



รหัสโครงการ 18p14c0528

MaRCS : การจัดการความน่าเชื่อถือของระบบที่อาศัยข้อมูลจากกลุ่มผู้ใช้

โปรแกรมเพื่องานการพัฒนาด้านวิทยาศาสตร์และเทคโนโลยี

รายงานฉบับสมบูรณ์

เสนอต่อ

ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ

กระทรวงวิทยาศาสตร์และเทคโนโลยี

ได้รับทุนอุดหนุนโครงการวิจัย พัฒนาและวิศวกรรม

โครงการแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย ครั้งที่ 18

ประจำปีงบประมาณ 2558

โดย

ชื่อผู้พัฒนา : นางสาวนภวรรณ ดุษฎีเวทกุล

ชื่อผู้พัฒนา : นายวงศธร ทองถาวร

ชื่ออาจารย์ที่ปรึกษาโครงการ : อาจารย์ ดร.ประภาพร รัตนธารัง

สถาบันการศึกษา : มหาวิทยาลัยธรรมศาสตร์

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงไปด้วยดีด้วยความช่วยเหลือของอาจารย์ ดร. ประภาพร รัตนธำรง อาจารย์ที่ปรึกษาโครงการ ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่าง ๆ อันเป็นประโยชน์อย่างยิ่งในการทำวิจัย อีกทั้งยังช่วยแก้ปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างการทำดำเนินงานอีกด้วย ขอขอบคุณภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยธรรมศาสตร์ที่เอื้อเฟื้ออุปกรณ์ และสถานที่ในการทำโครงการนี้ และสุดท้ายนี้ขอขอบคุณทางศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ ที่มอบทุนอุดหนุนโครงการการแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทยครั้งที่ 18 ให้กับโครงการ MaRCS : การจัดการความน่าเชื่อถือของระบบที่อาศัยข้อมูลจากกลุ่มผู้ใช้

นภวรรณ ดุษฎีเวทกุล

วงศธร ทองถาวร

บทคัดย่อ

ระบบรายงานข้อมูลจากกลุ่มผู้ใช้ (Crowdsourcing) คือ การกระจายปัญหาไปให้คนจำนวนมากช่วยในการแก้ปัญหาในโลกออนไลน์ เพื่อให้ได้มาซึ่งวิธีการที่มีข้อมูลหลากหลายตอบสนองต่อภาวะเปลี่ยนแปลงต่างๆ ต้นทุนต่ำ และเปิดโอกาสให้คนจำนวนมากสามารถเฝ้ามองการเปลี่ยนแปลงของปัญหานั้นได้จนรู้สึกมีส่วนร่วม และผูกพันกับปัญหานั้น เช่น Wikipedia ที่อนุญาตให้ทุกคนร่วมกันแก้ไขข้อมูลในเว็บไซต์ ระบบการให้ Rating ภาพยนตร์ หรือระบบการรายงานภัยพิบัติเป็นระบบที่ต้องอาศัยการรายงานข้อมูลจากกลุ่มคนจำนวนมากเพื่อแสดงผลข้อมูลที่ถูกต้อง เป็นต้น

แต่ปัญหาสำคัญของระบบที่อาศัยข้อมูลจากกลุ่มผู้ใช้จำนวนมากคือความน่าเชื่อถือของข้อมูลที่ผู้ใช้ส่งเข้ามาในระบบ จึงทำให้ข้อมูลจากผู้ใช้อาจมีความผิดพลาดจากความจริงบ้าง เนื่องจากผู้ใช้แต่ละคนมีความเชี่ยวชาญในแต่ละเรื่องไม่เท่ากันรวมไปถึงผู้ที่ตั้งใจเข้ามาก่อวุ่นในระบบ ซึ่งทั้งสองปัจจัยนี้อาจทำให้ระบบแสดงผลผิดพลาด และทำให้เกิดปัญหาอื่นๆตามมา โครงการวิจัยนี้ได้เล็งเห็นปัญหาในเรื่องความน่าเชื่อถือของข้อมูลที่ได้มาจากกลุ่มผู้ใช้ที่มีความหลากหลายนี้ จึงได้ศึกษาวิธีการจัดการความน่าเชื่อถือของผู้ใช้แต่ละคน (Reputation Management) เพื่อคัดกรองข้อมูลจากผู้ที่มีความน่าเชื่อถือมากพอที่นำไปประมวลผลได้ โครงการวิจัยนี้เป็นการจัดการความน่าเชื่อถือของข้อมูลที่ได้จากกลุ่มผู้ใช้โดยมีการรายงานสภาพน้ำท่วมถนนในพื้นที่บริเวณหนึ่งเป็นกรณีศึกษา และใช้วิธีแบบจำลองแบบเอเจนต์ (Agent-based model) ในการจำลองสภาพน้ำท่วมถนนร่วมกับการจำลองการรายงานข้อมูลของกลุ่มผู้ใช้ โดยมีเอเจนต์ (Agent) 3 ประเภทคือ เอเจนต์ผู้ใช้งานระบบ เอเจนต์ปัญหา และเอเจนต์ศูนย์ประมวลผลข้อมูล ซึ่งเอเจนต์ผู้ใช้งานระบบแบ่งออกเป็น 2 ประเภท คือผู้ใช้งานทั่วไป และผู้ใช้งานที่ตั้งใจก่อวุ่นระบบ โดยผู้ใช้งานที่มีพฤติกรรมที่จะรายงานสภาพถนนไปให้ศูนย์ประมวลผลข้อมูล โดยสภาพถนนที่ผู้ใช้งานรายงานจะมี 2 รูปแบบคือ ถนนใช้งานได้ดี และถนนใช้งานไม่ได้ ในส่วนของเอเจนต์ปัญหาจะจำลองระดับน้ำท่วมของถนนแต่ละเส้น โดยอ้างอิงข้อมูลมาจากแหล่งข้อมูลที่เปิดเผย และเอเจนต์ศูนย์ประมวลผลข้อมูลจะคัดกรองความน่าเชื่อถือของผู้ใช้โดยใช้วิธีการลงโทษแบบเบา (Soft Penalty) และวิธีการลงโทษแบบหนัก (Hard Penalty) และรวบรวมข้อมูลโดยวิธีระบบเสียงข้างมาก (Majority Voting) กับวิธีค่าคาดหวังสูงสุด (Expectation Maximization) โดยแบบจำลองนี้จะใช้ในการทดสอบประสิทธิภาพของอัลกอริทึมต่างๆด้วยการศึกษาในแบบจำลอง (Simulation study)

จากผลการทดลองจะสรุปได้ว่าวิธีการรวมข้อมูลแบบ EM Algorithm สามารถประคับประคองความถูกต้องของระบบได้ดีกว่า Majority Voting และส่วนของวิธีการจัดการความน่าเชื่อถือของผู้ใช้ Soft Penalty สามารถคัดกรองข้อมูลและเพิ่มความถูกต้องของระบบได้เพียง 0.5%

Abstact

Crowdsourcing is the process to accomplish work or solution by collecting opinions from citizens. When we ask interesting questions to the public and ask for their opinions, we can acquire various perspectives and information that is adaptable to changing environments with low cost and, at the same time, it can encourage people to observe and contribute the information that could make them feel dedication to the problem solution i.e. as Wikipedia, Movie rating website or Disaster warning system.

One of the main problems of crowdsourcing systems is managing reputation of users. Since different users can have unequal domain skills and some users are malicious that intend to harm the systems, accuracy of system results could decrease and cause other problems.

We realize the significance of this problem, so we are interested in the reputation management of crowdsourcing systems for filtering out the unreliable information from users that have reputation lower than threshold. In this project, we consider flooded road reporting as a case study and use an agent-base model to simulate flooding on roads and reporting messages by users. There are 3 types of agent: user agent, problem agent and datacenter agent. The user agent is divided in 2 types: normal user and adversary user. User agents report road status, i.e., usable, risk to use or not usable, to the datacenter agent. The problem agent generates a flood level on a road using collected data from Open data. The datacenter agent manages reputation of users by using soft penalty and hard penalty algorithms and aggregates user data using majority voting and expectation maximization. The performance of these algorithms are examined using simulation study.

As a result, EM algorithm is more stable than Majority Voting in data fusion. Soft Penalty reputation algorithm improves the correctness of system only 0.5% by filter out untrustworthy data

Keywords

- Reputation Management
- Data Fusion
- Trust Management
- Crowdsourcing
- Agent-Based Simulation
- Simulation

บทนำ

ในปัจจุบันจากการรับและส่งข้อมูลในอินเทอร์เน็ตที่มากขึ้น ทำให้เกิดระบบในรูปแบบใหม่ หนึ่งในนั้นคือ ระบบที่อาศัยข้อมูลจากกลุ่มผู้ใช้ (Crowdsourcing System) เป็นระบบที่รับข้อมูลจากการรายงานของผู้ใช้ และนำมาวิเคราะห์ผลออกมาในรูปแบบต่างๆ เช่น การให้คะแนน Rating ของภาพยนตร์ แต่ปัญหาสำคัญของระบบที่อาศัยข้อมูลจากกลุ่มผู้ใช้ คือความน่าเชื่อถือของผู้ใช้ที่รายงานผลเข้ามาในระบบเราไม่สามารถตรวจสอบข้อมูลที่ส่งเข้ามาในระบบว่าถูกต้องหรือไม่ ซึ่งสาเหตุอาจมาจากผู้ใช้แต่ละคนมีความเชี่ยวชาญเกี่ยวกับปัญหาที่กำหนดไว้ไม่เท่ากัน รวมไปถึงผู้ใช้ที่ตั้งใจเข้ามาโกงในระบบ ซึ่งทั้งสองปัจจัยนี้อาจทำให้ระบบประมวผลผิดพลาด และทำให้เกิดปัญหาอื่นๆตามมา โครงการงานวิจัยนี้ได้เล็งเห็นปัญหาในเรื่องความน่าเชื่อถือของข้อมูลที่ได้มากจากกลุ่มผู้ใช้ที่มีความหลากหลายนี้ จึงได้ศึกษาวิธีการจัดการความน่าเชื่อถือของผู้ใช้ (Reputation Management) เพื่อคัดกรองข้อมูลจากผู้ใช้ที่มีความน่าเชื่อถือไม่มากพอกจากการประมวผล คณะผู้จัดทำโครงการจะใช้ระบบจำลองในการศึกษาวิธีการจัดการความน่าเชื่อถือ โดยต้องการค้นหาอัลกอริทึมที่สามารถจัดการความน่าเชื่อถือของผู้ใช้ และเปรียบเทียบข้อดีข้อเสียของอัลกอริทึมในสถานการณ์ต่างๆ สาเหตุที่ใช้ระบบจำลองเพื่อสามารถทดสอบอัลกอริทึม โดยกำหนดสถานการณ์ได้หลายรูปแบบ และไม่จำเป็นต้องหาผู้ใช้งานจำนวนมากมาทำการทดสอบระบบ โดยทางคณะผู้จัดทำโครงการได้เลือกการจำลองสภาพน้ำท่วมบริเวณใดๆเป็นกรณีศึกษา เพื่อเป็นตัวอย่างให้เห็นว่าระบบ Crowdsourcing และระบบจัดการความน่าเชื่อถือสามารถนำไปใช้ประโยชน์ในชีวิตจริงได้

สารบัญ

กิตติกรรมประกาศ	ก
บทคัดย่อ.....	ข
Abstract.....	ค
Keywords	ง
บทนำ	จ
สารบัญ	ฉ
วัตถุประสงค์และเป้าหมาย	1
รายละเอียดของการพัฒนา.....	2
ตัวอย่างโปรแกรม หรือ ผลงานที่สื่อให้เห็นผลงานที่พัฒนาขึ้น	2
ทฤษฎีหลักการและเทคนิคหรือเทคโนโลยีที่ใช้ เช่น เทคโนโลยีด้านปัญญาประดิษฐ์ Algorithms ที่ใช้โครงสร้างข้อมูล เป็นต้น โดยผู้พัฒนาต้องให้รายละเอียดที่เกี่ยวข้องประกอบด้วย.....	9
เครื่องมือที่ใช้ในการพัฒนา ได้แก่ ภาษาที่ใช้เขียน Tools อื่นๆ ที่ช่วยในการพัฒนาโปรแกรม และอื่นๆ.	10
รายละเอียดโปรแกรมที่ได้พัฒนาในเชิงเทคนิค (Software Specification)	10
Input/Output Specification.....	10
Functional Specification	11
โครงสร้างของซอฟต์แวร์ (Design)	11
อื่นๆ (ผู้พัฒนาต้องชี้แจงส่วนสำคัญที่ทีมงาน/ผู้พัฒนาได้พัฒนาขึ้นเองรวมทั้งต้องระบุแหล่งที่มาของโปรแกรม หรือ Source Code อื่นที่มาประกอบในโปรแกรมไว้ด้วย โดยไม่ต้องจัดพิมพ์ Source Code แนบมา)	13
ขอบเขตและข้อจำกัดของโปรแกรมที่พัฒนา	14
กลุ่มผู้ใช้โปรแกรม.....	15
ผลของการทดสอบโปรแกรม	16
ปัญหาและอุปสรรค.....	26
แนวทางในการพัฒนาและประยุกต์ใช้ร่วมกับงานอื่นๆ ในขั้นต่อไป	27

