_k \in F(h_i)} \eta_{ik}\tau_{ik}(t)} & \text{ถ้า } e(h_i, h_j) \in E, \\ 0 & \text{กรณีอื่นๆ} \end{cases} \tag{13}$$

เมื่อ η_{ij} และ $\tau_{ij}(t)$ เป็นฮิวริสติกฟังก์ชันและฟีโรโมนฟังก์ชันในรอบเวลาที่ t ถ้าพิจารณาจากสมการข้างต้นจะพบว่ามีตัวแปร x_j ประกอบอยู่ด้วย ซึ่งตัวแปรนี้ช่วยป้องกันมดเทียมในการเดินซ้ำมายังโฮสต์ที่เคยเดินผ่าน โดยให้ค่าความน่าจะเป็นในเส้นทางที่เคยเดินผ่านเป็น 0 สามารถสร้างเป็นเงื่อนไขได้ดังนี้คือ ให้ $F(h_i)$ เป็นเซตของโฮสต์ที่มีลิงค์ชี้มาจากโฮสต์ h_i และ Γ เป็นเซตของโฮสต์ที่มดเทียมเคยเดินผ่าน โดยที่มดเทียมจะมีหน่วยความจำแยกออกจากกัน เราสามารถอธิบายตัวแปร x_j ได้ดังนี้

$$x_j = \begin{cases} 1 & \text{ถ้า } h_j \notin \Gamma, \\ 0 & \text{กรณีอื่นๆ} \end{cases} \tag{14}$$

ในงานวิจัยนี้ได้นำแนวคิดของทรัสแร็งค์ (Gyöngyi *et al*., 2004) คือเว็บเพจปกติส่วนใหญ่จะมีลิงค์ชี้ไปยังเว็บเพจปกติ โดยงานวิจัยนี้ได้นำแนวคิดนี้มาใช้ในระดับโฮสต์เสนอสมมติฐานที่พบจากการสังเกตจากฐานข้อมูล WEB Spam UK 2006 จำนวนลิงค์ที่ชี้จากโฮสต์ปกติไปยังโฮสต์ปกติ (ในฐานข้อมูล WEB Spam UK 2006 มีจำนวนทั้งสิ้น 181401 ลิงค์) จะมากกว่าจำนวนลิงค์ที่ชี้จากโฮสต์ปกติไปยังโฮสต์สแปม (ในฐานข้อมูล WEB Spam UK 2006 มีจำนวนทั้งสิ้น 3792 ลิงค์) จากสมมติฐานข้างต้นจำนวนลิงค์ที่ชี้ออกจากโฮสต์ h_i สามารถนำมาใช้เป็นฟังก์ชันฮิวริสติก เพื่อให้มดเทียมสามารถหาเส้นทางไปยังโฮสต์อื่น h_j จำนวนลิงค์ที่เชื่อมระหว่างสองโฮสต์แปรผันกับค่าฮิวริสติกดังนี้ ถ้าให้ h_i เป็นโฮสต์ปกติแล้ว มีลิงค์ที่ชี้ออกจากโฮสต์ปกติ h_i ไปยังโฮสต์ h_j มากแสดงว่าโฮสต์ h_j มีความน่าจะเป็นที่จะเป็นโฮสต์ปกติมากขึ้นตามจำนวนลิงค์ เราสามารถเขียนสมการฮิวริสติกฟังก์ชันเมื่อจุดเริ่มต้นของมดเทียมมีคลาสเป็นโฮสต์ปกติดังนี้

$$\eta_{ij} = \frac{|e(h_i, h_j)|}{\sum_{h_k \in F(h_i)} |e(h_i, h_k)|} \tag{15}$$



ภาพที่ 11 ตัวอย่างโฮสต์กราฟที่ใช้ในตัวอย่างการคำนวณค่าฮิวริสติก

จากโฮสต์กราฟดังภาพที่ 11 จะประกอบไปด้วยสัญลักษณ์ต่างๆ สามารถอธิบายได้ดังนี้

1. วงกลมแทนโฮสต์ ในวงกลมประกอบไปด้วยตัวเลข ซึ่งแต่ละตัวเลขจะบ่งบอกถึงหมายเลขกำกับ (identification) ประจำโฮสต์

2. ลูกศรแทนลิงค์เชื่อมโยงระหว่างโฮสต์ โดยที่โฮสต์ทางด้านหัวลูกศรจะเป็นโฮสต์ปลายทาง (destination) และโฮสต์ทางด้านหางลูกศรจะเป็นโฮสต์ต้นทาง (source) แต่ละลิงค์จะมีตัวเลขกำกับ ตัวเลขเหล่านี้แสดงถึงจำนวนเส้นเชื่อมระหว่างโฮสต์ต้นทางกับโฮสต์ปลายทาง

ตัวอย่างการคำนวณค่าฮิวริสติกเมื่อจุดเริ่มต้นเป็นโฮสต์ปกติจากดังอย่างภาพที่ 11 โดยแสดงตัวอย่างของโฮสต์หมายเลขกำกับ 0 ดังนี้ (ให้โฮสต์หมายเลข 0 เป็นโฮสต์ปกติ)

$$\eta_{0,3} = \frac{|e(h_0,h_3)|}{\sum_{h_k \in F(h_0)} |e(h_0,h_k)|} = \frac{7}{7+1}$$

$$\eta_{0,4} = \frac{1}{7+1}$$

อีกสมมติฐานหนึ่งที่ได้จากการแนวคิดของแอนติทรัสแร็งค์ (Krishnan and Rashmi, 2006) ที่ว่าเว็บเพจที่ชี้ไปยังเว็บเพจสแปมจะเป็นเว็บเพจสแปมด้วย งานวิจัยนี้จึงนำแนวคิดนี้มาใช้ในระดับของโฮสต์ โดยสังเกตจากฐานข้อมูล WEB Spam UK 2006 จำนวนลิงค์ที่ชี้เข้าหาโฮสต์สแปมจากโฮสต์สแปม (ในฐานข้อมูล WEB Spam UK 2006 มีจำนวนทั้งสิ้น 28378 ลิงค์) จะมากกว่าจำนวน

ลิงค์ที่เข้าหาโฮสต์สแปมจากโฮสต์ปกติ (ในฐานข้อมูล WEB Spam UK 2006 มีจำนวนทั้งสิ้น 3792 ลิงค์) จากสมมติฐานที่ได้กล่าวมาถ้าให้ h_j เป็นโฮสต์สแปมแล้ว h_i มีลิงค์ชี้ไปยังโฮสต์ h_j จำนวนมากแล้วโฮสต์ h_i ก็มีความน่าจะเป็นที่เป็นโฮสต์สแปมด้วยเช่นกัน ถ้าให้ $B(h_i)$ เป็นเซตของโฮสต์ที่มีลิงค์ชี้ไปยังโฮสต์ h_i สามารถเขียนสมการฮิวริสติกฟังก์ชันเมื่อจุดเริ่มต้นของมดเทียมมีคลาสเป็นโฮสต์สแปมดังนี้

$$\eta_{ij} = \frac{|e(h_j, h_i)|}{\sum_{h_k \in B(h_i)} |e(h_k, h_i)|} \tag{16}$$

จะสังเกตเห็นว่าลักษณะฟังก์ชันจะคล้ายกับสมการที่ (15) ต่างกันที่ทิศทางลิงค์ที่ใช้ในการคำนวณ และเนื่องทิศทางการเดินของมดเทียมที่มีจุดเริ่มต้นเป็นโฮสต์สแปมนั้นจะมีทิศทางที่แตกต่างกับจุดเริ่มต้นเป็นโฮสต์ปกติ ทำให้จำเป็นต้องปรับสมการความน่าจะเป็นที่มดเทียมสุ่มเลือกเส้นทาง (ปรับจากสมการที่ (13)) มาเป็นสมการดังต่อไปนี้

$$P_{ij} = \begin{cases} x_j \cdot \dfrac{\eta_{ij}\tau_{ij}(t)}{\sum_{h_k \in B(h_i)} \eta_{ik}\tau_{ik}(t)} & \text{ถ้า } e(h_i, h_j) \in E, \\ 0 & \text{กรณีอื่นๆ} \end{cases} \tag{17}$$

เนื่องจากค่าฮิวริสติกเป็นค่าคงที่จึงสามารถคำนวณล่วงหน้าได้ ซึ่งช่วยลดเวลาในการคำนวณก่อนจะดำเนินการตามอัลกอริทึมแอนท์โคโลนีออพติไมเซชัน

ตัวอย่างการคำนวณค่าฮิวริสติกเมื่อจุดเริ่มต้นเป็นโฮสต์สแปมจากดังอย่างภาพที่ 11 โดยแสดงตัวอย่างของโฮสต์หมายเลขกำกับ 2 ดังนี้ (ให้โฮสต์หมายเลข 2 เป็นโฮสต์สแปม)

$$\eta_{1,2} = \frac{|e(h_1, h_2)|}{\sum_{h_k \in B(h_0)} |e(h_k, h_2)|} = \frac{9}{9+1},$$

$$\eta_{5,2} = \frac{1}{9+1}$$

ค่าฮิวริสติกฟังก์ชันเป็นเพียงส่วนประกอบที่หนึ่งของอัลกอริทึมแอนท์โคโลนีออพติไมเซชัน ส่วนประกอบที่สอง ที่ทำให้อัลกอริทึมสามารถทำการหาคำตอบที่เหมาะสม (optimization) ได้

คือฟังก์ชันฟรีโรโมนซึ่งเป็นส่วนประกอบของสมการที่ (13) และสมการที่ (17) ฟังก์ชันนี้เป็นเป็นตัวควบคุมการเปลี่ยนแปลงของการทำงานในแต่ละรอบ ให้รอบการคำนวณ $t = 1,2,3,...$ เราสามารถคำนวณค่าของฟรีโรโมนได้ดังต่อไปนี้

$$\tau_{ij}(t) = \tau_{ij}(t-1) + \tau_{ij}(t-1)q \tag{18}$$

เมื่อ q เป็นตัววัดคุณภาพของกฎซึ่งขั้นตอนการสร้างกฎ รายละเอียดเพิ่มเติมจะกล่าวไว้ในหัวข้อการสร้างกฎที่จะกล่าวถึงในภายหลัง โดยค่า q จะคำนวณจากข้อมูลชุดฝึกสอน กฎที่ได้จะอยู่ในรูปแบบของกฎเกณฑ์ คือมีเงื่อนไขกับผลที่ตามมา ซึ่งผลที่ตามมาจะถูกนำมาใช้ทำนายโฮสต์ว่าเป็นโฮสต์สแปมหรือโฮสต์ปกติ รูปแบบของกฎจะเป็นไปดังนี้

ถ้า (If) <เงื่อนไข (condition)> แล้ว (Then) <ผลการทำนาย (prediction)>

ในการนำกฎมาใช้ จะทำนายเฉพาะโฮสต์ที่ถูกครอบคลุมโดยกฎเท่านั้น ซึ่งการครอบคลุมจะพิจารณาจากเงื่อนไขของกฎกับค่าคุณลักษณะของโฮสต์ ถ้าค่าคุณลักษณะตรงตามเงื่อนไขของกฎจะเรียกโฮสต์นั้นว่าถูกครอบคลุมโดยกฎ ถ้าค่าคุณลักษณะโฮสต์ไม่ตรงตามเงื่อนไขของกฎจะเรียกโฮสต์นั้นว่าไม่ถูกครอบคลุมโดยกฎ