ข้อสรุปและข้อเสนอแนะ	28
เอกสารอ้างอิง (Reference)	29
ข้อมูลของผู้พัฒนา	30
ภาคผนวก	31

วัตถุประสงค์และเป้าหมาย

- 1) ศึกษาวิธีการจัดการความน่าเชื่อถือของผู้ใช้แต่ละคน (Reputation Management) เพื่อคัดกรองข้อมูลจากผู้ที่มีความน่าเชื่อถือไม่เพียงพอออกจากระบบ
- 2) ออกแบบและพัฒนาแบบจำลองเอเจนต์ (Agent-based Simulation) เพื่อใช้ศึกษาระบบ Crowdsourcing ของรายงานข้อมูลจากกลุ่มผู้ใช้จำนวนมาก
- 3) ศึกษาเปรียบเทียบประสิทธิภาพของวิธีการจัดการความน่าเชื่อถือของผู้ใช้ โดยใช้สถานการณ์น้ำท่วมบริเวณพื้นที่ที่กำหนด จากข้อมูลที่เปิดให้ใช้อย่างอิสระ
- 4) แสดงผลข้อมูลที่ได้จากการประมวลผลภายในระบบ ในรูปแบบของแผนที่

รายละเอียดของการพัฒนา

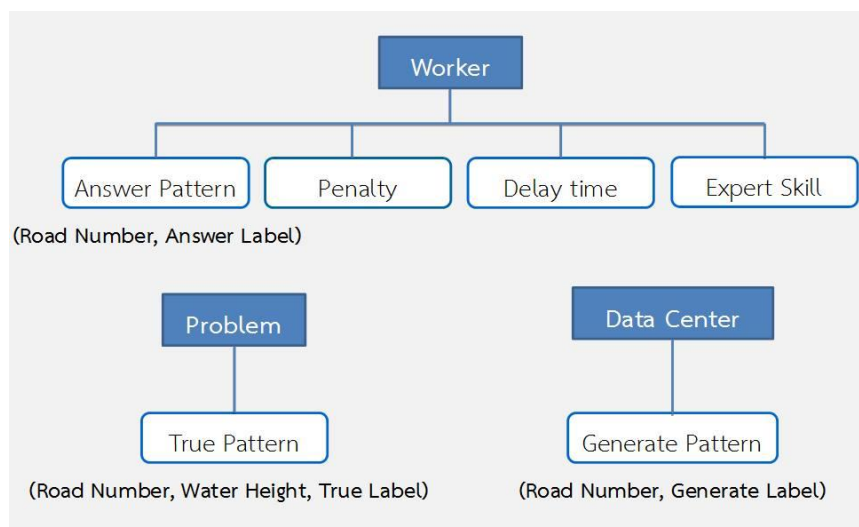
ตัวอย่างโปรแกรม หรือ ผลงานที่สื่อให้เห็นผลงานที่พัฒนาขึ้น

1. ภาพรวมการทำงานในระบบ

Agent types:

- Worker (Normals and Adversary) Agent
- Problem Agent
- Data center Agent

Agent properties:



Worker :

- Penalty
- Answer-pattern (Road-number ,answer-label)
- Delay-ans time
- Professional skill

Problem :

- Real-pattern (Road-number ,True-label)
- ระดับน้ำ (Random)
- เวลาที่น้ำจะลด (เป็นผลมาจากระดับน้ำ)

Data center :

- Generate-pattern (Road-number ,Generate-label)

Agent behaviors:

Worker:

- Worker รายงานภาพถนนไปให้ Data center

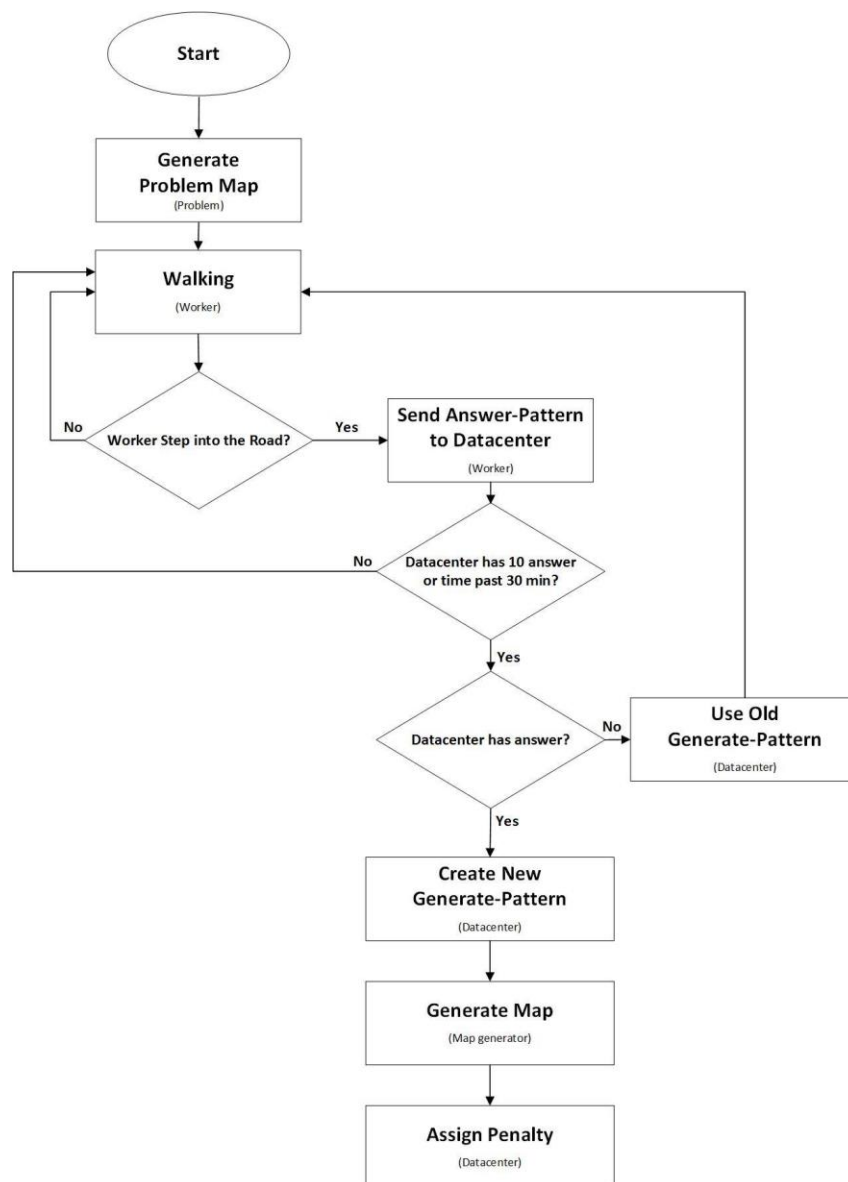
Problem :

- สุ่ม Problem โดยจะสุ่มค่าน้ำว่าจะท่วมถนนหรือไม่ หากท่วมจะท่วมนานกี่นาที และจะเก็บเวลาที่น้ำจะลด(ได้มากจากระดับน้ำที่ท่วม)ไว้เพื่อจะได้สุ่ม Problem รอบต่อไป
- Problem จะส่ง Real-answer ไปให้ Datacenter หลังจากการสุ่ม Problem

Data center :

- รับ Answer-label จาก Worker
- นำ Answer-label มาประมวลผลเป็น Generate-pattern ส่งไปให้ Map Generator
- หลังจากคำนวณความแม่นยำของคำตอบเรียบร้อยแล้ว จะส่งค่า Penalty กลับไปที่ Worker ทุกคน

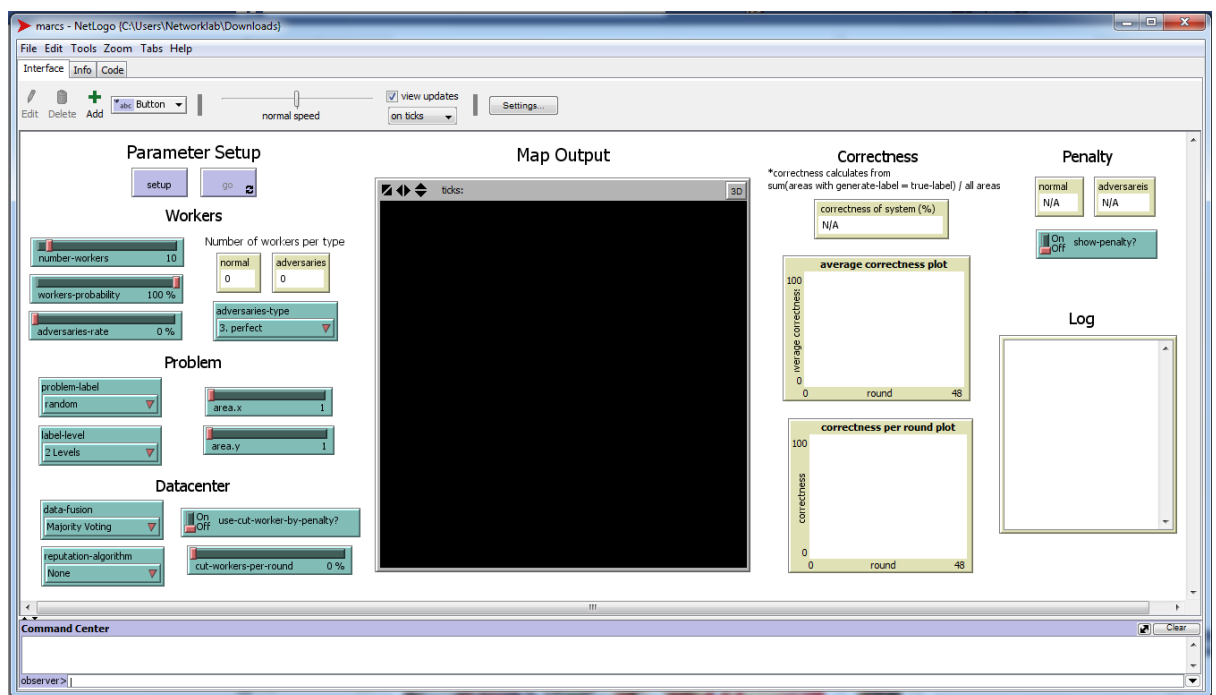
Time steps:



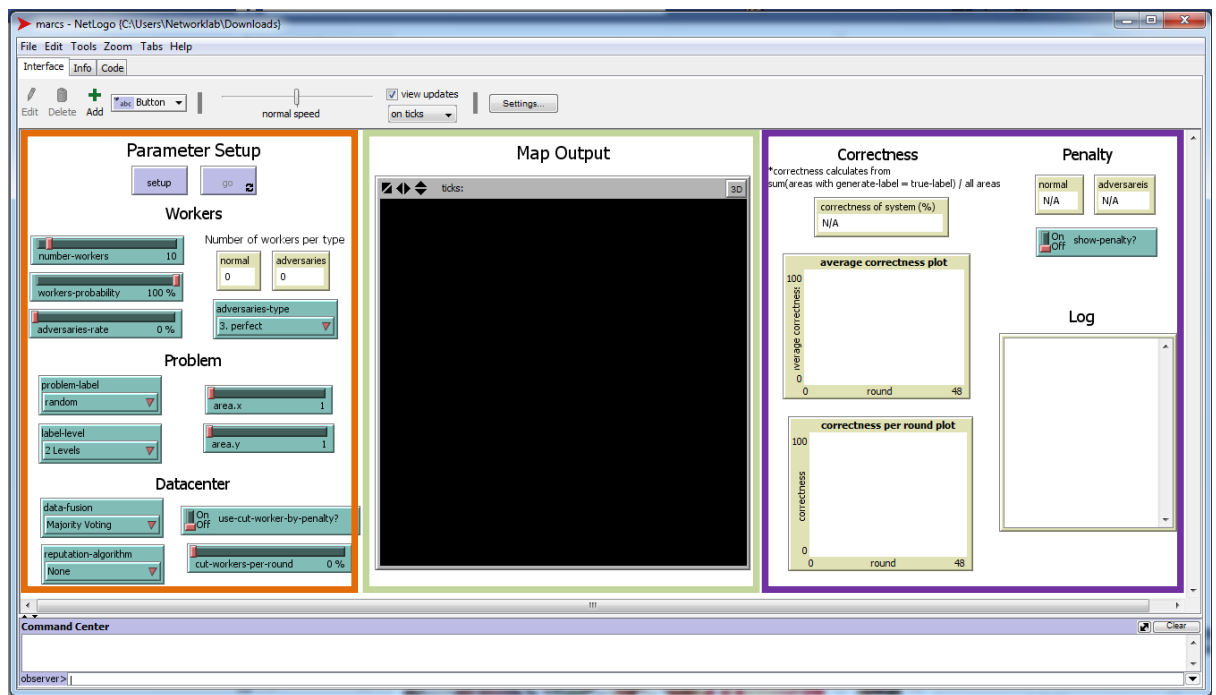
- 1) สุ่ม Problem โดยจะสุ่มค่าน้ำว่าจะท่วมถนนหรือไม่ หากท่วมจะท่วมนานกี่นาที และจะเก็บเวลา