ตัววัดคุณภาพในงานวิจัยนี้จะมีพื้นฐานมากจากค่าความเชื่อมั่น (confident) สามารถคำนวณได้ดังต่อไปนี้

$$confident = \frac{|CH \cap EH|}{|CH|} \tag{19}$$

โดยที่ CH เป็นเซตของโฮสต์ที่ครอบคลุมโดยกฎและ EH เป็นเซตของโฮสต์ที่มีคลาสเหมือนกับคลาสของกฎ นอกจากนั้นตัววัดคุณภาพของกฎในงานวิจัยนี้จะมีพื้นฐานมาจากค่าซับพอร์ต (support) สามารถคำนวณได้ดังต่อไปนี้

$$support = \frac{|CH|}{|training\_data|} \tag{20}$$

โดยที่ $|training\_data|$ เป็นจำนวนของข้อมูลที่มีคลาส ค่าคุณภาพของกฎสามารถคำนวณได้ดังนี้

$$q = \begin{cases} confident & \text{ถ้า } support > ts\_sup \text{ และ } confident > ts\_con \\ 0 & \text{กรณีอื่นๆ} \end{cases} \quad (21)$$

เมื่อ ts_sup คือค่าต่ำสุดของจำนวนโฮสต์ที่ครอบคลุมโดยกฎที่สามารถยอมรับได้และ ts_con คือค่าต่ำสุดของค่าความเชื่อมั่นจากกฎที่สามารถยอมรับได้ ซึ่งก็คือจำนวนโฮสต์ที่ครอบคลุมโดยกฎและทำนายถูกโดยกฎหารด้วยจำนวนโฮสต์ทั้งหมดที่ครอบคลุมโดยกฎ ถ้ากฎใดมีค่า $q = 0$ (ไม่ผ่านเงื่อนไขข้างต้น) จะถูกกำจัดออกไปและมดเทียมตัวที่ให้ค่า $q = 0$ จะต้องถูกทำลาย การปรับค่าฟีโรโมนจะปรับที่เส้นทาง โดยจะเลือกจากมดเทียมตัวที่ให้ค่า q สูงที่สุดมาปรับค่าฟีโรโมน แต่ถ้ามีมดเทียมหลายตัวที่ให้ค่า q สูงสุดเท่ากัน งานวิจัยนี้จะนำมดเทียมที่ให้ค่า q สูงสุดเหล่านั้นมาปรับค่าฟีโรโมนทั้งหมด เมื่อทำการคำนวณครบจำนวนรอบแล้วจะนำกฎที่ได้กฎที่มีค่า q สูงสุดของแต่ละรอบมาพิจารณาอีกครั้งโดยเลือกกฎจากรอบที่ดีมีค่า q สูงสุดมาเก็บไว้ในชุดของกฎ จากนั้นจึงทำการเลือกจุดเริ่มต้นขึ้นมาใหม่เพื่อทำการหากฎต่อไป

สำหรับรอบการคำนวณแรก จำเป็นต้องกำหนดค่าของฟีโรโมนเริ่มต้น คำนวณได้จากจำนวนลิงค์ในกราฟเท่ากันหมดดังนี้

$$\tau_{ij}(t=0) = \frac{1}{|E|} \quad (22)$$

ซึ่งค่าฟีโรโมนของแต่ละโฮสต์จะแยกออกจากกัน เมื่อทำการเปลี่ยนจุดเริ่มต้นในการสร้างกฎ ค่าฟีโรโมนจะถูกตั้งค่าให้เป็นค่าเริ่มต้นใหม่

ขั้นตอนการสร้างเส้นทางของมดเทียม

โฮสต์จากข้อมูลชุดฝึกสอน จะประกอบไปด้วยข้อมูลที่มีคลาสกับข้อมูลที่ไม่มีคลาส เราจะนำข้อมูลที่มีคลาสมากำหนดเป็นจุดเริ่มต้น โดยจุดเริ่มต้นจะถูกแบ่งออกเป็นสองกลุ่มได้แก่กลุ่มโฮสต์ปกติและกลุ่มโฮสต์สแปม โดยแต่ละกลุ่มจะมีขั้นตอนการเดินดังนี้

1. คำนวณค่าฮิวริสติก ค่าฟีโรโมนเริ่มต้น

2. เลือกโฮสต์ภายในกลุ่มขึ้นมา 1 โฮสต์เพื่อกำหนดเป็นจุดเริ่มต้น

3. กลุ่มมดเทียมเริ่มเดินจากโฮสต์ที่ถูกเลือก โดยจำกัดจำนวนก้าว (hop) ไว้ การตัดสินใจเลือกเส้นทางจะใช้การสุ่มด้วยค่าความน่าจะเป็น ถ้าจุดเริ่มต้นเป็นโฮสต์ปกติจะใช้สมการที่ (13) แต่ถ้าจุดเริ่มต้นเป็นโฮสต์สแปมจะใช้สมการที่ (17)

3.1 ถ้าจุดเริ่มต้นเป็นโฮสต์ปกติ เส้นทางให้เลือกจากลิงค์ที่ชี้ออกจากโฮสต์เท่านั้น ตัวอย่างจากโฮสต์กราฟภาพที่ 11 สมมติให้โฮสต์หมายเลขกำกับ 0 เป็นโฮสต์ปกติและถูกเลือกมาเป็นจุดเริ่มต้น โดยที่พิจารณามดเทียมทีละตัว ให้จำนวนก้าวจำกัดไว้ที่ 2 ก้าว มีตัวอย่างดังนี้ มดเทียมจากโฮสต์หมายเลขกำกับ 0 จะสามารถเลือกไปโฮสต์หมายเลขกำกับ 3 หรือโฮสต์หมายเลขกำกับ 4 ได้สองทาง สมมติมดเทียมตัดสินใจเลือกไปโฮสต์หมายเลขกำกับ 3 เมื่อมดเทียมอยู่ที่โฮสต์หมายเลขกำกับ 3 สามารถเลือกไปที่โฮสต์หมายเลขกำกับ 5 หรือโฮสต์หมายเลขกำกับ 6 ได้สองทาง สมมติมดเทียมเลือกโฮสต์หมายเลขกำกับ 5 เมื่อมดเทียมอยู่ที่โฮสต์หมายเลขกำกับ 5 จะหยุดเดินทันทีเนื่องจากจำนวนก้าวถูกจำกัดไว้ที่สองเก้า

3.2 ถ้าจุดเริ่มต้นเป็นโฮสต์สแปม เส้นทางให้เลือกจากลิงค์ที่ชี้เข้าหาโฮสต์เท่านั้น ตัวอย่างจากโฮสต์กราฟภาพที่ 11 สมมติให้โฮสต์หมายเลขกำกับ 2 เป็นโฮสต์สแปมและถูกเลือกมาเป็นจุดเริ่มต้น โดยที่พิจารณามดเทียมทีละตัว ให้จำนวนก้าวจำกัดไว้ที่ 2 ก้าว มีตัวอย่างดังนี้ มดเทียมจากโฮสต์หมายเลขกำกับ 2 จะสามารถเลือกไปโฮสต์หมายเลขกำกับ 1 หรือโฮสต์หมายเลขกำกับ 5 ได้สองทาง สมมติมดเทียมตัดสินใจเลือกไปโฮสต์หมายเลขกำกับ 5 เมื่อมดเทียมอยู่ที่โฮสต์หมายเลขกำกับ 5 สามารถเลือกไปที่โฮสต์หมายเลขกำกับ 3 ได้เพียงทางเดียว มดเทียมจำต้องเลือกโฮสต์หมายเลขกำกับ 3 เมื่อมดเทียมอยู่ที่โฮสต์หมายเลขกำกับ 3 จะหยุดเดินทันทีเนื่องจากจำนวนก้าวถูกจำกัดไว้ที่สองเก้า

4. ปรับค่าฟีโรโมน ทำซ้ำขั้นตอนที่ 3 จนถึงรอบการคำนวณที่ได้กำหนดไว้

5. กำหนดค่าฟีโรโมนเป็นค่าเริ่มต้น เลือกโฮสต์ภายในกลุ่มที่ไม่เคยถูกเลือก มาเป็นจุดเริ่มต้น ทำซ้ำขั้นตอนที่ 3 จนกระทั่งโฮสต์ภายในกลุ่มถูกเลือกทั้งหมด

จำนวนก้าวที่ถูกจำกัดของมดเทียมแต่ละตัวจะไม่เท่ากัน มดเทียมแต่ละตัวจะมีจำนวนก้าวที่ได้จากการสุ่ม โดยจะสุ่มจากเลขจำนวนเต็มบวกและเลขจำนวนเต็มบวกที่สุ่มได้นั้นต้องอยู่ใน

ขอบเขตที่ตัวแปรกำหนดไว้ เงื่อนไขการหยุดเดินนอกจากจำนวนก้าวที่ถูกจำกัดแล้วมดเทียมจะหยุดเดินเมื่อเจอโฮสต์กลุ่มตรงข้าม ถ้าเจอโฮสต์ที่ไม่มีคลาสก็จะทำการเดินต่อไป

หลังจากแต่ละรอบการคำนวณผลลัพธ์ของอัลกอริทึมแอนท์โคโลนีออพติไมเซชันเป็นเส้นทางการเดินของมดเทียม ซึ่งเส้นทางนี้สามารถนำมาสร้างเป็นกฎเพื่อจำแนกข้อมูล อัลกอริทึมจากงานวิจัยนี้จะเรียกว่า *LSD-ACO* (link-based spam detection using ant colony optimization) มีรหัสโค้ดเทียม (pseudo-code) ดังนี้

```
function splitData ( seeds,          // set of labeled hosts
                     nIters,         // number of iterations
                     nAnts,          // number of ants
                     minHopsS,       // number of minimum hops for seed is spam host
                     maxHopsS,       // number of maximum hops for seed is spam host
                     minHopsN,       // number of minimum hops for seed is normal host
                     maxHopsN, // number of maximum hops for seed is normal host
                     ts_con,         // threshold of confident
                     ts_sup)         // threshold of support
 1: seedSpam = ϕ
 2: seedNormal = ϕ
 3: Repeat
 4:     s ← dequeue (seeds)
 5:     if s.class = spam then
 6:         seedSpam = seedSpam ∪ s
 7:     Else
 8:         seedNormal = seedNormal ∪ s
 9:     end if
10: until seeds is empty
11: LSD-ACO (seedSpam, nIters, nAnts, minHopsS, maxHopsS)
12: LSD-ACO(seedNormal,nIters,nAnts, minHopsN, maxHopsN )
```

ภาพที่ 12 รหัสโค้ดเทียมการแบ่งกลุ่มข้อมูลชุดฝึกสอน

```
function LSD-ACO (seeds,        // set of labeled hosts
                  nIters,       // number of iterations
                  nAnts,        // number of ants
                  minHops,      // number of minimum hops
                  maxHops)      // number of maximum hops
                  ts_con,       // threshold of confident
                  ts_sup)       // threshold of support
 1: rules = φ
 2: η← calculateHeuristic ( )
 3: Repeat
 4:    s ← dequeue (seeds)
 5:    τ ← initializePheromone ( )
 6:    bestRules = φ
 7:    maxQ = 0
 8:    for t = 0 to nIters – 1 do
 9:       paths = φ
10:       for a = 0 to nAnts – 1 do
11:          nHops = randomBetween(minHops, maxHops)
12:          path ← walk (s, τ, nHops)
13:          paths ← paths ∪ path
14:       end for
15:       allRules ← generateRuleOnPath (paths, ts_con , ts_sup)
16:       (curRule, curPath, q) ← selectHighQRule (allRules)
17:       τ ← updatePheromone (q, curPaths)
18:       if maxQ < q then
19:          bestRules = curRules
20:          maxQ = q
21:       end if
22:    end for
23:    rules ← rules ∪ bestRules
24: until seeds is empty
25: return rules

function walk (s,               // starting host
               τ,               // pheromone information
               nHops)           // number of hops
 1: path = φ
 2: for i = 0 to nHops – 1 do
 3:    if s.class = spam then r ← randomlySelectHostByBackwardLink (s, τ)
 4:    else r ← randomlySelectHostByForwardLink (s, τ)
 5:    path ← appendLink(path, e(s, r))
 6:    s = r
 7: end for
 8: return path
```