- ที่น้ำจะลด(ได้มาจากระดับน้ำที่ท่วม) และ Problem จะส่ง Real-pattern ไปให้ Datacenter
- 2) Worker เดินไปอย่างอิสระ ถ้า Worker เดินเจอ Problem ให้ Worker ดู และ รายงานสถานะของพื้นที่ไปให้ Data center แบบ Real-time
 - 3) หลังจากที่ Worker รายงานสถานะของพื้นที่ Worker จะไม่สามารถรายงานสถานะของถนนเส้นนั้นจนกว่าจะครบ 60 นาที หรือสถานะของพื้นที่เปลี่ยนแปลงไป
 - 4) Data center รับ Answer-pattern จาก Worker
 - 5) Data center นำ Answer-pattern มาประมวลผลเมื่อมีผู้ใช้ส่งมา 10 คน หรือประมวลผลทุก 30 นาที ขึ้นอยู่กับเหตุการณ์ใดเกิดขึ้นก่อน (เวลาและจำนวนคนอาจมีการเปลี่ยนแปลง) แต่หากขณะใดขณะหนึ่งมีผู้ใช้ส่งข้อมูลมาพร้อมๆกันจำนวนมากกว่า 10 คน ระบบก็จะนำข้อมูลทั้งหมดมาประมวลผล
 - 6) หากครบ 30 นาทีแล้วยังไม่มีข้อมูลใดเข้ามาในระบบ ก็ใช้ Generate-pattern ครึ่งก่อนมาเป็น Generate-pattern ครึ่งปัจจุบัน และส่งข้อมูลไปให้ Nap Generator แสดงผลในรูปแบบของแผนที่
 - 7) หลังจากคำนวณความแม่นยำของคำตอบเรียบร้อยแล้ว จะส่งค่า Penalty กลับไปที่ผู้ใช้ทุกคน

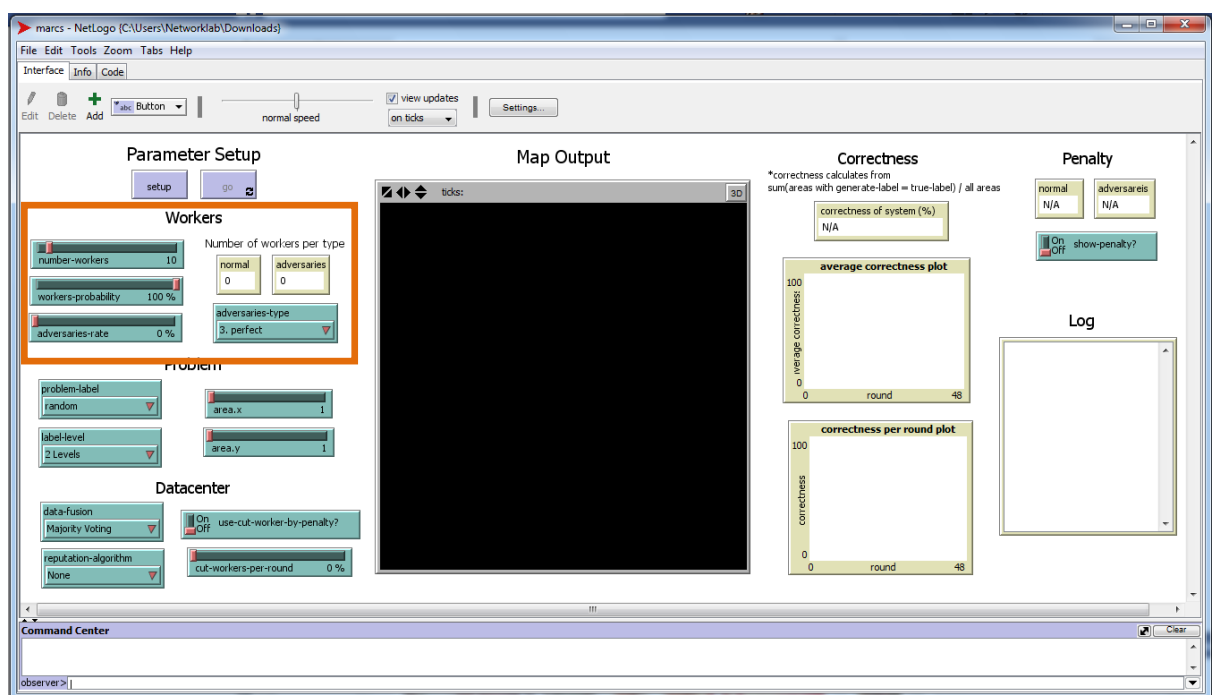
2. ตัวอย่างโปรแกรม



- 1) เมื่อเปิด source code จะพบกับหน้าจอ ซึ่งด้านซ้าย(กรอบสีส้ม) จะเป็นในส่วนตั้งค่า Parameter หน้าจอตรงกลาง(กรอบสีเขียว) เป็นส่วนที่แสดงให้เห็น Simulation และด้านขวา(กรอบสีม่วง) เป็นส่วนของการประมวลออกมาในรูปแบบต่างๆ

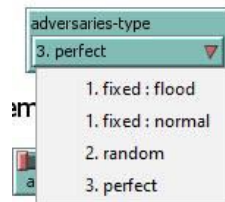


2) ตั้งค่าในส่วนของ Worker Parameter

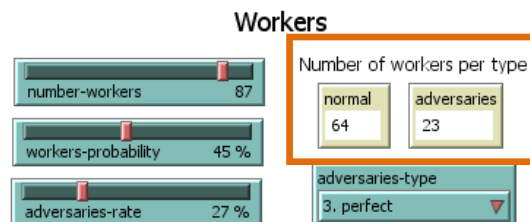


- number-worker คือส่วนกำหนดจำนวน Worker ในระบบเป็นไปได้ตั้งแต่ 1-100 คน
- worker-probability คือเป็นส่วนที่จะระบุ ว่า Normal Worker มีโอกาสที่จะตอบถูกต้อง เป็นกี่เปอร์เซ็นต์
- adversaries-rate คือส่วนที่กำหนดว่าจาก Worker ในระบบที่มีทั้งหมด จะเป็น Adversary Worker ทั้งหมดกี่เปอร์เซ็นต์
- adversaries-type คือการกำหนดประเภทของ Adversary Worker สามารถเลือกได้

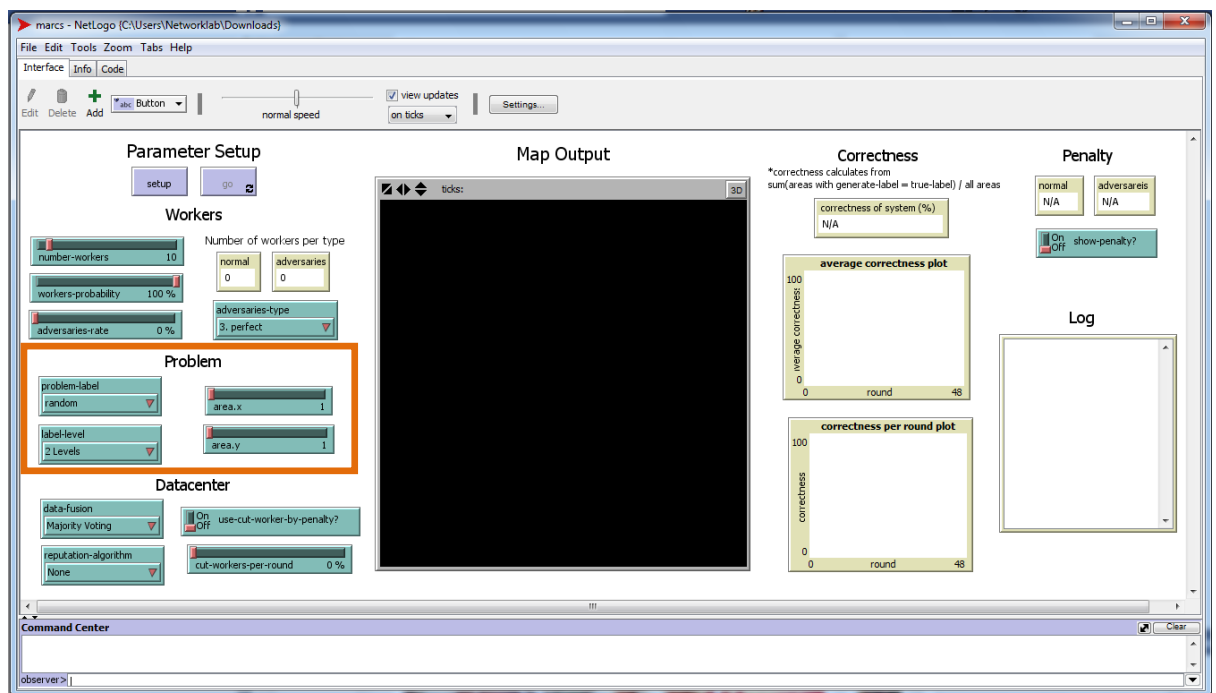
ทั้งหมด 4 ประเภท คือ fixed : flood(ก่อวินโดยรายงานสถานะถนนไม่สามารถใช้งานได้ตลอดเวลา) ,fixed : normal (ก่อวินโดยรายงานสถานะถนนสามารถใช้งานได้ตลอดเวลา),random (ก่อวินโดยรายงานสถานะถนนโดยการสุ่ม) และperfect(ก่อวินโดยรายงานสถานะถนนตรงข้ามกับความจริงตลอดเวลา)



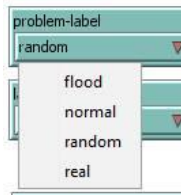
เมื่อตั้งค่าเรียบร้อยแล้ว และกด Set-up จะขึ้นจำนวนของ Worker แต่ละประเภทในช่อง ดังนี้



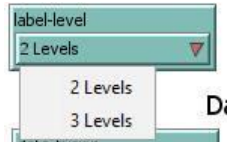
3) ตั้งค่าในส่วน Problem Parameter



- problem-label คือ ส่วนลักษณะปัญหาของ Problem มี 4 ประเภท คือ Flood(Problem มีสถานะน้ำท่วมตลอดเวลา) ,Normal(Problem มีสถานะปกติตลอดเวลา) ,Random(Problem มีสถานะแบบสุ่มตลอดเวลา) และReal(Problem มีสถานะมาจากข้อมูลจริง)

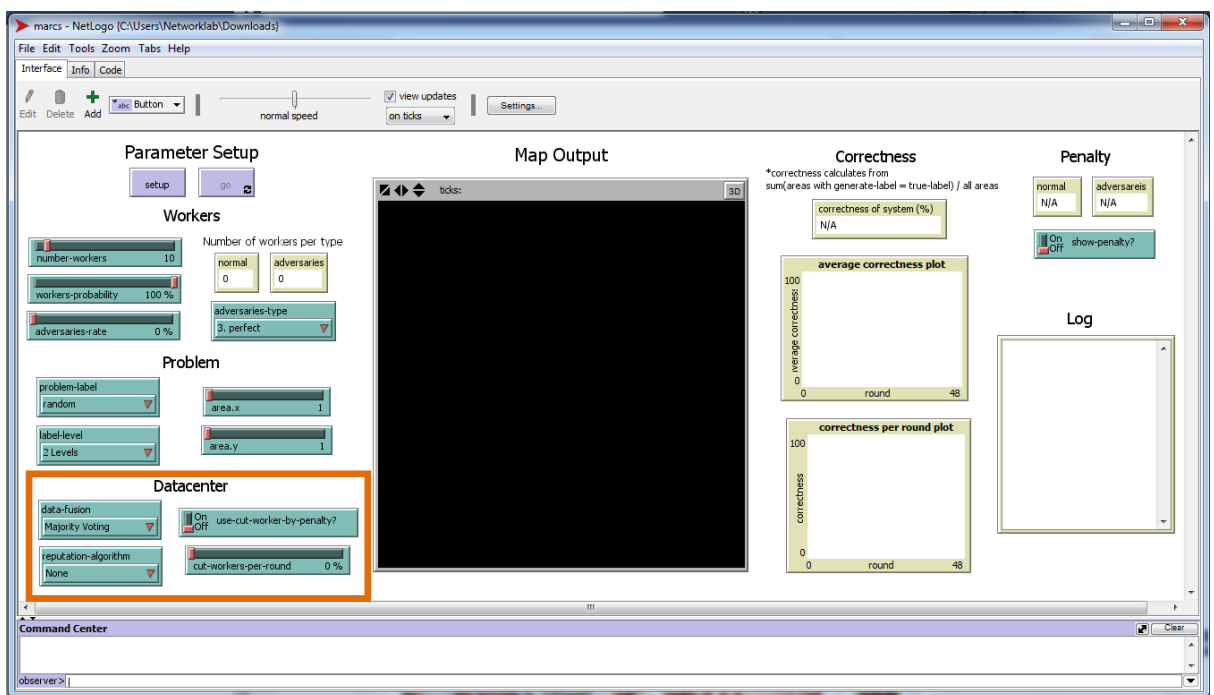


- label-level คือ ในส่วนของ Problem จะมีสถานะที่เป็นไปได้กี่สถานะ มี 2 ประเภทด้วยกัน คือ 2 Level(ถนนสามารถใช้งานได้ และถนนไม่สามารถใช้งานได้) และ 3 Level (ถนนสามารถใช้งานได้ ถนนพอใช้งานได้ และถนนไม่สามารถใช้งานได้)