ภาพที่ 13 รหัสโค้ดเทียมฟังก์ชัน *LSD-ACO*

อธิบายรหัสโค้ดเทียมการแบ่งกลุ่มข้อมูลชุดฝึกสอน (ภาพที่ 12)

- บรรทัดที่ 1 กำหนดเซตกลุ่มโฮสต์สแปมเป็นเซตว่าง
- บรรทัดที่ 2 กำหนดเซตกลุ่มโฮสต์ปกติเป็นเซตว่าง
- บรรทัดที่ 3 ถึง 10 จะพิจารณาโฮสต์ในข้อมูลชุดฝึกสอนทีละโฮสต์ ถ้าโฮสต์ใดมีคลาสเป็นโฮสต์สแปมให้นำไปใส่ในเซตกลุ่มโฮสต์สแปม ถ้าโฮสต์ใดมีคลาสเป็นโฮสต์ปกติให้นำไปใส่ในเซตกลุ่มโฮสต์ปกติ
- บรรทัดที่ 11 จากเซตกลุ่มโฮสต์สแปมนำไปสร้างกฎด้วยฟังก์ชัน *LSD-ACO*
- บรรทัดที่ 12 จากเซตกลุ่มโฮสต์ปกตินำไปสร้างกฎด้วยฟังก์ชัน *LSD-ACO*

อธิบายรหัสโค้ดเทียมฟังก์ชัน *LSD-ACO* (ภาพที่ 13)

- บรรทัดที่ 2 คำนวณค่าฟรีโรโมน
- ลูพ (loop) บรรทัดที่ 3 ถึง 25 เลือกโฮสต์เพื่อสร้างเป็นจุดเริ่มต้นจนครบทุกโฮสต์
- บรรทัดที่ 5 กำหนดค่าฟีโรโมนเป็นค่าเริ่มต้น
- ลูพ บรรทัดที่ 8 ถึง 22 เป็นการคำนวณแบบวนซ้ำ ตามจำนวนรอบที่ได้ตั้งไว้
- ลูพ บรรทัดที่ 10 ถึง 14 มดเทียมแต่ละตัวสร้างเส้นทางจนครบทุกตัว
- บรรทัดที่ 11 กำหนดจำนวนก้าวให้มดเทียมแต่ละตัว โดยที่จำนวนก้าวได้มาจากการสุ่มระหว่างช่วงของจำนวนจริงบวก ซึ่งช่วงจำนวนจริงบวกจะถูกกำหนดด้วยตัวแปรและจุดเริ่มต้นที่เป็นโฮสต์ปกติกับโฮสต์สแปมมีช่วงที่ถูกระบุมา
- บรรทัดที่ 12 มดเทียมเดินโดยการเรียกใช้ฟังก์ชัน *walk*
- บรรทัดที่ 15 เมื่อมดเทียมทุกตัวสิ้นสุดการเดิน นำเส้นทางการเดินของแต่ละมดเทียมไปสร้างเป็นกฎ กฎที่ได้จะถูกตรวจคุณสมบัติว่ามีเงื่อนไขจากค่า ts_sup และ ts_con ถ้าไม่ผ่านเงื่อนไขดังกล่าวกฎจะถูกลบทิ้ง
- บรรทัดที่ 16 ค้นหาเส้นทางที่นำมาสร้างกฎแล้วให้ค่า q สูงที่สุด (อาจจะมีมากกว่า 1 เส้นทาง)
- บรรทัดที่ 17 ปรับค่าฟีโรโมน
- บรรทัดที่ 18 ถึง 21 เลือกกฎที่มีค่า q สูงที่สุดจากแต่ละรอบการคำนวณ
- บรรทัดที่ 23 เมื่อสิ้นสุดรอบการคำนวณแล้วนำกฎที่ดีที่สุดเก็บไว้ในเซต *rules* จากนั้นทำซ้ำบรรทัดที่ 3 จนครบทุกโฮสต์

อธิบายรหัสโค้ดเทียมฟังก์ชัน *walk* (ภาพที่ 13)

- ลูพ บรรทัดที่ 2 ถึง 7 มดเทียมจากฟังก์ชัน *LSD-ACO* จะเดินด้วยจำนวนก้าวที่ได้กำหนดไว้
- บรรทัดที่ 3 ถึง 4 มดเทียมตัดสินใจเลือกโฮสต์เพื่อสร้างเส้นทาง โดยการสุ่มด้วยค่าความน่าจะเป็น ถ้าจุดเริ่มต้นเป็นโฮสต์ปกติค่าความน่าจะเป็นจะคำนวณจากสมการที่ (13) ถ้าจุดเริ่มต้นเป็นโฮสต์สแปมค่าความน่าจะเป็นจะคำนวณค่าสมการที่ (17)
- บรรทัดที่ 5 นำโฮสต์ที่ได้มาสร้างเป็นเส้นทาง
- บรรทัดที่ 6 มดเทียมเดินไปยังโฮสต์ที่เลือกจากบรรทัดที่ 3

การสร้างกฎ

ในขั้นตอนการสร้างเส้นทางที่ได้อธิบายมา เส้นทางสามารถจำแนกประเภทของโฮสต์ได้ โดยถือว่าทุกโฮสต์ภายในเส้นทางจะมีคลาสเหมือนกับจุดเริ่มต้น แต่เส้นทางไม่สามารถบรรยายความแตกต่างระหว่างคุณลักษณะของโฮสต์สแปมและโฮสต์ปกติได้ การนำเส้นทางมาสร้างเป็นกฎช่วยให้เห็นข้อแตกต่างระหว่างคุณลักษณะของโฮสต์สแปมและโฮสต์ปกติ โดยที่แต่ละโฮสต์จะประกอบไปด้วยคุณลักษณะต่างๆ ที่สกัดมาได้คุณลักษณะเหล่านี้ต้องเป็นข้อมูลเชิงกลุ่ม (categorical data) เท่านั้น ให้ $\{A_1, A_2, ..., A_m\}$ เป็นเซตของคุณลักษณะโฮสต์จำนวน m คุณลักษณะ และ $\{a_{i1}, a_{i2}, ..., a_{in_i}\}$ เป็นเซตของค่าทั้งหมดที่เป็นไปได้จำนวน n_i ค่าของคุณลักษณะ A_i ซึ่งจากคุณลักษณะพวกนี้สามารถนำมาสร้างเป็นกฎให้อยู่ให้รูปแบบดังนี้

$$(A_1 = a_{1x}, A_2 = a_{2y}, ..., A_m = a_{mz}) \Rightarrow (C = normal\_or\_spam) \quad (23)$$

โดยกฎจะประกอบไปด้วยสองส่วนคือส่วนของเงื่อนไข $(A_1 = a_{1x}, A_2 = a_{2y}, ..., A_m = a_{mz})$ และส่วนของผลการทำนาย $(C = normal\_or\_spam)$ สำหรับการสร้างกฎจะสร้างจากคุณลักษณะของโฮสต์ที่มีค่าเหมือนกันในแต่ละโฮสต์จากเส้นทางที่ได้รับ ให้ Γ_{EH_p} เป็นเซตของโฮสต์ที่อยู่บนเส้นทาง p กฎที่ได้จากเส้นทางถูกกำหนดดังนี้

$$\underset{h \in \Gamma_{EH_p}}{COMMON}(A_1 = a_{1x}^h, ..., A_m = a_{mx}^h) \Rightarrow (C = class) \quad (24)$$

ซึ่ง C จะพิจารณาจากจุดเริ่มต้นของเส้นทาง ถ้าจุดเริ่มต้นว่าเป็นโฮสต์สแปม $C = spam$ หรือถ้าจุดเริ่มต้นเป็นโฮสต์ปกติ $C = normal$ ซึ่งถ้าหากพิจารณาปริมาณโฮสต์ในเส้นทางที่นำมาสร้างเป็นกฎแล้วจะพบว่า เมื่อจำนวนโฮสต์มากขึ้นคุณลักษณะที่มีค่าเหมือนกันจะมีจำนวนลดลงส่งผลให้เงื่อนไขสั้นลงไปด้วย หลังจากที่ได้กฎมาจากข้อมูลชุดฝึกสอนทั้งหมดแล้ว จะนำกฎนี้จะไปทดสอบกับข้อมูลชุดทดสอบเพื่อวัดประสิทธิภาพในขั้นตอนต่อไป

การใช้กฎกับข้อมูลชุดทดสอบ

เมื่อเลือกโฮสต์จากข้อมูลชุดทดสอบซึ่งจะไม่สามารถทราบคลาสของโฮสต์ มีความเป็นไปได้ที่จะถูกครอบคลุมจากทั้งกฎที่มีคำตอบเป็นโฮสต์ปกติและกฎที่มีคำตอบเป็นโฮสต์สแปม ในงานวิจัยนี้จะใช้วิธีการลงคะแนนเสียงแบบถ่วงน้ำหนักตามค่าคุณภาพที่คำนวณจากข้อมูลชุดฝึกสอน ให้ R_s เป็นเซตของกฎที่มีคำตอบเป็นโฮสต์สแปมและมีเงื่อนไขของกฎครอบคลุมคุณลักษณะของโฮสต์ที่เลือกจากข้อมูลชุดทดสอบ สามารถคำนวณคะแนนของคำตอบที่เป็นสแปมได้ดังนี้

$$spam\_score = \sum_{r \in R_s} q_r \tag{25}$$

โดยที่ q_r เป็นค่าคุณภาพของกฎ

ในข้อมูลชุดฝึกสอนเช่นเดียวกันกับคำตอบที่เป็นโฮสต์ปกติ ให้ R_n เป็นเซตของกฎที่มีคำตอบเป็นโฮสต์ปกติและมีเงื่อนไขของกฎครอบคลุมคุณลักษณะของโฮสต์ที่เลือกจากข้อมูลชุดทดสอบ

$$normal\_score = \sum_{r \in R_n} q_r \tag{26}$$

จากนั้นทำการเปรียบเทียบค่าคะแนนให้คำตอบเป็นไปตามคลาสที่มีคะแนนมากกว่า แต่ก็ยังมีความเป็นไปได้อยู่บ้างที่ค่าคะแนนจะเท่ากันในงานวิจัยนี้เลือกให้คำตอบเป็นโฮสต์ปกติ และความเป็นไปได้อีกอย่างคือเมื่อเลือกโฮสต์มาจากข้อมูลชุดทดสอบอาจพบว่าไม่สามารถหากฎที่ครอบคลุมโฮสต์นี้ได้ในงานวิจัยนี้แก้ไขปัญหานี้โดยให้โฮสต์นั้นมีคำตอบเป็นโฮสต์สแปม

```
function Classifired (listRule,     // set of rules
                      testData,     // testing data
                      trainData)    // training data
 1: repeat
 2:     t ← dequeue (testData)
 3:     spam_score = 0
 4:     normal_score = 0
 5:     for i = 0 to listRule.numberRule – 1 do
 6:         r=getRule(listRule,i)
 7:         isCover =checkCover(t,r)
 8:         if isCover  then
11:             q=calculateQuality(r,trainData)
12:             if r.class = spam then spam_score += q
13:             else normal_score += q end if
14:         end if
15:     end for
16:     if spam_score > normal_score then
17:         class = spam
18:     else class = normal end if
19:     listClass ← listClass ∪ class
20: until testData is empty
21: return listClass
```

ภาพที่ 14 รหัสโค้ดเทียมจำแนกประเภทโฮสต์

อธิบายรหัสโค้ดเทียมจำแนกประเภทโฮสต์ (ภาพที่ 14)

- ลูพ บรรทัดที่ 1 ถึง 20 เลือกโฮสต์จากชุดข้อมูลทดสอบมาทีละโฮสต์
- ลูพ บรรทัดที่ 5 ถึง 15 พิจารณากฎทั้งหมดที่มีอยู่
- บรรทัดที่ 7 ถึง 10 ตรวจสอบว่ากฎครอบคลุมโฮสต์ที่ถูกเลือกอยู่หรือไม่
- บรรทัดที่ 8 ถึง 14 ถ้าหากข้อมูลนั้นถูกครอบคลุมโดยกฎ จะทำการคำนวณค่า q
- บรรทัดที่ 11 คำนวณค่า q ของกฎที่ครอบคลุมกับข้อมูลชุดฝึกสอน
- บรรทัดที่ 12 ถึง 13 ตรวจสอบคำตอบของกฎที่ครอบคลุม ถ้าคำตอบเป็นโฮสต์สแปมให้บวกคะแนน *spam_score* ด้วยค่า q แต่ถ้าคำตอบเป็นโฮสต์ปกติให้บวกคะแนน *normal_score* ด้วยค่า q
- บรรทัดที่ 16 ถึง 18 เปรียบเทียบคะแนน *spam_score* กับ *normal_score* ถ้า *spam_score* มีค่ามากกว่าให้โฮสต์ที่ถูกเลือกจากบรรทัดที่ 2 มีคลาสเป็นโฮสต์สแปม แต่ถ้าไม่ใช่แล้วให้คลาสเป็นโฮสต์ปกติ

ในการพัฒนาโปรแกรมจริง เพื่อลดเวลาการคำนวณค่าคุณภาพของกฎ ในบรรทัดที่ 11 สามารถคำนวณเก็บไว้เนื่องจากจะไม่เปลี่ยนแปลงยกเว้นจะเปลี่ยนข้อมูลชุดฝึกสอน

แผนผังการทำงาน

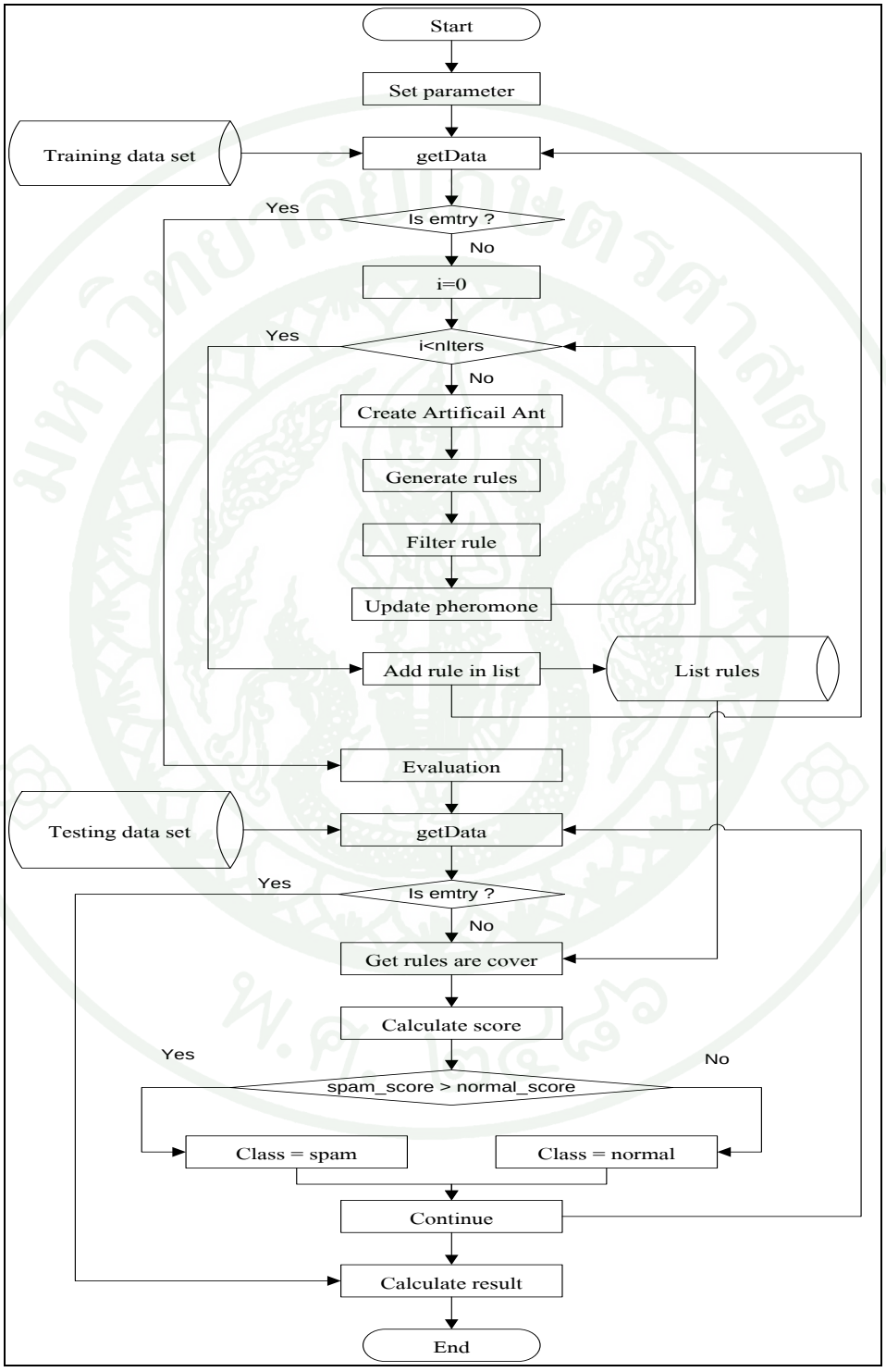
จากรายละเอียดทั้งหมดที่ได้กล่าวมาในข้างต้นจะนำมาสรุปเป็นภาพรวมเพื่อง่ายต่อการเข้าใจ พิจารณารูปประกอบจากภาพที่ 15 เริ่มแรกในโปรแกรมต้องมีกำหนดค่าตัวแปรเสียก่อนโดยตัวแปรมีดังนี้

1. จำนวนมดเทียม
2. จำนวนรอบการคำนวณ
3. จำนวนก้าวน้อยที่สุดเมื่อจุดเริ่มต้นเป็นโฮสต์ปกติ
4. จำนวนก้าวมากที่สุดเมื่อจุดเริ่มต้นเป็นโฮสต์ปกติ
5. จำนวนก้าวน้อยที่สุดเมื่อจุดเริ่มต้นเป็นโฮสต์สแปม
6. จำนวนก้าวมากที่สุดเมื่อจุดเริ่มต้นเป็นโฮสต์สแปม
7. ค่าความแม่นยำต่ำสุดของกฎที่สามารถยอมรับได้
8. จำนวนข้อมูลที่ครอบคลุมต่ำสุดของกฎจากข้อมูลชุดฝึกสอนที่สามารถยอมรับได้

เมื่อกำหนดค่าตัวแปรเรียบร้อยทั้งหมดแล้วขั้นตอนต่อไปคือนำเอาโฮสต์จากข้อมูลชุดฝึกสอนที่มีคลาสมาเป็นจุดเริ่มต้น จากนั้นสร้างจำนวนมดเทียมเท่ากับตัวแปรที่ได้กำหนดไว้ ปรับค่าฟีโรโมนเป็นค่าเริ่มต้น มดเทียมแต่ละตัวจะสร้างกฎมาหนึ่งกฎด้วยจำนวนก้าวที่สุ่มจากขอบเขตตอบตัวแปรที่ได้กำหนดไว้คลาสของกฎจะเป็นไปตามจุดเริ่มต้นของมดเทียม ตรวจสอบดูว่ากฎที่ได้มามีคุณสมบัติผ่านตามตัวแปรหรือไม่ถ้าไม่ให้มดตัวนั้นจะถูกทำลายทิ้ง เลือกมดเทียมตัวที่ให้กฎดีที่สุด ปรับค่าฟีโรโมนตามคุณภาพของกฎจนครบจำนวนรอบตามตัวแปรจำนวนรอบการคำนวณ จากนั้นเลือกกฎที่ดีที่สุดใส่ในรายการกฎ กำหนดค่าฟรีโรโมนเป็นค่าเริ่มต้นจากนั้นจึงสร้างกฎจากโฮสต์ต่อไป ทำไปเรื่อยๆ จนครบทุกโฮสต์ในข้อมูลชุดฝึกสอน

ส่วนขั้นตอนการนำกฎมาใช้นั้น เลือกโฮสต์จากข้อมูลชุดทดสอบตรวจสอบกฎในรายการดูว่ามีกฎใดที่ครอบคลุมบ้าง ให้นำกฎเหล่านั้นมาคิดคะแนน $spam\_score$ และ $normal\_score$

จากนั้นจึงทำการเปรียบเทียบคะแนนคลาสของโฮสต์ที่เลือกจากข้อมูลชุดทดสอบให้เป็นไปตามฝั่งที่คะแนนมากกว่า ทำจนครบทุกโฮสต์ในข้อมูลชุดทดสอบจากนั้นจึงทำการวัดผล



ภาพที่ 15 แผนผังการทำงาน

ผลและวิจารณ์

ข้อมูล

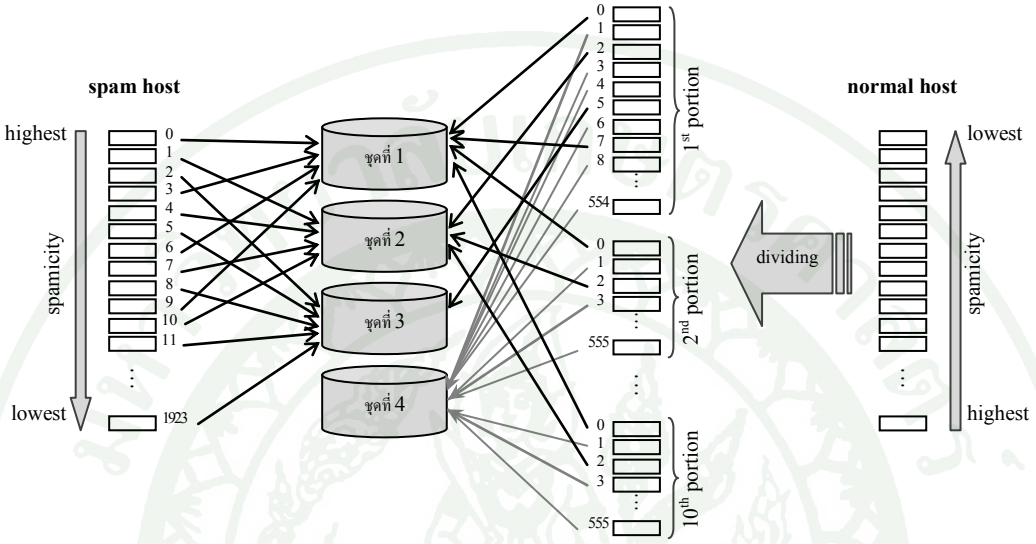
ในงานวิจัยนี้ได้ใช้ชุดข้อมูล Web Spam-UK 2006 (Castillo *et al*., 2006) ซึ่งเป็นข้อมูลทดสอบมาตรฐาน (benchmark) ใช้ในงานวิจัยต่างๆ ด้านเว็บสแปม ข้อมูลรวบรวมจากโดเมน (domain) .UK ในปี ค.ศ. 2006 โดยห้องปฏิบัติการ Laboratory of Web Algorithmics มหาวิทยาลัย Degli Studi di Milano ประกอบไปด้วยเว็บเพจจำนวน 77 ล้านเว็บเพจ 2,965 ล้านลิงค์ คิดเป็นจำนวนโฮสต์ทั้งสิ้น 11,402 โฮสต์ได้รับการคัดแยกระหว่างโฮสต์สแปมและโฮสต์ปกติ จากอาสาสมัครนักวิจัย พบว่ามีโฮสต์สแปมจำนวน 1924 โฮสต์และโฮสต์ปกติจำนวน 5549 โฮสต์ที่เหลือยังไม่ได้รับการคัดแยก