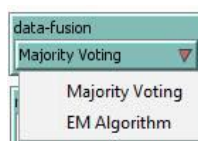


- area คือการกำหนดจำนวนพื้นที่ของ Problem ในการรายงานข้อมูล
area.x คือ การกำหนดด้านย่อยของด้าน x มีค่าตั้งแต่ 1 ถึง 20
area.y คือ การกำหนดด้านย่อยของด้าน y มีค่าตั้งแต่ 1 ถึง 20

4) ตั้งค่าในส่วน Datacenter Parameter

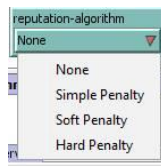


- Data-fusion คือส่วนของการรวบรวมข้อมูล มี 2 ประเภท คือ Majority Voting (วิธีระบบเสียงข้างมาก) และ EM Algorithm (วิธีค่าคาดหวังสูงสุด)



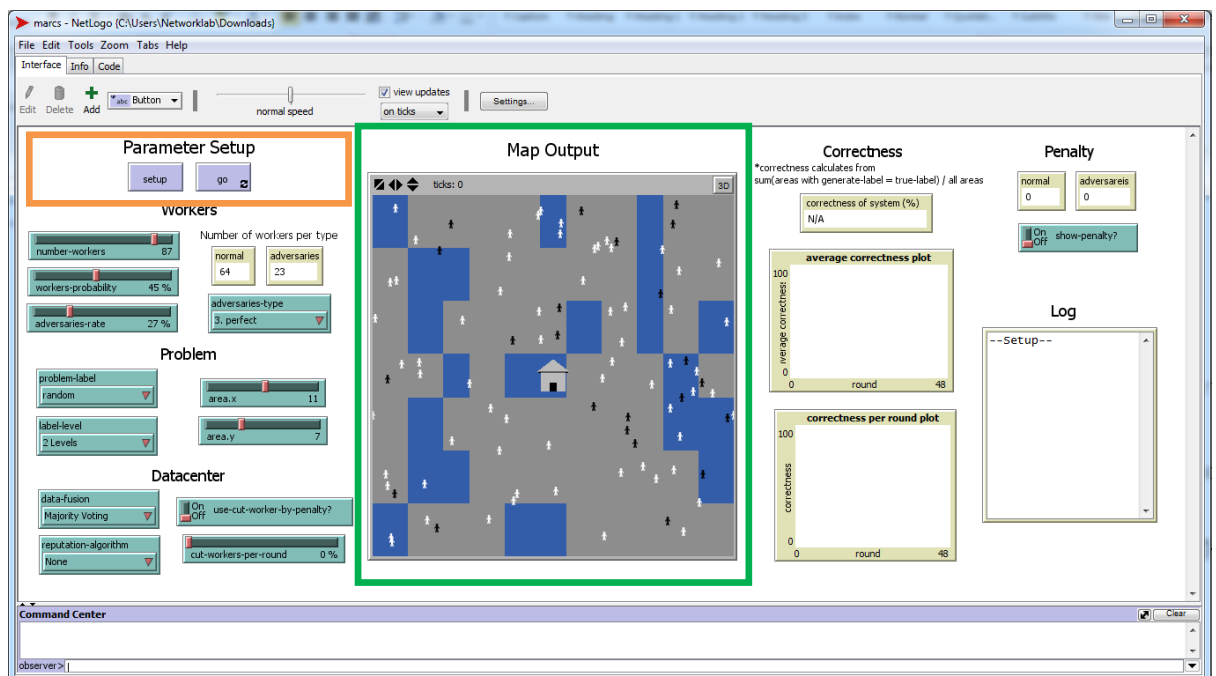
- Reputation-management คือ ส่วนที่จัดการความน่าเชื่อถือของผู้ใช้ มี 4 ประเภท คือ None(ไม่มี Reputation Management) ,Simple Penalty (วิธีการลงโทษอย่างง่าย) ,Soft

Penalty (วิธีการลงโทษแบบเบา) และ Hard Penalty (วิธีการลงโทษแบบหนัก)

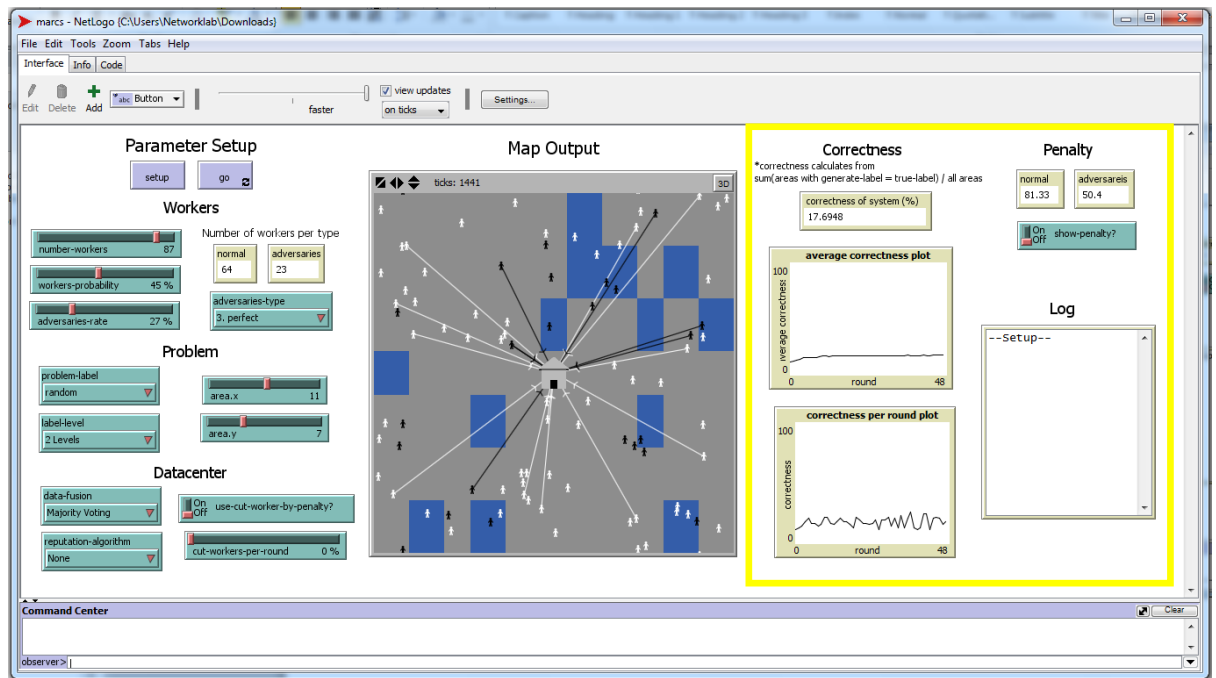


- Use-cut-worker-by-penalty? คือ ต้องการตัด Worker ที่ไม่มีความน่าเชื่อถือออกจากระบบหรือไม่
- cut-worker-per-round คือ เปอร์เซ็นที่ต้องการตัด Worker ที่ไม่มีความน่าเชื่อถือออกจากระบบต่อรอบ

5) หลังจากที่ตั้งค่า Parameter ต่างๆเรียบร้อยแล้ว ให้กด Setup (กรอบสีส้ม) เพื่อให้ Worker และ Problem สุ่มตำแหน่งใน Map Output(กรอบสีเขียว) และกด Go (กรอบสีส้ม) เพื่อเริ่มทำการ Simulation



6) ในส่วนกรอบสีเหลืองจะเป็นส่วนของการแสดงผลหลังจากการประมวลผลของระบบ



- Correctness of System คือ ค่าความถูกต้องของระบบ โดยความถูกต้องคำนวณจากข้อมูล น้ำท่วมที่แสดงผลออกมาจากระบบมีค่าตรงกับสภาพน้ำท่วมจริงในแต่ละพื้นที่ และคำนวณ ค่าเฉลี่ยความถูกต้องของระบบจากพื้นที่ทั้งหมด
- Average Correctness Plot คือ กราฟแสดงผลค่าเฉลี่ยความถูกต้องของระบบเมื่อเวลาผ่านไป
- Correctness Plot คือ กราฟแสดงผลความถูกต้องของระบบในแต่ละรอบของการประมวลผลข้อมูล
- Penalty คือ ใช้ตรวจสอบค่าความไม่น่าเชื่อถือของผู้ใช้ว่าระบบสามารถให้ค่า Penalty ได้ ถูกต้องกับประเภทของ Worker หรือไม่ ถ้าวิธีการให้ค่า Penalty นั้นมีประสิทธิภาพ Normal Worker จะต้องมียค่า Penalty ที่ต่ำกว่า Adversaries
- Show-Penalty? คือ ต้องการแสดงผล Penalty ของ Worker ทุกคนหรือไม่
- Log คือ แสดงผล Flow การทำงานของระบบ

ทฤษฎีหลักการและเทคนิคหรือเทคโนโลยีที่ใช้ เช่น เทคโนโลยีด้านปัญญาประดิษฐ์ Algorithms ที่ใช้ โครงสร้างข้อมูล เป็นต้น โดยผู้พัฒนาต้องให้รายละเอียดที่เกี่ยวข้องประกอบด้วย

1. Data Fusion

1.1 วิธีระบบเสียงข้างมาก (Majority Voting)

เป็นวิธีการจัดการความน่าเชื่อถือของข้อมูลที่ส่งมาเข้ามาในระบบ โดยระบบจะนำ ข้อมูลที่ได้จากเสียงข้างมากมาเป็นคำตอบให้กับระบบ แต่ถ้าทั้งสองค่าเท่ากันจะใช้องค์ประกอบ

อื่นในการดู เช่น Penalty

1.2 วิธีค่าคาดหวังสูงสุด (EM Algorithm)

เป็นอัลกอริทึมที่จัดการแบบเป็นรอบๆ โดยมี 2 ขั้นตอนคือ Maximum Likelihood กับ Expectation Maximization โดยอัลกอริทึม นี้เป็นการหาโอกาสสูงสุดทางสถิติที่จะทำให้เกิดเหตุการณ์ที่สนใจจากนำข้อมูลจำนวนหนึ่งนำมาท ้าซ้ำไปเรื่อยจนค่าที่ออกมาคงที่ (Convergence)

2. Reputation management

2.1.1 วิธีการลงโทษแบบเบา (Soft Penalty)

2.1.2 วิธีการลงโทษแบบหนัก (Hard Penalty)

เครื่องมือที่ใช้ในการพัฒนา ได้แก่ ภาษาที่ใช้เขียน Tools อื่นๆ ที่ใช้ช่วยในการพัฒนาโปรแกรม และอื่นๆ

1. เครื่องมือที่ใช้ในการพัฒนาโปรแกรม

1.1 NetLogo

1.2 RStudio

2. ภาษาที่ใช้ในการพัฒนาโปรแกรม

2.1 NetLogo Programming Language

2.2 R Programming Language

รายละเอียดโปรแกรมที่ได้พัฒนาในเชิงเทคนิค (Software Specification)