การเตรียมข้อมูล

เนื่องจากการคัดแยกโฮสต์สแปมในงานวิจัยนี้ไม่รองรับคุณลักษณะของข้อมูลแบบต่อเนื่อง (continuous data) จึงจำเป็นต้องอาศัยงานวิจัยก่อนหน้าที่ศึกษาเกี่ยวกับการแบ่งข้อมูลเป็นช่วงๆ (discretization) ในงานวิจัยนี้ได้เลือกงานวิจัยของ (Fayyad and Irani, 1993) ซึ่งมีการนำมาใช้ในงานวิจัยอื่นเป็นจำนวนมาก มาใช้ในการแบ่งข้อมูลเป็นช่วงๆ จากนั้นจึงเตรียมข้อมูล เนื่องจากอัลกอริทึมที่ใช้ในการเรียนรู้โดยทั่วไปแล้ว ถ้าข้อมูลชุดฝึกสอนในแต่ละคลาสมีจำนวนที่แตกต่างกันมากแล้ว อัลกอริทึมส่วนใหญ่จะมีผลการทำนายจะโน้มเอียงไปทางคลาสที่มีจำนวนมากกว่า เพื่อหลีกเลี่ยงปัญหานี้ ข้อมูลชุดฝึกสอนจะถูกกำหนดให้มีจำนวนข้อมูลโฮสต์สแปมกับโฮสต์ปกติใกล้เคียงกัน สามารถดูรูปประกอบการอธิบายได้จากภาพที่ 16

- สำหรับข้อมูลโฮสต์สแปม ให้เริ่มต้นเรียงลำดับโฮสต์ด้วยค่าความเป็นสแปม (spamicity) จากมากไปน้อย แต่ละโฮสต์จะมีหมายเลขกำกับโดยเริ่มตั้งแต่ 0 และเพิ่มขึ้นทีละ 1 โฮสต์สแปมจะถูกแบ่งออกเป็นสามชุด โดยพิจารณาจากเศษที่เหลือของการหารหมายเลขกำกับด้วย 3 หากเศษที่เหลือมีค่าเป็น 0 1 และ 2 โฮสต์สแปมดังกล่าวนั้นจะถูกจำแนกลงใน “ชุดที่1” “ชุดที่2”และ“ชุดที่3” ตามลำดับซึ่งสามชุดจะมีโฮสต์สแปมประมาณ 640 โฮสต์
- สำหรับโฮสต์ปกติจะกระทำในทำนองเดียวกันกล่าวคือ เรียงลำดับโฮสต์ด้วยค่าควา

เป็นสแปม จากน้อยไปมาก จากนั้นจึงแบ่งออกเป็น 10 ส่วน แต่ละส่วนจะมีหมายเลขกำกับโดยเริ่มต้นจาก 0 หมายเลขกำกับทั้งหมดจะถูกหารด้วย 7 โฮสต์ที่มีเศษเหลือเป็น 0, 2, 5 จะถูกใส่ไว้ในที่ “ชุดที่ 1”, “ชุดที่ 2” และ “ชุดที่ 3” ตามลำดับ สำหรับโฮสต์ปกติ ที่เหลือจะถูกใส่ไว้ใน “ชุดที่ 4”



ภาพที่ 16 การแบ่งข้อมูลในขั้นตอนการเตรียมข้อมูล

รูปแบบในการทดลอง

การเรียนรู้ของอัลกอริทึม *LSD-ACO* เป็นประเภทการเรียนรู้แบบมีผู้สอน (supervised learning) ขั้นตอนการเรียนรู้ต้องอาศัยข้อมูลชุดฝึกสอน หลังจากนั้นจะทดสอบประสิทธิภาพการเรียนรู้ด้วยข้อมูลชุดทดสอบ ในการทดสอบประสิทธิภาพออกเป็น 3 การทดลองดังนี้

1. การทดลองที่ 1: ใช้ข้อมูล “ชุดที่ 1” เป็นข้อมูลชุดฝึกสอนข้อมูล ใช้ข้อมูล “ชุดที่ 2”, “ชุดที่ 3” และ “ชุดที่ 4” เป็นข้อมูลชุดทดสอบ

2. การทดลองที่ 2: ใช้ข้อมูล “ชุดที่ 2” เป็นข้อมูลชุดฝึกสอนข้อมูล ใช้ข้อมูล “ชุดที่ 1”, “ชุดที่ 3” และ “ชุดที่ 4” เป็นข้อมูลชุดทดสอบ

3. การทดลองที่ 3: ใช้ข้อมูล “ชุดที่ 3” เป็นข้อมูลชุดฝึกสอนข้อมูล ใช้ข้อมูล “ชุดที่ 1”, “ชุดที่ 2” และ “ชุดที่ 4” เป็นข้อมูลชุดทดสอบ

ในขั้นตอนการเรียนรู้จะนำข้อมูลทั้งหมดมาใช้ แต่ขั้นตอนในนี้อัลกอริทึมจะสามารถรู้ว่าโฮสต์ใดเป็นโฮสต์สแปมหรือโฮสต์ปกติได้เฉพาะโฮสต์ที่อยู่ในข้อมูลชุดฝึกสอน เมื่อได้กฎมาแล้วจึงนำมาทำการทดสอบด้วยข้อมูลชุดทดสอบ แต่ละการทดลองจะทำซ้ำจำนวน 5 ครั้ง เพื่อหาค่าเฉลี่ย

มาตรวัด

ในงานวิจัยนี้ผลลัพธ์ที่ได้จากการเรียนรู้จะอยู่ในรูปแบบของกฎ ซึ่งกฎสามารถแบ่งออกตามการทำนายได้เป็น กฎที่ทำนายว่าเป็นโฮสต์สแปมกับกฎที่ทำนายว่าเป็นโฮสต์ปกติ เมื่อทำการวัดผลจะนำคลาสที่ได้จากการทำนายมาเปรียบเทียบกับคลาสที่แท้จริงของโฮสต์ ซึ่งเหตุการณ์ที่เกิดขึ้นจากการทดลองเป็นไปได้ทั้งหมด 4 กรณีดังตารางที่ 1 จากนั้นจึงนับจำนวนที่เกิดขึ้นในแต่ละกรณี โดยนิยามให้

1. tp คือจำนวนโฮสต์ปกติที่ถูกทำนายถูกจริง
2. fp คือจำนวนโฮสต์สแปมที่ถูกทำนายผิดว่าเป็นโฮสต์ปกติ
3. tn คือจำนวนโฮสต์สแปมที่ถูกทำนายถูกจริง
4. fn คือจำนวนโฮสต์ปกติที่ถูกทำนายผิดว่าเป็นโฮสต์สแปม

ตารางที่ 1 ตารางแสดงผลลัพธ์จากการทำนาย

		คลาสที่แท้จริง	
		โฮสต์ปกติ	โฮสต์สแปม
คลาสจากการทำนาย	โฮสต์ปกติ	true positive (tp)	false positive (fp)
	โฮสต์สแปม	false negative (fn)	true negative (tn)

มาตรที่ใช้ในการวัดประสิทธิภาพได้แก่ (1) true positive rate บ่งบอกถึงอัตราส่วนข้อมูลส่วนที่ทำนายว่าเป็นโฮสต์ปกตินั้นครอบคลุมข้อมูลโฮสต์ปกติที่แท้จริง ต่อข้อมูลโฮสต์ปกติที่แท้จริงทั้งหมด สามารถคำนวณได้ดังนี้

$$tp\_rate = \frac{tp}{tp + fn} \tag{27}$$

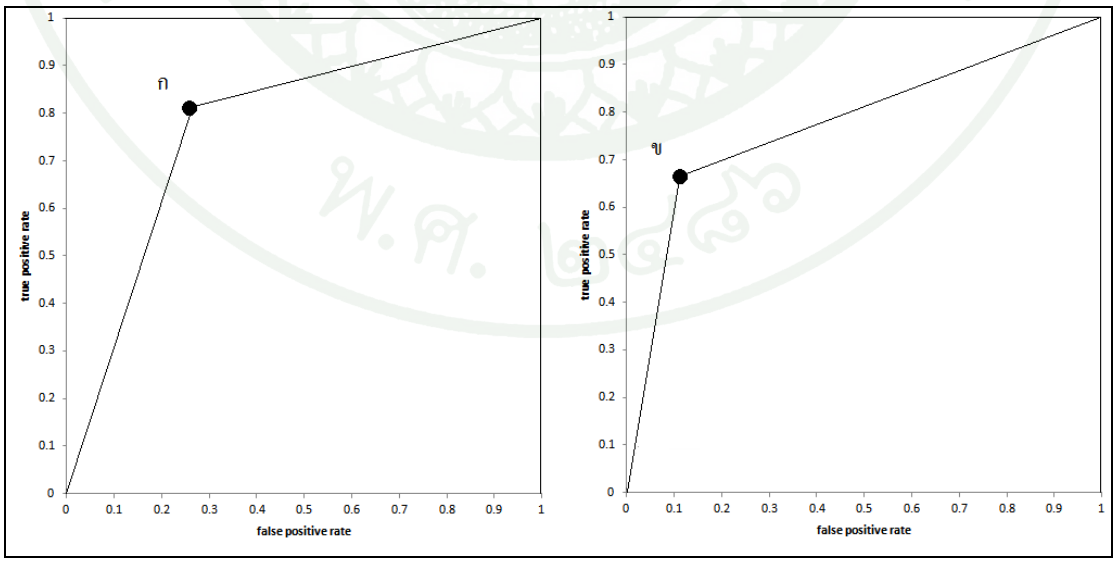
(2) false positive rate บ่งบอกถึง อัตราส่วนข้อมูลส่วนที่ทำนายว่าเป็นโฮสต์ปกตินั้นครอบคลุมข้อมูลที่เป็นโฮสต์สแปม ต่อจำนวนโฮสต์สแปมที่แท้จริงทั้งหมด ซึ่งสามารถคำนวณได้ดังนี้

$$fp\_rate = \frac{fp}{fp + tn} \tag{28}$$

อัลกอริทึมที่สามารถทำนายอย่างมีประสิทธิภาพนั้นจะต้องมีค่า true positive rate ที่มาก (น้อยที่สุดเป็น 0 และมากที่สุดเป็น 1) และมีค่า false positive rate ที่น้อย (น้อยที่สุดเป็น 0 และมากที่สุดเป็น 1) อย่างไรก็ดี อาจมีคำถามเกี่ยวกับความขัดแย้งที่เกิดขึ้นได้ ยกตัวอย่างการเปรียบเทียบประสิทธิภาพ ระหว่างอัลกอริทึม ก และอัลกอริทึม ข ดังนี้

- ค่า true positive rate ของอัลกอริทึม ก สูงกว่า อัลกอริทึม ข
- ค่า false positive rate ของอัลกอริทึม ข น้อยกว่า อัลกอริทึม ก

ซึ่งความขัดแย้งในการวัดประสิทธิภาพที่เกิดขึ้นดังกล่าว ในงานวิจัยนี้จะตัดสินด้วยการคำนวณพื้นที่ใต้กราฟ โดยมีขั้นตอนในการวาดกราฟดังนี้ ให้แกนนอนเป็นค่า false positive rate และแกนตั้งเป็นค่า true positive rate แล้วจึงลงจุดค่าที่ได้ของอัลกอริทึมทั้งสอง พร้อมลากเส้นเชื่อมจากจุดดังกล่าวไปยังจุดมุมบนขวา (1,1) และจุดมุมล่างซ้าย (0,0) แสดงดังภาพที่ 17



ภาพที่ 17 ตัวอย่างพื้นที่ใต้กราฟเปรียบเทียบระหว่างสองอัลกอริทึม

ซึ่งอัลกอริทึมที่ให้พื้นที่ใต้กราฟมากกว่าย่อมมีประสิทธิภาพมากกว่า วิธีการดังที่ได้กล่าวไปนี้เป็นวิธีที่ถูกนำมาใช้อย่างแพร่หลายในการวัดประสิทธิภาพด้านเครื่องจักรเรียนรู้ (machine leaning)

ผลการทดลอง

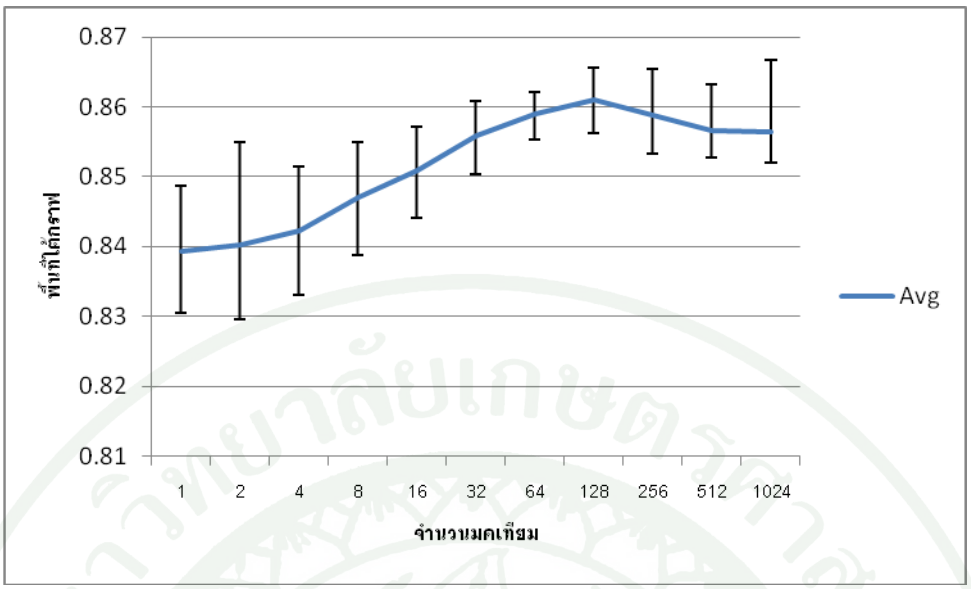
ในแต่ละการทดลองจะทำทั้งหมด 5 ครั้งจากนั้นจึงนำมาหาค่าเฉลี่ย กราฟของผลการทดลองแกนตั้งจะเป็นค่าของพื้นที่ใต้กราฟ แกนนอนจะเป็นค่าของตัวแปรต่างๆ โดยที่กราฟจะแสดงค่าเฉลี่ย ค่าสูงสุดที่ได้จากการทดลองซ้ำ 5 ครั้งและค่าต่ำสุดที่ได้จากการทดลองซ้ำ 5 ครั้ง

ทดลองปรับค่าตัวแปร

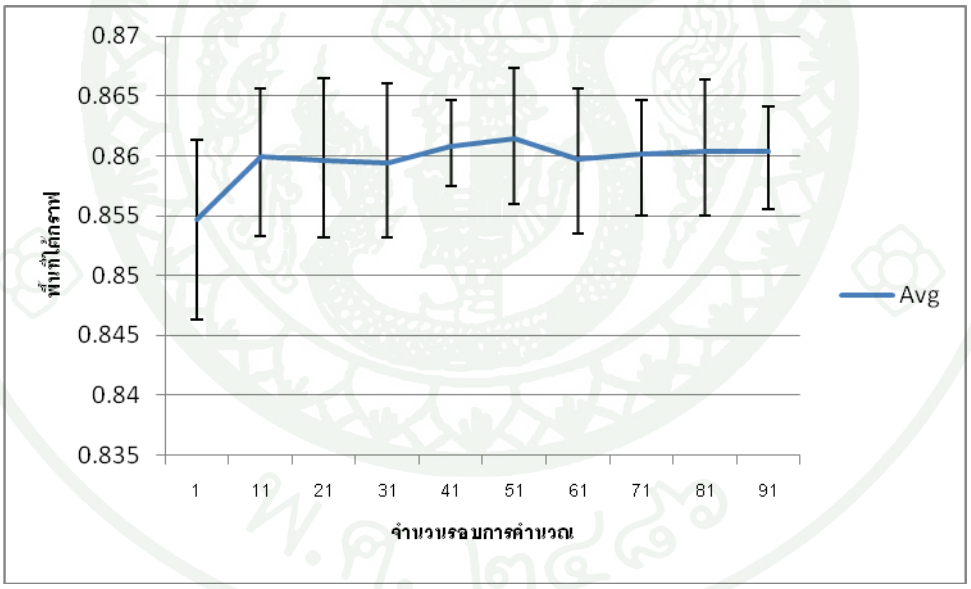
ในการทดลองปรับค่าตัวแปรนั้นจะทำการปรับค่าตัวแปรทีละตัว เพื่อพิจารณาการส่งผลต่อประสิทธิภาพ ตัวแปรที่เหลือจะมีค่าคงที่

การทดลองแรกจะทำการปรับค่าตัวแปรจำนวนมดเทียมที่ใช้ โดยเริ่มต้นจำนวนมดเทียม 1 ตัวจนถึง 1024 โดยการปรับจะเพิ่มจำนวนมดเทียมเป็นสองเท่าของจำนวนมดเทียมที่ใช้ก่อนหน้านี้ สามารถพิจารณาผลการทดลองได้จากภาพที่ 18 จากรูปจะเห็นว่าในช่วงแรก (จำนวนมดเทียมน้อยกว่า 128 ตัว) ประสิทธิภาพจะดีขึ้นเรื่อยๆ เมื่อจำนวนมดเทียมเพิ่มขึ้น แต่เมื่อเพิ่มจำนวนมดเทียมมากกว่า 128 ตัวแล้วประสิทธิภาพจะลดลงเล็กน้อย เนื่องจากจำนวนมดเทียมจะส่งผลโดยตรงต่อจำนวนกฎที่ได้ถ้าจำนวนมดเทียมมีมากกฎก็ที่ได้ก็จะมีมากขึ้นไปด้วย ทำให้จำนวนกฎที่ผ่านเงื่อนไขมีมากขึ้นไปด้วย

การทดลองปรับค่าตัวแปรรอบในการคำนวณ ในการทดลองตัวแปรมีค่าตั้งแต่ 1 ถึง 91 รอบ โดยจะปรับเพิ่มจำนวนรอบเพิ่มขึ้นสิบรอบ เมื่อเปลี่ยนค่าตัวแปร สามารถพิจารณาผลการทดลองได้จากภาพที่ 19 โดยที่ค่าตัวแปรตั้งแต่ 1 ถึง 51 ประสิทธิภาพของอัลกอริทึมมีแนวโน้มเพิ่มขึ้นเรื่อยๆ แต่เมื่อค่าตัวแปรเป็น 61 ผลการทดลองจะตกเล็กน้อยจากนั้นจะค่อนข้างคงที่ ซึ่งจำนวนรอบจะส่งผลถึงคุณภาพกฎที่ได้เพราะเนื่องจากอัลกอริทึมจะเลือกกฎที่ดีที่สุด (วัดจากค่าคุณภาพ) ในแต่ละรอบ ทำให้เมื่อเพิ่มจำนวนรอบโอกาสที่จะได้กฎที่ดีนั้นมีมากขึ้น



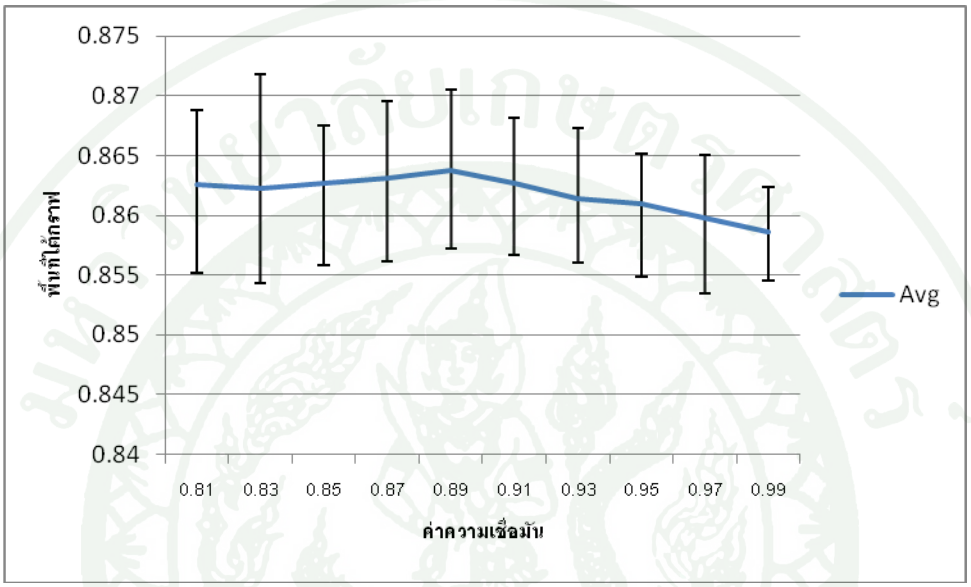
ภาพที่ 18 ผลการทดลองเปรียบเทียบประสิทธิภาพเมื่อปรับเปลี่ยนจำนวนมดเทียม



ภาพที่ 19 ผลการทดลองเปรียบเทียบประสิทธิภาพเมื่อปรับเปลี่ยนจำนวนรอบการคำนวณ

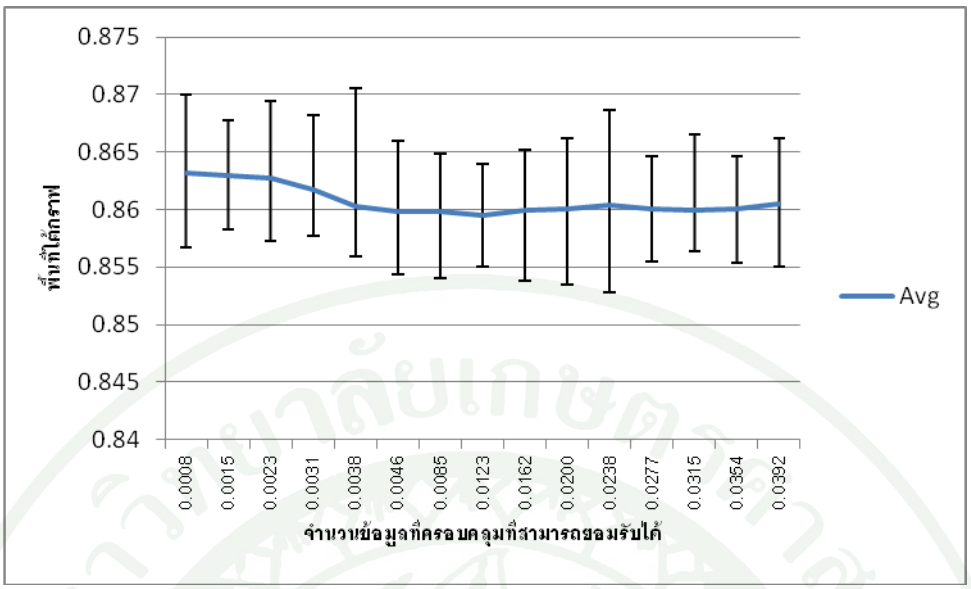
การทดลองปรับค่าความเชื่อมั่นต่ำสุดของกฎที่สามารถยอมรับได้ ในการทดลองตัวแปรมีค่าตั้งแต่ 0.81 ถึง 0.99 โดยจะเพิ่มค่าตัวแปรทีละ 0.03 เมื่อเปลี่ยนค่าตัวแปร สามารถพิจารณาผลการทดลองได้จากภาพที่ 20 โดยจะสังเกตเห็นว่าเมื่อช่วงค่าตัวแปรอยู่ระหว่าง 0.81 ถึง 0.89 ประสิทธิภาพของอัลกอริทึมจะเพิ่มขึ้นเรื่อยๆ และเมื่อค่าตัวแปรมากกว่า 0.89 ประสิทธิภาพของ