Input/Output Specification

1. Input ของโปรแกรม

- 1) จำนวนผู้ใช้ในระบบ
- 2) ประเภทของผู้ก่อกรวน
- 3) เพอร์เซ็นต์ของผู้ก่อกรวนระบบ
- 4) ค่าเฉลี่ยความชำนาญของผู้ใช้ระบบแบบปกติ
- 5) อัลกอริทึมที่ใช้ในการจัดการผู้ใช้ในระบบ
- 6) อัลกอริทึมที่ใช้ในการจัดการข้อมูลที่ระบบนำมาแสดงผล
- 7) ลักษณะของปัญหา
- 8) จำนวนสถานะของปัญหาที่จะทำการจำลอง
- 9) ต้องการตัดผู้ใช้ออกจากระบบหรือไม่
- 10) เพอร์เซ็นต์ผู้ใช้ที่ต้องการตัดออกในแต่ละรอบ
- 11) แบ่งจำนวนช่องแกน x
- 12) แบ่งจำนวนช่องแกน y

2. Output ของโปรแกรม

- 1) เปอร์เซ็นความแม่นยำระหว่างสภาพถนนจริง กับคำตอบที่ได้มาจากระบบ
- 2) แผนที่แสดงสภาพถนน

Functional Specification

- 1) จำลองการรายงานสภาพถนนของกลุ่มผู้ใช้
- 2) จัดการความน่าเชื่อถือของผู้ใช้ในระบบ
- 3) ประมวลผลคำตอบจากกลุ่มผู้ใช้
- 4) แสดงผลลัพธ์ออกมาในรูปแบบที่จริง

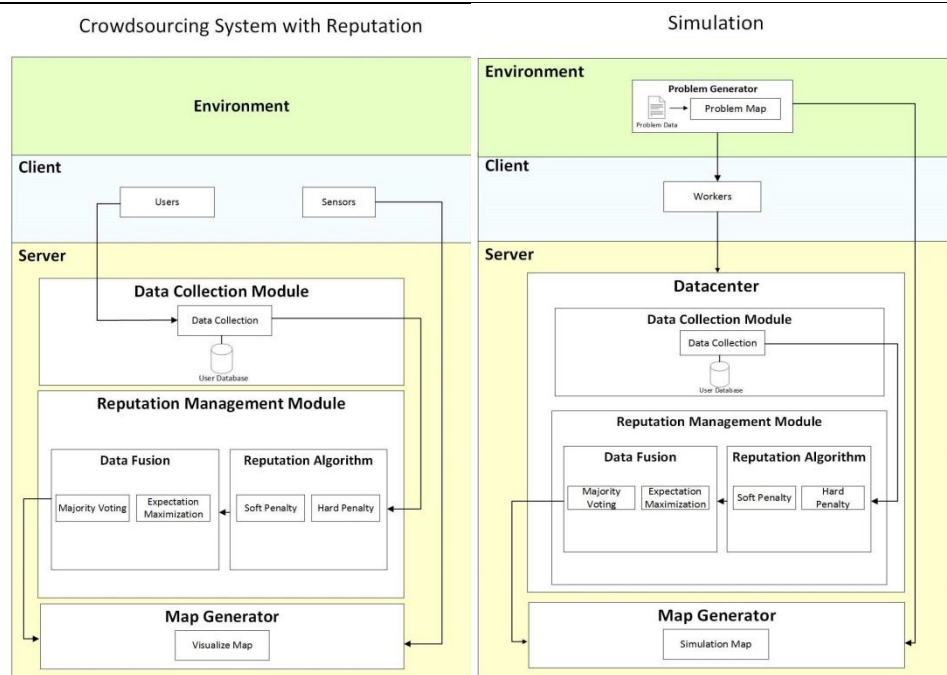
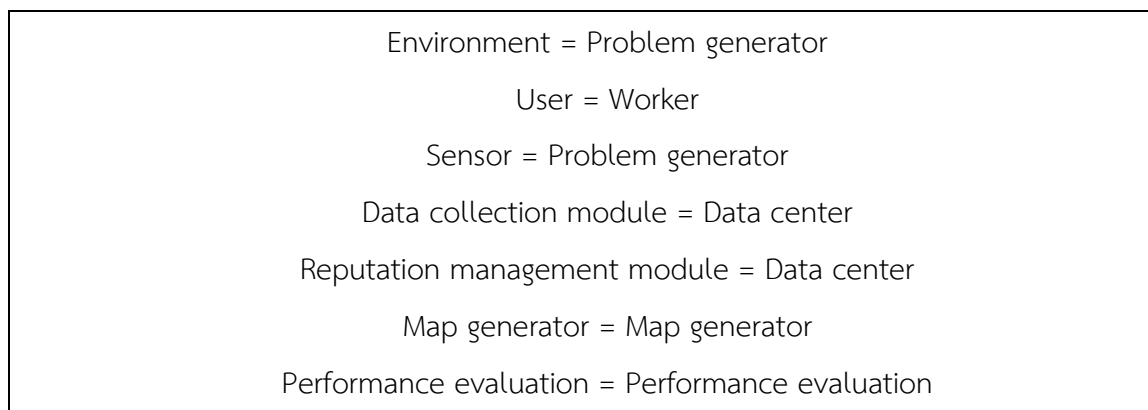
โครงสร้างของซอฟต์แวร์ (Design)

ในโครงการนี้จะสร้างระบบจำลองเพื่อจัดการระบบที่อาศัยข้อมูลจากกลุ่มผู้ใช้ โดยใช้ระบบจำลองสถานการณ์น้ำบริเวณพื้นที่ต่างๆเป็นกรณีศึกษา โดยระบบจะมีสถาปัตยกรรม 2 รูปแบบคือ สถาปัตยกรรมของระบบรายงานข้อมูลจากกลุ่มผู้ใช้จริง และสถาปัตยกรรมของระบบจำลองการรายงานสถานการณ์ ซึ่งระบบหลังได้อ้างอิงการทำงานจากระบบที่มีอยู่จริง ระบบที่รับข้อมูลจากการจำลองสถานการณ์(ภาพด้านขวา) และผู้ใช้จริง(ภาพด้านซ้าย) ประกอบด้วย 3 layer คือ Environment Client และ Server ในส่วน Environment คือ สภาพแวดล้อมของเหตุการณ์ที่จะเกิดขึ้นจริง ส่วนของ Client คือส่วนที่ติดต่อ และส่งข้อมูลเข้ามาในระบบประกอบด้วย ผู้ใช้(User) กับค่าจาก เซนเซอร์(Sensor) และในส่วนของ Server เป็นส่วนที่ทำหน้าที่ประมวลผลข้อมูลทั้งหมด และนำผลที่ได้จากการประมวลผลมาสร้างเป็นแผนที่จริง ในส่วนของ Server แบ่งออกเป็น 4 Module ย่อยด้วยกันคือ Data collection module, Reputation management module, Map generator module และ Evaluation module ในData collection module ทำหน้าที่เก็บรวบรวมข้อมูลจาก Client มาเก็บใน User database โดยมี Data collection เป็นตัวกลางในการติดต่อกับฐานข้อมูล ด้าน Reputation management module มีหน้าที่หลักคือการจัดการข้อมูลโดยผ่านอัลกอริทึมใดๆเพื่อให้ได้ผลลัพธ์เพียง 1 ค่าออกมา และทำหน้าที่จัดการความน่าเชื่อถือของผู้ใช้โดยการให้ Penalty กับผู้ใช้ในระบบ ส่วน Map generator module ทำหน้าที่คือสร้างแผนที่จากผลลัพธ์ที่ออกมาจาก Reputation management module และในส่วนของ Module สุดท้าย Evaluation module ทำหน้าที่คือการเก็บค่าจากเซนเซอร์ และเก็บค่าจากข้อมูลที่ ออกมาจาก Reputation management module ในรูปแบบของไฟล์ .csv เพื่อใช้ในการวัดประสิทธิภาพของอัลกอริทึม

ระบบจำลองสถานการณ์ และจำลองพฤติกรรมของผู้ใช้(ภาพด้านขวา) โดยจำลององค์ประกอบเหมือนกับข้อ 1 คือ Environment Client และ Server ในส่วน Environment คือ สภาพแวดล้อมของเหตุการณ์ที่จะเกิดขึ้นจากการนำข้อมูลจริงมาผ่านการกระบวนการต่างๆ เช่น การนำข้อมูลจริงไปผ่าน Distribution เพื่อทำ Distribution มาทำการสุ่มชุดข้อมูลออกมาจำลอง ส่วนของ Client คือส่วนที่ติดต่อและส่งข้อมูลเข้ามาในระบบประกอบด้วย ผู้ใช้(Worker) และในส่วนของ Server เป็นส่วนที่ทำหน้าที่ประมวล

ข้อมูลทั้งหมด และนำผลที่ได้จากการประมวลผลมาสร้างเป็นแผนที่ใน ส่วนของ Server แบ่งออกเป็น 4 Module ย่อยด้วยกันคือ Data collection module, Reputation management module, Map generator module และ Evaluation module ใน Environment จะมี Map generator module เพื่อแสดงผลเหตุการณ์ออกมาในรูปแบบของแผนที่อย่างง่ายที่จำลองมาจากข้อมูลจริง ใน Data center module แบ่งออกเป็น 2 ส่วนย่อยคือ Data collection Module ย่อยทำหน้าที่เก็บรวบรวมข้อมูลจาก Client มาเก็บใน User database โดยมี Data collection เป็นตัวกลางในการติดต่อกับฐานข้อมูล ด้าน Reputation management module ย่อย มีหน้าที่หลักคือ การจัดการข้อมูลโดยผ่านอัลกอริทึมใดๆเพื่อให้ได้ผลลัพธ์เพียง 1 ค่าออกมา และทำหน้าที่จัดการความน่าเชื่อถือของผู้ใช้โดยการให้ Penalty กับผู้ใช้ในระบบ ส่วน Map generator module ทำหน้าที่คือสร้างแผนที่จากข้อมูลที่ออกมาจาก Reputation management module และในส่วน Module สุดท้าย Evaluation module ทำหน้าที่หลักคือการเก็บค่าการจำลองสถานการณ์ และเก็บค่าจากข้อมูลที่ออกมาจาก Reputation management module ในรูปแบบของไฟล์ .csv เพื่อให้ในการวัดประสิทธิภาพของอัลกอริทึม

เมื่อเปรียบเทียบระหว่างสถาปัตยกรรมของทั้งสองส่วนจะพบว่า



อื่นๆ(ผู้พัฒนาต้องชี้แจงส่วนสำคัญที่ทีมงาน/ผู้พัฒนาได้พัฒนาขึ้นเองรวมทั้งต้องระบุแหล่งที่มาของโปรแกรม หรือ Source Code อื่นที่มาประกอบในโปรแกรมไว้ด้วย โดยไม่ต้องจัดพิมพ์ Source Code แนบมา)

ได้นำข้อมูลการรายงานสถานะน้ำท่วม มาจากแหล่งที่เปิดเผยข้อมูลให้ใช้ได้(Open data) ของ ประเทศฟิลิปปินส์ แล้วนำข้อมูลเหล่านั้นมาทำ Data Clearing โดยใช้ภาษา R จากนั้นเปลี่ยนข้อมูล ให้อยู่ในรูปแบบที่ต้องการ และสามารถใช้องค์ข้อมูลแบ่งพื้นที่ได้อย่าง Dynamic เวลาที่มีการตั้งค่าบริเวณ พื้นที่ที่จำลองใน NetLogo

ในส่วนของโปรแกรมที่ทำหน้าที่จำลองการทำงานของกลุ่ม Crowdsourcing ได้ทำการศึกษา Model ที่มีอยู่ใน Library และได้นำส่วนของการสร้างลูกศรในการส่งข้อมูลมาจาก Model หนึ่งใน Library ในส่วนการทำงาน และอัลกอริทึมอื่นๆที่ใช้ทางทีมผู้พัฒนาเป็นคนพัฒนาเองทั้งสิ้น

ขอบเขตและข้อจำกัดของโปรแกรมที่พัฒนา

1. ใช้อัลกอริทึมในการจัดการข้อมูล และจัดการผู้ใช้ คือ Majority Voting, EM Algorithm, Soft Penalty และ Hard Penalty
2. บริเวณที่นำมาจำลองสภาพน้ำท่วม เป็นพื้นที่ส่วนหนึ่งของข้อมูลที่เปิดให้ใช้ (Open data)
3. ระบบมีผู้ใช้ระบบทั้งหมด 2 แบบ คือ ผู้ใช้ทั่วไป กับผู้ใช้ที่ตั้งใจก่อวิน
4. ผู้ใช้ระบบแต่ละคนมีพฤติกรรมกรรายงานข้อมูลที่เป็นอิสระต่อกัน จะไม่มีเหตุการณ์ร่วมกัน รายงานข้อมูลเพื่อก่อวินระบบร่วมกัน
5. ระบบรองรับผู้ใช้แบบตั้งใจก่อวินได้ที่ละประเภทเท่านั้น ไม่สามารถใส่ลักษณะของผู้ใช้แบบ ก่อวินหลายประเภทในการจำลองพร้อมๆกัน
6. ผู้ใช้รายงานข้อมูลเข้ามาในระบบได้ 2 ค่า คือ ถนนสามารถใช้งานได้ดี และถนนไม่สามารถใช้งาน ได้
7. ระบบไม่ประมวลผลข้อมูลที่อยู่ในรูปแบบที่ไม่ถูกต้อง