อัลกอริทึมจะค่อยๆ ลดลง เนื่องจากค่าความเชื่อมั่นเป็นเงื่อนไขในการตรวจสอบกฎที่สร้างขึ้นมาว่าสามารถยอมรับได้หรือไม่ ถ้าค่านี้ต่ำกฎที่ผ่านเงื่อนไขนี้จะมีจำนวนมาก ทำให้กฎที่ได้อาจจะได้กฎที่มีประสิทธิภาพต่ำมาด้วย แต่ในทางกลับกันถ้าค่านี้สูงกฎที่ผ่านเงื่อนไขนี้จะมีจำนวนน้อย โอกาสที่โฮสต์จากข้อมูลชุดทดสอบจะไม่ถูกครอบคลุมโดยกฎที่ผ่านเงื่อนไขจะมีมากขึ้น



ภาพที่ 20 ผลการทดลองเปรียบเทียบประสิทธิภาพเมื่อปรับเปลี่ยนค่าความเชื่อมั่นต่ำสุดที่สามารถยอมรับได้

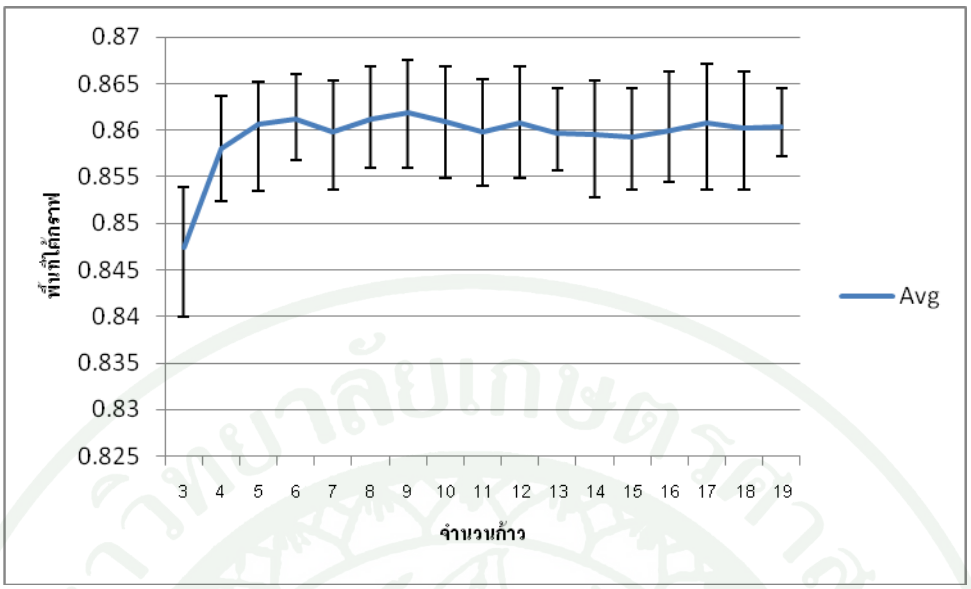
การทดลองปรับค่าซัพพอร์ต (ค่าจำนวนข้อมูลที่ครอบคลุมต่ำสุดของกฎจากข้อมูลชุดฝึกสอนที่สามารถยอมรับได้) ในการทดลองตัวแปรมีค่าตั้งแต่ 0.0008 ถึง 0.0392 โดยจะเพิ่มค่าตัวแปรทีละ 5 เมื่อเปลี่ยนค่าตัวแปร สามารถพิจารณาผลการทดลองได้จากภาพที่ 21 จากผลการทดลองจะเห็นว่าประสิทธิภาพของอัลกอริทึมจะลดลงเมื่อเปลี่ยนค่าตัวแปรจาก 0.0008 เป็น 0.0038 โดยที่หลังจากนั้นประสิทธิภาพจะค่อนข้างคงที่เมื่อเปลี่ยนค่าตัวแปร ซึ่งเงื่อนไขค่าซัพพอร์ตจะส่งผลต่อจำนวนกฎที่ได้รับ หากตั้งไว้มากกฎที่ผ่านเงื่อนไขจะมีจำนวนน้อยแต่กฎหนึ่งกฎที่ได้นั้นจะครอบคลุมข้อมูลในปริมาณมาก ในทางกลับกันหากตั้งไว้น้อยกฎที่ผ่านเงื่อนไขจะมีจำนวนมากแต่กฎหนึ่งกฎที่ได้นั้นจะครอบคลุมข้อมูลในปริมาณมาก จากการทดลองจะเห็นว่ามีจำนวนกฎมากโดยที่แต่ละกฎครอบคลุมข้อมูลจำนวนน้อยจะให้ผลลัพธ์ที่ดีกว่า



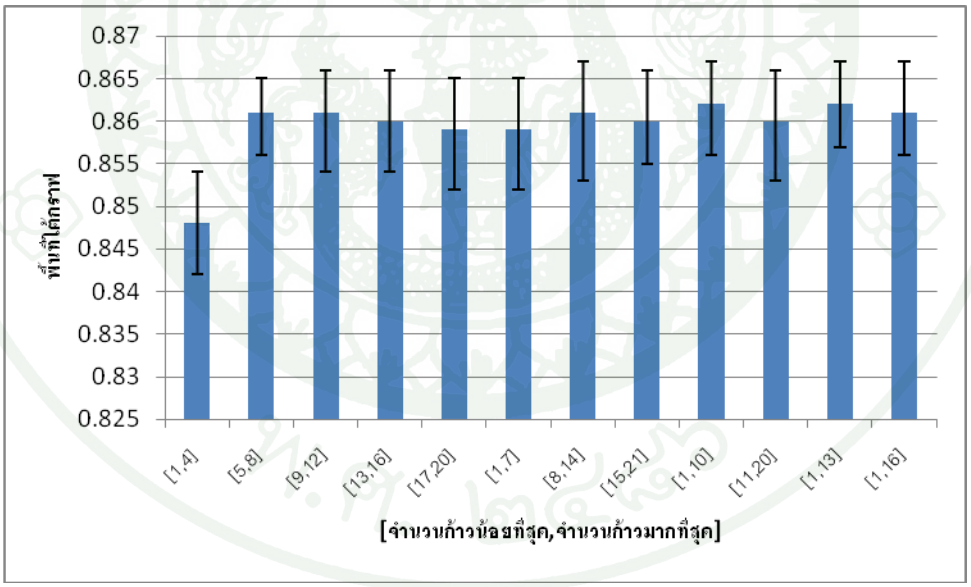
ภาพที่ 21 ผลการทดลองประสิทธิภาพเมื่อปรับเปลี่ยนค่าจำนวนข้อมูลที่ครอบคลุมต่ำสุดของกฎที่สามารถยอมรับได้

การทดลองปรับค่าจำนวนก้าวของมดเทียมเมื่อจุดเริ่มต้นเป็นโฮสต์ปกติ โดยขั้นแรกจะให้จำนวนก้าวที่มากที่สุดกับจำนวนก้าวที่น้อยที่สุดเท่ากันก่อน ในการทดลองตัวแปรมีค่าตั้งแต่ 1 ถึง 19 ก้าว โดยจะเพิ่มค่าตัวแปรทีละ 1 ก้าวเมื่อเปลี่ยนค่าตัวแปร สามารถพิจารณาผลการทดลองได้จากภาพที่ 22 ค่าตัวแปรที่ให้ประสิทธิภาพสูงสุดคือ 9 ก้าว ซึ่งจำนวนก้าวของมดเทียมจะส่งผลต่อความครอบคลุมของกฎ ซึ่งถ้าหากพิจารณาที่จำนวนมดเท่ากันหมายความว่าจำนวนกฎที่ได้จะเท่ากันด้วย ซึ่งเมื่อจำนวนกฎเท่ากัน ถ้าหากแต่ละกฎครอบคลุมข้อมูลน้อยโอกาสที่โฮสต์จากข้อมูลชุดทดสอบไม่สามารถหากฎที่ครอบคลุมได้จะมาก แต่ในขณะเดียวกันถ้าให้มดเทียมมีจำนวนก้าวที่มากโอกาสที่จะเดินไปแล้วเจอโฮสต์สแปมจะมีเพิ่มขึ้น

ขั้นที่สองจะปรับให้จำนวนก้าวที่มากที่สุดกับจำนวนก้าวที่น้อยที่สุดเท่าไม่กัน ค่าตัวแปรและผลการทดลองสามารถดูได้จากภาพที่ 22 ซึ่งค่าตัวแปรนั้นอยู่ในคอลัมน์ขอบเขต โดยจะประกอบไปด้วยตัวเลขสองตัว ตัวเลขตัวแรกแทนจำนวนก้าวน้อยที่สุด และตัวเลขที่สองแทนจำนวนก้าวมากที่สุด ผลการทดลองดูจากค่าเฉลี่ยแล้วจะเห็นว่าไม่ต่างกันมากนักยกเว้นค่าตัวแปร [1,4] ซึ่งก็เป็นไปตามการทดลองขึ้นแรกที่หากมีจำนวนก้าวมากกว่า 4 แล้วจะให้ผลการทดลองไม่ต่างกันมากนัก



ภาพที่ 22 ผลการทดลองเปรียบเทียบประสิทธิภาพเมื่อปรับเปลี่ยนค่าจำนวนก้าวของมดเทียมเมื่อจุดเริ่มต้นเป็นโฮสต์ปกติแบบมดทุกตัวมีจำนวนก้าวเท่ากัน



ภาพที่ 23 ผลการทดลองเปรียบเทียบประสิทธิภาพเมื่อปรับเปลี่ยนค่าจำนวนก้าวของมดเทียมเมื่อจุดเริ่มต้นเป็นโฮสต์ปกติแบบมดทุกตัวมีจำนวนก้าวไม่เท่ากัน

เปรียบเทียบกับ C4.5

การเปรียบประสิทธิภาพของอัลกอริทึม *LSD-ACO* จะเปรียบเทียบกับอัลกอริทึมต้นไม้ตัดสินใจ C4.5 เนื่องจากเป็นอัลกอริทึมในการเรียนรู้ที่ให้ผลลัพธ์ออกมาเป็นกฎเหมือนกัน อีกทั้งยังเป็นที่นิยมใช้ในการเปรียบเทียบจากงานวิจัยทางด้านนี้ ซึ่งในงานวิจัยนี้จะใช้โปรแกรม Weka ในการสร้างต้นไม้ตัดสินใจ C4.5 (Witten และ Frank, 2005)

อัลกอริทึม *LSD-ACO* จะใช้ค่าตัวแปรที่ให้พื้นที่ใต้กราฟมากที่สุดจากการทดลองที่ผ่านมา ส่วนอัลกอริทึม C4.5 จะใช้ค่าตัวแปรเริ่มต้นในโปรแกรม Weka โดยขั้นตอนการทดลองสามารถดูได้จากหัวข้อการทดลอง สามารถพิจารณาผลการทดลองได้จากตารางที่ 2

ตารางที่ 2 แสดงผลการทดลองของอัลกอริทึม LSD-ACO เทียบกับ C4.5

ข้อมูลชุดฝึกสอน	LSD-ACO	C4.5
ชุดที่ 1	0.877	0.851
ชุดที่ 2	0.862	0.831
ชุดที่ 3	0.856	0.845
เฉลี่ย	0.865	0.842

ผลการทดลองจากตารางที่ 2 เมื่อพิจารณาจะพบว่า ค่าเฉลี่ยของพื้นที่ใต้กราฟจากอัลกอริทึม *LSD-ACO* นั้นจะอยู่ที่ 0.865 ส่วนอัลกอริทึม C4.5 จะอยู่ที่ 0.842 ซึ่งจะเห็นได้ว่าอัลกอริทึม *LSD-ACO* นั้นมีประสิทธิภาพในการตรวจจับสแปมที่ดีกว่า ซึ่งเป็นผลมาจากการใช้ประโยชน์โครงสร้างกราฟ

สรุปและข้อเสนอแนะ

สรุป

วิธีการสร้างกฎจากเส้นทางที่ได้จากการเดินของมดเทียมจากงานวิจัยนี้นำเสนอเพื่อนำไปตรวจจับโฮสต์สแปม โดยการคัดแยกโฮสต์สแปมและโฮสต์ปกติออกจากกัน โดยวิธีที่นำเสนอนั้นจะใช้ประโยชน์จากโครงสร้างโฮสต์กราฟ ซึ่งกฎที่ได้มาสามารถนำมาใช้ในการทำนายว่าเป็นโฮสต์ปกติหรือโฮสต์สแปม

จากการทดสอบด้วยชุดข้อมูล Web Spam UK 2006 เมื่อทำการทดลองเปลี่ยนแปลงค่าตัวแปรต่างๆ แล้วจะพบว่า ตัวแปรจำนวนมดเทียมจะส่งผลต่อประสิทธิภาพอย่างเห็นได้ชัดที่สุด หากสังเกตจากผลการทดลองจะพบว่า กราฟจะถูกแบ่งออกเป็นสองช่วงคือช่วงที่เพิ่มจำนวนมดเทียมแล้วประสิทธิภาพจะเพิ่มขึ้นตามไปด้วย กับช่วงที่เพิ่มจำนวนมดเทียมแล้วประสิทธิภาพจะลดลง โดยที่ตัวแปรจำนวนมดจะส่งผลต่อจำนวนกฎที่ได้มาเนื่องจาก มดเทียมหนึ่งตัวจะสร้างมาหนึ่งเส้นทาง หากมีมดเทียมหลายตัวก็จะได้มาหลายเส้นทาง ซึ่งเส้นทางเหล่านั้นจะถูกนำมาแปลงเป็นกฎ ทำให้อัลกอริทึมในส่วนของขั้นตอนการเลือกมีตัวเลือกที่หลากหลายขึ้น แต่จากการทดลองจะเห็นว่าการมีกฎในขั้นตอนการเลือกกฎมากเกินไปนั้นก็จะส่งผลให้ประสิทธิภาพลดลงได้ ส่วนของตัวแปรที่เหลือจะไม่ส่งผลต่อต่อประสิทธิภาพเท่ากับจำนวนมด เริ่มจากตัวแปรจำนวนรอบในการคำนวณนั้น เมื่อจำนวนรอบที่มากจะทำให้บางเส้นทางมีค่าฟีโรโมนที่สูง ส่งผลให้เส้นทางที่ได้แล้วแตกต่างกันนั้นมีจำนวนลดลง เมื่อมีเส้นทางเหมือนกันเพิ่มขึ้นจำนวนของกฎก็จะลดลงตามไปด้วย สำหรับตัวแปรที่เป็นเงื่อนไขของกฎคือตัวแปรค่าความเชื่อมั่นและจำนวนข้อมูลที่ครอบคลุมต่ำสุดที่สามารถยอมรับได้นั้น ถ้าดูจากผลการทดลองจะเห็นว่าตัวแปรค่าความเชื่อมั่นจะส่งผลต่อประสิทธิภาพมากกว่า เพราะว่าส่งผลโดยตรงกับจำนวนของกฎที่ได้รับมา ส่วนการปรับค่าตัวแปรจำนวนข้อมูลที่ครอบคลุมขึ้นนั้นพบว่า แม้จะมีจำนวนกฎที่ได้รับน้อยลงแต่กฎเหล่านั้นจะครบคลุมข้อมูลเป็นจำนวนมาก ซึ่งสามารถชดเชยได้บางส่วน สำหรับตัวแปรจำนวนก้าว เมื่อเพิ่มจำนวนก้าวขึ้นเรื่อยๆ ผลการทดลองจะเห็นว่าประสิทธิภาพได้ลดลงเล็กน้อย ผลเนื่องมาจากยิ่งมดเทียมเดินห่างจากจุดเริ่มต้นมากเท่าไหร่โอกาสยิ่งพบโฮสต์ที่มีคลาสต่างกันออกไปก็จะมากขึ้น สำหรับขั้นตอนสุดท้ายที่นำมาเปรียบเทียบประสิทธิภาพกับอัลกอริทึมต้นไม้ตัดสินใจที่ผลลัพธ์จากการเรียนรู้เป็นกฎเหมือนกัน จากผลการทดลองพบว่ากฎงานวิจัยนี้สามารถคัดแยกโฮสต์ได้ดีกว่าต้นไม้ตัดสินใจ ซึ่งเป็นผลเนื่องมาจากการใช้ประโยชน์จากโครงสร้างโฮสต์กราฟ ซึ่งไม่มีในต้นไม้ตัดสินใจ

ข้อเสนอแนะ

ส่วนประกอบที่งานวิจัยนี้เลือกใช้เป็นงานวิจัยก่อนหน้าที่ได้รับการยอมรับพอสมควร ดังนั้นบางส่วนจึงเป็นงานที่ถูกพัฒนาขึ้นแล้วยกตัวอย่างเช่นการแบ่งช่วงของคุณลักษณะ งานวิจัยที่นำมาใช้จะเป็นการแบ่งช่วงคุณลักษณะจากการพิจารณาทีละคุณลักษณะ ซึ่งบางครั้งคุณลักษณะไม่ได้เป็นอิสระจากกัน เช่นคุณลักษณะจำนวนลิงค์กับคุณลักษณะค่าเพจแร็งค์ เพราะว่าค่าเพจแร็งค์ถูกคำนวณมาจากจำนวนลิงค์ ในงานวิจัยการแบ่งช่วงแบบใหม่แสดงให้เห็นว่าการแบ่งช่วงโดยพิจารณาทีละมากกว่าหนึ่งคุณลักษณะจะให้ประสิทธิภาพที่ดีกว่า

ในเรื่องของการหาเส้นทางเนื่องจากอัลกอริทึมแอนท์โคโลนีออพติไมเซชันใช้เวลาอย่างมากในการหาเส้นทาง เพราะว่าการหาเส้นทางนั้นต้องพึ่งพามดเทียมหลายตัวกับรอบการทำงานจำนวนหนึ่ง ดังนั้นเพื่อลดเวลาการคำนวณและเพิ่มประสิทธิภาพการคัดแยก เราจำเป็นต้องรู้วิธีการหาเส้นทางที่ดีกว่านี้ และในเรื่องจำนวนก้าวที่มดเดินเนื่องจากเป็นการสุ่ม ซึ่งบางเส้นทางสามารถเดินต่อไปได้เพื่อที่กฎจะได้ครอบคลุมข้อมูลเพิ่มขึ้น หากสามารถหาวิธีการตรวจสอบเส้นทางเหล่านี้ได้อัลกอริทึมก็จะมีประสิทธิภาพเพิ่มขึ้น ข้อเสนอแนะอีกจุดคือการเลือกจุดเริ่มต้น ในงานวิจัยนี้เลือกจุดเริ่มต้นจากโฮสต์ที่มีคลาสซึ่งบางครั้งโฮสต์ปกติจำพวกเว็บบอร์ดเมื่อนำมาใช้เป็นจุดเริ่มต้นจะมีโอกาสที่มีลิงค์ไปยังโฮสต์สแปมสูง จึงเสนอแนะว่าควรมีวิธีการเลือกโฮสต์มาเป็นจุดเริ่มต้นด้วย

เอกสารและสิ่งอ้างอิง

Becchetti, L., C. Castillo, D. Donato, S. Leonardi and R. Baeza-Yates. 2006a. Link-based characterization and detection of web spam, pp. 1-8. *In* **Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web**. Seattle, USA.

_____, _____, _____, _____ and _____. 2006b. Using rank propagation and probabilistic counting for link-based spam detection, pp. 84-89. *In* **Proceedings of the Workshop on Web Mining and Web Usage Analysis (WebKDD)**. Philadelphia, Pennsylvania.

_____, _____, _____, R. Baeza-Yates and S. Leonardi. 2008. Link analysis for web spam detection. **ACM Transactions on the Web (TWEB)** 2(1): 1-42.

Brin, S. and L. Page. 1998. The anatomy of a large-scale hypertextual web search engine, pp. 107-117. *In* **Proceedings of the 7th International Conference on the World Wide Web**. Brisbane, Australia.

Castillo, C., D. Donato, L. Becchetti, P. Boldi, S. Leonardi, M. Santini and S. Vigna. 2006. A reference collection for web spam. **ACM SIGIR Forum** 40(2): 11-24.

_____, _____, A. Gionis, V. Murdock and F. Silvestri. 2007. Know your neighbors: Web spam detection using the web topology, pp. 423-430. *In* **Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**. Amsterdam, Netherlands.

Dorigo, M. and L. M. Gambardella. 1997. Ant colony system: A cooperative learning approach to the traveling salesman problem. **IEEE Transactions on Evolutionary Computation** 1(1): 53-66.

_____ and _____. 1997b. Ant colonies for the traveling salesman problem. **BioSystems** 43(2): 73-81.

Fayyad, U.M. and K.B. Irani. 1993. Multi-interval discretization of continuous-valued attributes for classification learning, pp. 1022-1027. *In* **Proc. of the 13th International Joint Conference on Artificial Intelligence.** Chambery, France.

Gyöngyi, Z., H. Garcia-Molina and J. Pedersen. 2004. Combating web spam with TrustRank, pp. 576-587. *In* **Proceedings of the 30th International Conference on Very Large Data Bases.** Toronto, Canada.

_____ and _____. 2005a. Web spam taxonomy, pp. 39-47. *In* **Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web.** Chiba, Japan.

_____ and _____. 2005b. Link Spam Alliances, pp. 517-528. *In* **Proceedings of the 31th International Conference on Very Large Data Bases (VLDB)**. Trondheim, Norway.

Kleinberg, J.M. 1999. Authoritative source in a hyperlinked environment, pp. 604-632. *In* **Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA-98)**. San Francisco, CA.

Krishnan, V. and R. Rashmi. 2006. Web Spam Detection with AntiTrust Rank, pp. 37-40. *In* **Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web**. Seattle, USA**.**

Ntoulas, A., M. Najork, M. Manasse and D. Fetterly. 2006. Detecting spam web pages through content analysis, pp. 83-92. *In* **Proceedings of the 15th International World Wide Web Conference**. Edinburgh, Scotland.

Page, L., S. Brin, R. Motwani and T. Winograd. 1999. The PageRank citation ranking: Bringing order to the web. **Technical Report Stanford InfoLab 29 January 1998.** 17 pages.

Quinlan, J.R. 1993. **C4.5: Programs for Machine Learning**. Morgan Kaufmann, San Mateo, CA.

Witten, I.H. and E. Frank. 2005. **Data mining: Practical machine learning tools and techniques with Java implementations**. 2nd edition. Morgan Kaufmann, San Mateo, CA.

ประวัติการศึกษา และการทำงาน

ชื่อ – นามสกุล	นายอภิชาติ ทวีศิริเวทย์
เกิดวันที่	11 ตุลาคม 2530
สถานที่เกิด	ตำบลมหาชัย อำเภอเมืองฯ จังหวัดสมุทรสาคร
ประวัติการศึกษา	วศ.บ. (วิศวกรรมคอมพิวเตอร์) คณะวิศวกรรมศาสตร์ มหาวิทยาลัยมหิดล
ตำแหน่งงานปัจจุบัน	
สถานที่ทำงานปัจจุบัน	
ผลงานดีเด่นและรางวัลทางวิชาการ	
ทุนการศึกษาที่ได้รับ	ทุนโครงการบัณฑิตศึกษาภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์