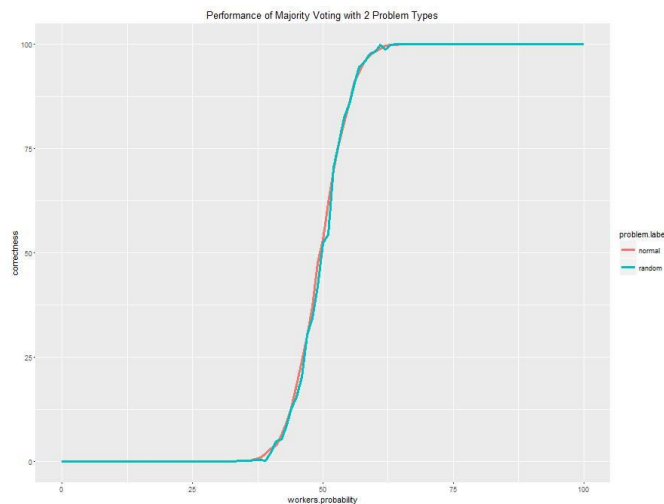
กลุ่มผู้ใช้โปรแกรม

ผู้ที่ต้องการสร้างระบบ Crowdsourcing หรือผู้ที่ต้องการนำอัลกอริทึมส่วนนี้เข้าไปเป็นส่วนหนึ่งของระบบ Crowdsourcing ที่มีอยู่ในปัจจุบัน และผู้ที่ต้องการเปรียบเทียบอัลกอริทึมที่จัดการความน่าเชื่อถือของระบบ Crowdsourcing

ผลของการทดสอบโปรแกรม

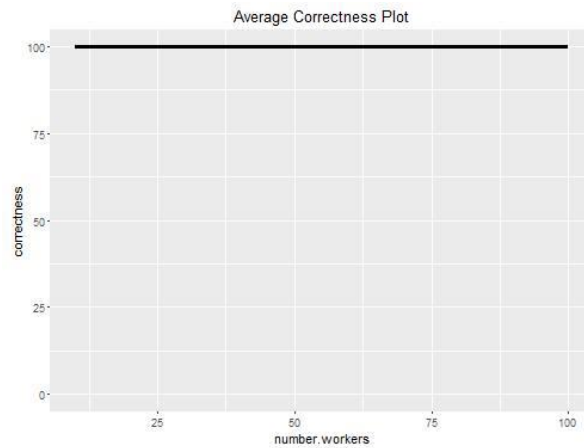
ในการทดสอบผลการทดลองโดยโปรแกรม จะทดสอบตามสมมติฐานต่างๆที่ผู้พัฒนาได้กำหนดไว้ ซึ่งกำหนดตัวแปรต้นด้วยค่าต่างๆที่สนใจ และมีตัวแปรตามคือค่าความถูกต้องของระบบที่คำนวณจากค่าที่ประมวลผลข้อมูลจาก Workers ทั้งหมด กับค่าคำตอบจริง และควบคุมตัวแปรอื่นๆเพื่อไม่ส่งผลกระทบกับตัวแปรที่เราสนใจ ตัวแปรต่างๆที่นำมาทดสอบ เช่น จำนวนของผู้ใช้ ความน่าจะเป็นของผู้ใช้ที่จะตอบคำตอบที่ถูกต้อง วิธีการรวมข้อมูลของระบบ วิธีการจัดการความน่าเชื่อถือของผู้ใช้ เป็นต้น ซึ่งในแต่ละการทดลองต่างๆ จะมีการทำซ้ำทั้งหมด 100 ครั้ง และนำค่าทั้งหมดมาเฉลี่ย เพื่อลดความคลาดเคลื่อนของข้อมูล

สาเหตุที่ทำการทดลองโดยใช้ Problem แบบ Normal เพียงอย่างเดียว เนื่องจากได้ทำการเปรียบเทียบค่าความถูกต้องของคำตอบโดยให้ตัวแปรตัวอื่นเหมือนกันทุกประการ แต่วิธีการที่ Problem เป็น Normal กับ Random ดังกราฟ



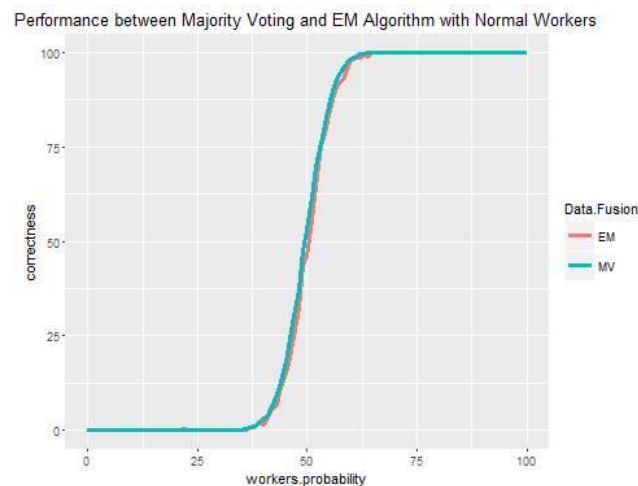
จากกราฟจะพบว่าลักษณะของ Problem ทั้งสองลักษณะดังกล่าวไม่มีผลต่อค่าความถูกต้องของระบบ ดังนั้นจึงเลือกที่จะใช้ Problem แบบ Normal ตลอด จะได้ลดตัวแปรที่ควบคุมไม่ได้ออกไป 1 ตัว แต่ในส่วนถ้าต้องการทดลองแบบที่มีผู้ใช้ออกวนประเภทที่ 1 จะใช้ Problem แบบ Random เนื่องจากผู้ที่ตั้งใจออกวนระบบประเภทที่ 1 เป็นผู้ใช้ที่จะตอบค่าเดิมเสมอจึงมีโอกาที่คำตอบเพิกทางตอบถูก หรือผิดเสมอ ทำให้ค่าความถูกต้องไม่ยุติธรรม

สาเหตุที่ทำการทดลองแล้วใช้จำนวน Worker เป็น 100 คนเสมอ เนื่องจากได้ทำการทดลองสมมติฐานว่าจำนวนมีผลต่อความถูกต้องหรือไม่หากตัวแปรอื่นๆถูกควบคุมให้เหมือนกัน ดังกราฟ



จากกราฟพบว่าจำนวนของผู้ใช้ไม่มีผลต่อค่าความถูกต้องของระบบ ดังได้เลือกที่จะกำหนดจำนวนผู้ใช้เป็น 100 คนเสมอ เพื่อลดตัวแปรที่ควบคุมไม่ได้ออกไป 1 ตัว

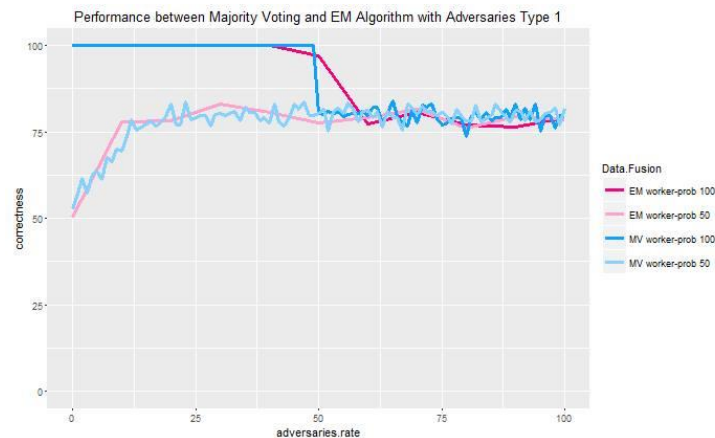
1. เปรียบเทียบประสิทธิภาพของวิธีรวมข้อมูลระหว่าง Majority Voting และ EM Algorithm ที่ไม่มี Reputation Management โดยให้มีจำนวนผู้ในระบบทั้งหมด 100 คน โดยทั้ง 100 คนเป็นผู้ใช้แบบปกติทั้งหมด โดยกำหนดโอกาสที่จะตอบคำตอบที่ถูกต้องของผู้ใช้ปกติตั้งแต่ 0% - 100%



จากกราฟพบว่า Majority Voting และ EM Algorithm มีประสิทธิภาพในการจัดการความถูกต้องของระบบใกล้เคียงกันมาก จะอาจกล่าวได้ว่ามีประสิทธิภาพเท่ากัน โดยทั้ง 2 อัลกอริทึมจะจัดการกลุ่มผู้ใช้ได้ดีในช่วงที่ผู้ใช้แบบปกติมีโอกาสตอบคำตอบที่ถูกต้องประมาณ 50% ส่วน และจะมีค่าความถูกต้องของระบบเป็น 100% เมื่อผู้ใช้แบบปกติมีโอกาสตอบคำตอบที่ถูกต้องตั้งแต่ 62 % ขึ้นไป

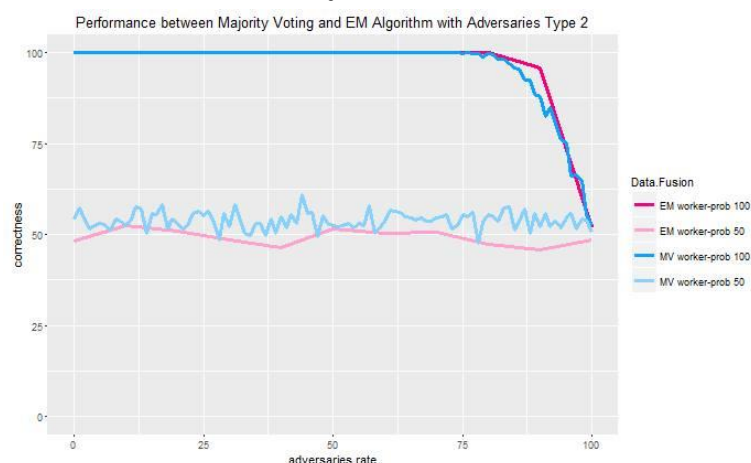
2. เปรียบเทียบประสิทธิภาพของวิธีรวมข้อมูลระหว่าง Majority Voting และ EM Algorithm ที่ไม่มี Reputation Management โดยให้มีจำนวนผู้ในระบบทั้งหมด 100 คน โดยกำหนดโอกาสที่จะตอบคำตอบที่ถูกต้องของผู้ใช้ปกติตั้งแต่ 0% - 100% และกำหนดว่ามีผู้ก่อวินาศกรรมระบบประเภทที่ 1 (ก่อวินาศกรรมตอบคำตอบเดียวเสมอ:ตอบค่าสถานะ Normal เสมอ) โดยจะมีอัตราของผู้ก่อวินาศกรรมตั้งแต่ 0 -

100%



จากกราฟพบว่า Majority Voting ที่มี Worker-prob เป็น 100% สามารถจัดการสถานการณ์ที่มีผู้ก่อวินประเพณีที่ 1 ได้ดีในช่วงที่มีเปอร์เซ็นต์ผู้ใช้ที่ตั้งใจก่อวินระบบในช่วง 0 – 49 % แต่พอช่วง 50 – 100 % จะเห็นได้ว่า Majority Voting ที่มี Worker-prob เป็น 100% จะมีความถูกต้องของระบบใกล้เคียงกับอัลกอริทึมชุดอื่น ในทางกลับกัน Majority Voting มี Worker-prob เป็น 50% ในช่วงที่มีผู้ใช้ที่ตั้งใจก่อวินระบบ 0 – 12% อัลกอริทึมชุดนี้สามารถในการจัดการความถูกต้องของระบบต่ำกว่าอัลกอริทึมชุดอื่นอย่างเห็นได้ชัด และในช่วงที่มีเปอร์เซ็นต์ผู้ใช้ที่ตั้งใจก่อวินระบบในช่วงตั้งแต่ 13% อัลกอริทึมชุดนี้จะมีประสิทธิภาพเทียบเท่าอัลกอริทึมชุดอื่น

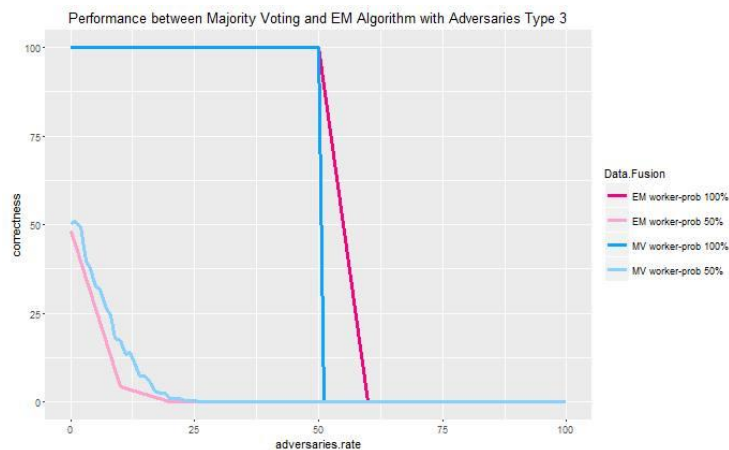
3. เปรียบเทียบประสิทธิภาพของวิธีรวมข้อมูลระหว่าง Majority Voting และ EM Algorithm ที่ไม่มี Reputation Management โดยให้มีจำนวนผู้ใช้ระบบทั้งหมด 100 คน โดยกำหนดโอกาสที่จะตอบคำตอบที่ถูกต้องของผู้ใช้ปกติตั้งแต่ 0% - 100% และกำหนดว่ามีผู้ก่อวินระบบประเภทที่ 2 (ก่อวินแบบสุ่มคำตอบเสมอ) โดยจะมีอัตราของผู้ก่อวินตั้งแต่ 0 – 100%



จากกราฟพบว่า EM Algorithm ที่มี Worker-prob เป็น 50% มีความสามารถในการจัดการค่าความถูกต้องในระบบได้อย่างคงที่อยู่ในช่วง 50 - 55 % ซึ่งกรณีนี้จะเห็นได้ว่าเปอร์เซ็นต์ของผู้ก่อวินประเพณีที่ 2 ไม่มีผลต่ออัลกอริทึมชุดนี้ แต่ถ้า EM Algorithm ที่มี Worker-prob เป็น 100%

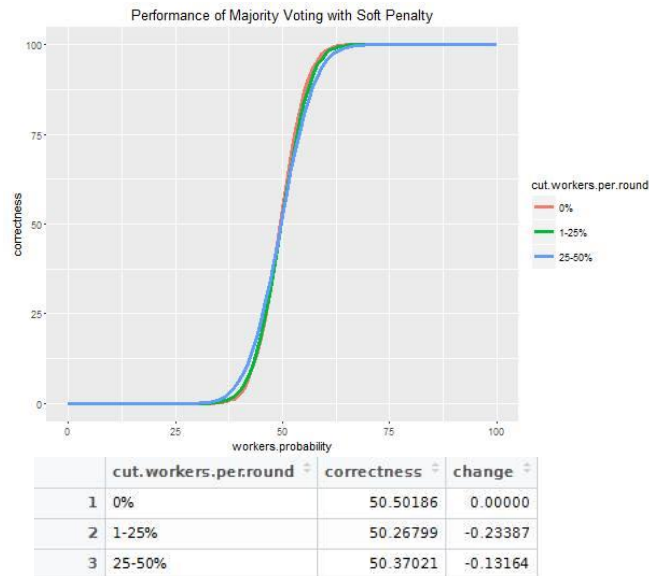
จะเห็นว่าเปอร์เซ็นต์ของผู้ก่อวินาศกรรมประเภทที่ 2 มีผลต่ออัลกอริทึมชุดนี้ และอัลกอริทึมชุดนี้มีค่าความถูกต้องของระบบดีมากในช่วงที่มีเปอร์เซ็นต์ของผู้ก่อวินาศกรรมประเภทที่ 2 เป็น 0 – 78% และตั้งแต่ 78% ค่าความถูกต้องจะลดลงเรื่อยๆจนเท่ากับอัลกอริทึมอีกชุดหนึ่ง

4. เปรียบเทียบประสิทธิภาพของวิธีรวมข้อมูลระหว่าง Majority Voting และ EM Algorithm ที่ไม่มี Reputation Management โดยให้มีจำนวนผู้ในระบบทั้งหมด 100 คน โดยกำหนดโอกาสที่จะตอบคำตอบที่ถูกต้องของผู้ใช้ปกติตั้งแต่ 0% - 100% และกำหนดว่ามีผู้ก่อวินาศกรรมประเภทที่ 3 (ก่อวินาศกรรมแบบส่งคำตอบตรงข้ามกับคำตอบที่ถูกต้องเสมอ) โดยจะมีอัตราของผู้ก่อวินาศกรรมตั้งแต่ 0 – 100%



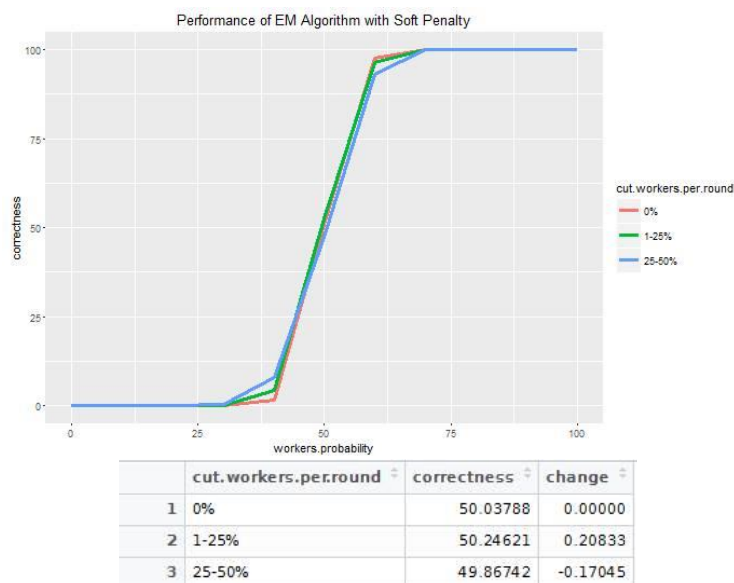
จากกราฟพบว่า EM Algorithm และ Majority Voting ที่มี Worker-prob เป็น 100% สามารถจัดการสถานการณ์ที่มีผู้ก่อวินาศกรรมประเภทที่ 3 ได้ดีมากในช่วงที่มีเปอร์เซ็นต์ผู้ใช้ที่ตั้งใจก่อวินาศกรรมประเภทที่ 3 น้อยกว่า 50% ตั้งแต่ 50% ขึ้นไปจะทำให้ค่าความถูกต้องของระบบลดลงเหลือ 0% ทันที ในส่วน Majority Voting และ EM Algorithm ที่มี Worker-prob เป็น 50% สามารถจัดการสถานการณ์ที่มีผู้ก่อวินาศกรรมประเภทที่ 3 ไม่ได้เท่าที่ควรที่มีเปอร์เซ็นต์ผู้ใช้ที่ตั้งใจก่อวินาศกรรมประเภทที่ 3 น้อยกว่า 20% แต่ถ้าตั้งแต่ 20% ขึ้นไปจะทำให้ค่าความถูกต้องของระบบลดลงเหลือ 0% เช่นกัน

5. เปรียบเทียบประสิทธิภาพของวิธีรวมข้อมูลของ Majority Voting โดยมี Soft Penalty เป็น Reputation Management โดยให้มีจำนวนผู้ในระบบทั้งหมด 100 คน โดยทั้ง 100 คนเป็นผู้ใช้แบบปกติทั้งหมด โดยกำหนดโอกาสที่จะตอบคำตอบที่ถูกต้องของผู้ใช้ปกติตั้งแต่ 0% - 100% นอกจากนี้ยังมีเปอร์เซ็นต์ที่จะตัดผู้ใช้ออกจากระบบต่อรอบมีในช่วง 0% , 1-25% และ 25-50%



จากกราฟจะพบว่า Majority Voting เมื่อทำงานร่วมกับ Soft Penalty จัดการกลุ่มผู้ใช้ได้ดีในช่วงที่ผู้ใช้แบบปกติมีโอกาสตอบคำถามที่ถูกต้องประมาณ 50% ส่วน และจะมีค่าความถูกต้องของระบบเป็น 100% เมื่อผู้ใช้แบบปกติมีโอกาสตอบคำถามที่ถูกต้องตั้งแต่ 62 % ขึ้นไป ซึ่งค่าคล้ายกับการที่มี Majority voting เพียงอย่างเดียว และเห็นได้ชัดว่าเปอร์เซ็นต์การตัดผู้ใช้ในแต่ละรอบออกไม่มีผลต่อความถูกต้องของระบบ สังเกตได้จากกราฟทั้งสามเส้นที่ใกล้เคียงกันมาก

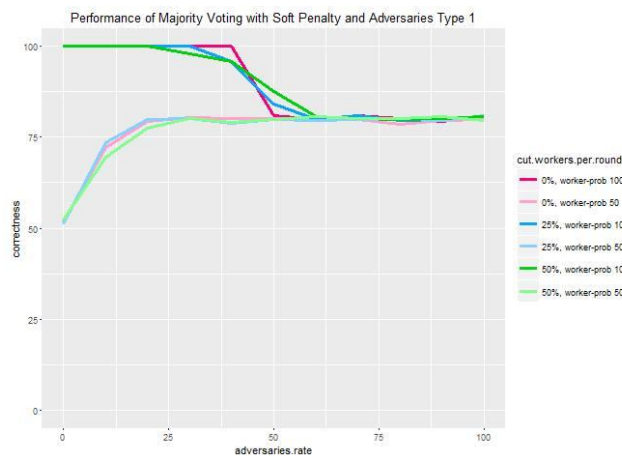
- เปรียบเทียบประสิทธิภาพของวิธีรวมข้อมูลของ EM Algorithm โดยมี Soft Penalty เป็น Reputation Management โดยให้มีจำนวนผู้ในระบบทั้งหมด 100 คน โดยทั้ง 100 คนเป็นผู้ใช้แบบปกติทั้งหมด โดยกำหนดโอกาสที่จะตอบคำถามที่ถูกต้องของผู้ใช้ปกติตั้งแต่ 0% - 100% นอกจากนี้ยังมีเปอร์เซ็นต์ที่จะตัดผู้ใช้ออกจากระบบต่อรอบมีในช่วง 0% ,1-25% และ 25-50%



จากกราฟจะพบว่า EM Algorithm เมื่อทำงานร่วมกับ Soft Penalty จัดการกลุ่มผู้ใช้ได้ดีในช่วงที่ผู้ใช้แบบปกติมีโอกาสตอบคำถามที่ถูกต้องประมาณ 50% ส่วน และจะมีค่าความถูกต้องของระบบ

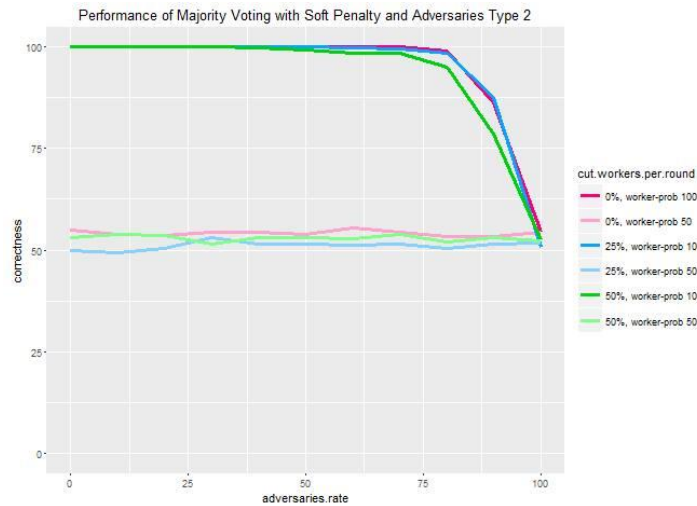
เป็น 100% เมื่อผู้ใช้แบบปกติมีโอกาสตอบคำถามที่ถูกต้องตั้งแต่ 65 % ขึ้นไป ซึ่งค่าคล้ายกับการที่มี EM Algorithm เพียงอย่างเดียว และเห็นได้ชัดว่าเปอร์เซ็นต์การตัดผู้ใช้ในแต่ละรอบออกมีผลต่อความถูกต้องของระบบเล็กน้อย ค่าจะเด่นชัดของโอกาสที่จะตอบคำถามที่ถูกต้องของผู้ใช้ปกติในช่วง 30 - 40% และ 55 - 65% แต่ในส่วนอื่นจะใกล้เคียงกันมาก

7. เปรียบเทียบประสิทธิภาพของวิธีรวมข้อมูลของ Majority Voting โดยมี Soft Penalty เป็น Reputation Management โดยให้มีจำนวนผู้ใช้ระบบทั้งหมด 100 คน โดยทั้ง 100 คน มีผู้ก่อวินระบบประเภทที่ 1 ตั้งแต่ 0 - 100% โดยกำหนดโอกาสที่จะตอบคำถามที่ถูกต้องของผู้ใช้ปกติตั้งแต่ 0% - 100% นอกจากนี้ยังมีเปอร์เซ็นต์ที่จะตัดผู้ใช้ออกจากระบบต่อรอบมีในช่วง 0% , 1-25% และ 25-50



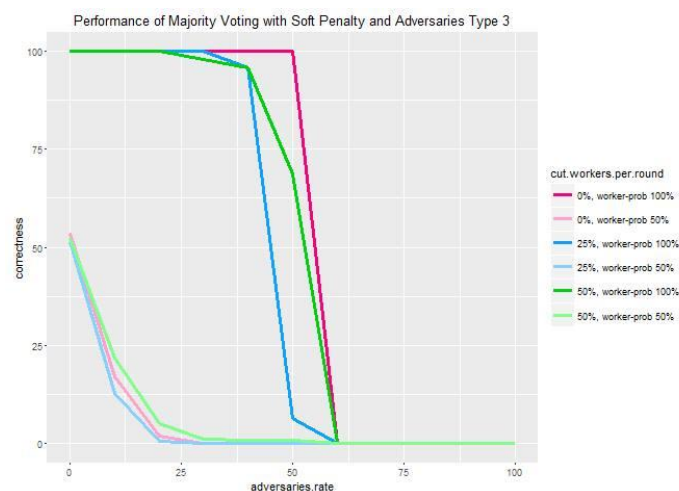
จากกราฟพบว่า Majority Voting ที่มี Soft Penalty ใช้งานกับผู้ก่อวินระบบประเภทที่ 1 และมีการตัดผู้ใช้งานระบบออกในแต่ละรอบ จะเห็นได้ว่าเปอร์เซ็นต์ที่ตัดผู้ใช้ออกจากระบบไม่มีผลต่อค่าความถูกต้องของระบบโดยรวมซึ่งต่างกับ Worker-prob ซึ่งถ้าเปลี่ยนจาก 100% มาเป็น 50% จะให้ค่าความถูกต้องของระบบต่างกันอย่างเห็นได้ชัด แต่ก็มีอีก 1 ปัจจัยคือเปอร์เซ็นต์ของผู้ก่อวิน มีเปอร์เซ็นต์มากก็จะทำให้ค่าความถูกต้องของระบบลดลงเช่นกันในทุกชุดกรณี และจะลงมาอยู่เท่าๆกัน

8. เปรียบเทียบประสิทธิภาพของวิธีรวมข้อมูลของ Majority Voting โดยมี Soft Penalty เป็น Reputation Management โดยให้มีจำนวนผู้ใช้ระบบทั้งหมด 100 คน โดยทั้ง 100 คน มีผู้ก่อวินระบบประเภทที่ 2 ตั้งแต่ 0 - 100% โดยกำหนดโอกาสที่จะตอบคำถามที่ถูกต้องของผู้ใช้ปกติตั้งแต่ 0% - 100% นอกจากนี้ยังมีเปอร์เซ็นต์ที่จะตัดผู้ใช้ออกจากระบบต่อรอบมีในช่วง 0% , 1-25% และ 25-50



จากกราฟพบว่า Majority Voting ที่มี Soft Penalty ใช้งานกับผู้ก่อวินาศกรรมระบบประเภทที่ 2 และมีการตัดผู้ใช้งานระบบออกในแต่ละรอบ จะเห็นได้ว่าเปอร์เซ็นต์ที่ตัดผู้ใช้ออกจากระบบไม่มีผลต่อค่าความถูกต้องของระบบโดยรวมซึ่งต่างกับ Worker-prob ซึ่งถ้าเปลี่ยนจาก 100% มาเป็น 50% จะให้ค่าความถูกต้องของระบบต่างกันอย่างเห็นได้ชัด แต่ในส่วน Worker-prob 50% จะมีความเสถียรไม่ขึ้นกับเปอร์เซ็นต์ของผู้ก่อวินาศกรรมระบบ แต่ก็มีอีก 1 ปัจจัยคือเปอร์เซ็นต์ของผู้ก่อวินาศกรรม ทำให้มีเปอร์เซ็นต์มากก็จะทำให้ค่าความถูกต้องของระบบของกรณี Worker-prob 100% ลดลงมาเท่ากับ Worker-prob 50%

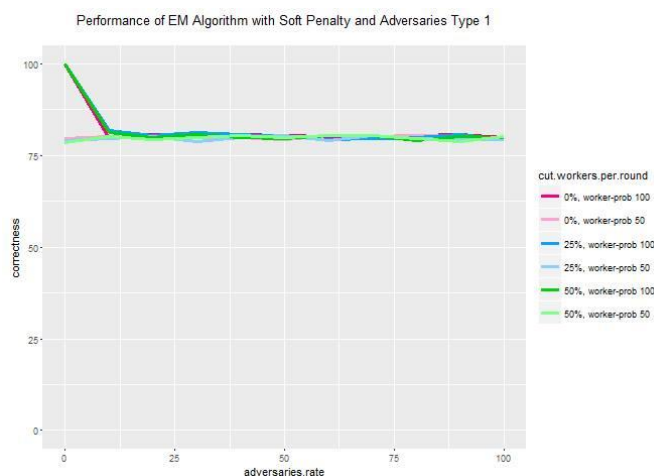
- เปรียบเทียบประสิทธิภาพของวิธีรวมข้อมูลของ Majority Voting โดยมี Soft Penalty เป็น Reputation Management โดยให้มีจำนวนผู้ในระบบทั้งหมด 100 คน โดยทั้ง 100 คน มีผู้ก่อวินาศกรรมระบบประเภทที่ 3 ตั้งแต่ 0 – 100% โดยกำหนดโอกาสที่จะตอบคำตอบที่ถูกต้องของผู้ใช้ปกติตั้งแต่ 0% - 100% นอกจากนี้ยังมีเปอร์เซ็นต์ที่จะตัดผู้ใช้ออกจากระบบต่อรอบมีในช่วง 0% ,1-25% และ 25-50



จากกราฟพบว่า Majority Voting ที่มี Soft Penalty ใช้งานกับผู้ก่อวินาศกรรมระบบประเภทที่ 3 และมีการตัดผู้ใช้งานระบบออกในแต่ละรอบ จะเห็นได้ว่าเปอร์เซ็นต์ที่ตัดผู้ใช้ออกจากระบบไม่มีผลต่อค่าความถูกต้องของระบบโดยรวมซึ่งต่างกับ Worker-prob ซึ่งถ้าเปลี่ยนจาก 100% มาเป็น 50% จะให้ค่าความถูกต้องของ

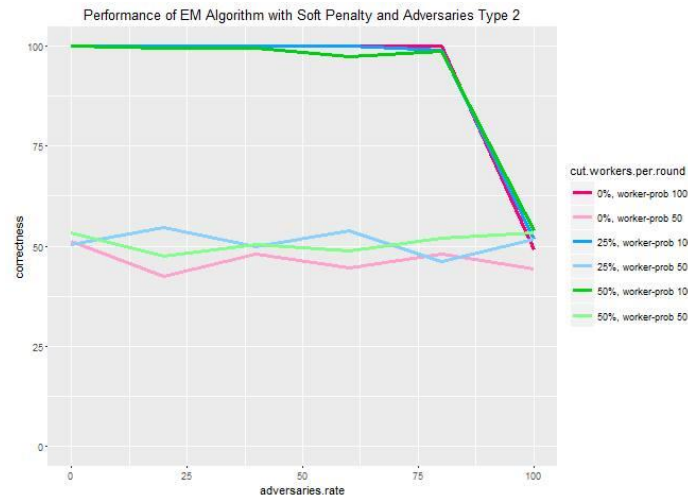
ระบบต่างกันอย่างเห็นได้ชัด แต่ก็มีอีก 1 ปัจจัยคือเปอร์เซ็นต์ของผู้ก่อวินาศกรรม มีเปอร์เซ็นต์มากก็จะทำให้ค่าความถูกต้องของระบบลดลงเช่นกันในทุกชุดกรณี และจะลงมาอยู่เท่าๆกัน จนสุดท้ายถ้ามีผู้ก่อวินาศกรรมประเภทที่ 3 มากๆจะทำให้ค่าความถูกต้องของระบบเป็น 0% ทันที

10. เปรียบเทียบประสิทธิภาพของวิธีรวมข้อมูลของ EM Algorithm โดยมี Soft Penalty เป็น Reputation Management โดยให้มีจำนวนผู้ใช้ระบบทั้งหมด 100 คน โดยทั้ง 100 คน มีผู้ก่อวินาศกรรมระบบประเภทที่ 1 ตั้งแต่ 0 – 100% โดยกำหนดโอกาสที่จะตอบคำถามที่ถูกต้องของผู้ใช้ปกติตั้งแต่ 0% - 100% นอกจากนี้ยังมีเปอร์เซ็นต์ที่จะตัดผู้ใช้ออกจากระบบต่อรอบมีในช่วง 0% ,1-25% และ 25-50



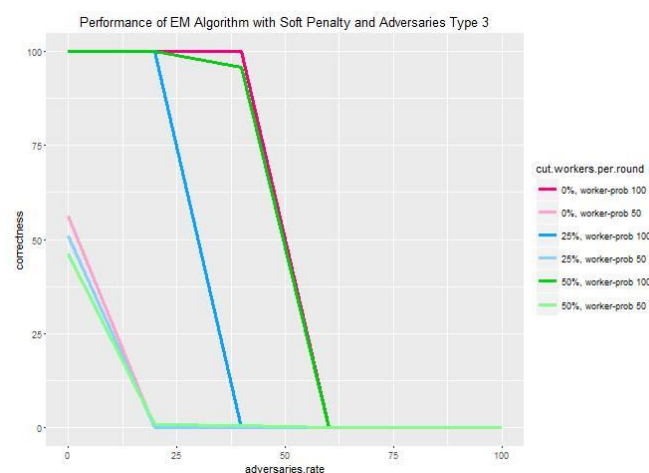
จากกราฟพบว่า EM Algorithm ที่มี Soft Penalty ใช้งานกับผู้ก่อวินาศกรรมระบบประเภทที่ 1 และมีการตัดผู้ใช้งานระบบออกในแต่ละรอบ จะเห็นได้ว่าเปอร์เซ็นต์ที่ตัดผู้ใช้ออกจากระบบไม่มีผลต่อค่าความถูกต้องของระบบโดยรวมซึ่งต่างกับ Worker-prob ซึ่งถ้าเปลี่ยนจาก 100% มาเป็น 50% จะให้ค่าความถูกต้องของระบบต่างกันอย่างเห็นได้ชัดในช่วงที่มีผู้ก่อวินาศกรรม 0 – 10% เมื่อเกิน 10% จะทำให้ทุกชุดทดสอบให้ค่าความถูกต้องของระบบออกมาอย่างเท่ากัน

11. เปรียบเทียบประสิทธิภาพของวิธีรวมข้อมูลของ EM Algorithm โดยมี Soft Penalty เป็น Reputation Management โดยให้มีจำนวนผู้ใช้ระบบทั้งหมด 100 คน โดยทั้ง 100 คน มีผู้ก่อวินาศกรรมระบบประเภทที่ 2 ตั้งแต่ 0 – 100% โดยกำหนดโอกาสที่จะตอบคำถามที่ถูกต้องของผู้ใช้ปกติตั้งแต่ 0% - 100% นอกจากนี้ยังมีเปอร์เซ็นต์ที่จะตัดผู้ใช้ออกจากระบบต่อรอบมีในช่วง 0% ,1-25% และ 25-50



จากกราฟพบว่า EM Algorithm ที่มี Soft Penalty ใช้งานกับผู้ก่อวินาศกรรมระบบประเภทที่ 3 และมีการตัดผู้ใช้งานระบบออกในแต่ละรอบ จะเห็นว่าเปอร์เซ็นต์ที่ตัดผู้ใช้งานออกจากระบบมีผลต่อค่าความถูกต้องของระบบโดยรวมเล็กน้อย ถ้า Worker-prob ซึ่งถ้าเปลี่ยนจาก 100% มาเป็น 50% จะให้ค่าความถูกต้องของระบบต่างกันเล็กน้อยเห็นได้ชัดในช่วงที่มีผู้ก่อวินาศกรรมในช่วงต้น 0 – 78% เมื่อเกิน 78% จะทำให้ทุกชุดทดสอบให้ค่าความถูกต้องของระบบออกมาอย่างเท่ากัน แต่ข้อดีของ Worker-prob 50% คือจะให้ค่าความถูกต้องของระบบออกมาค่อนข้างเสถียร

12. เปรียบเทียบประสิทธิภาพของวิธีรวมข้อมูลของ EM Algorithm โดยมี Soft Penalty เป็น Reputation Management โดยให้มีจำนวนผู้ใช้งานระบบทั้งหมด 100 คน โดยทั้ง 100 คน มีผู้ก่อวินาศกรรมระบบประเภทที่ 3 ตั้งแต่ 0 – 100% โดยกำหนดโอกาสที่จะตอบคำถามที่ถูกต้องของผู้ใช้ปกติตั้งแต่ 0% - 100% นอกจากนี้ยังมีเปอร์เซ็นต์ที่ตัดผู้ใช้งานออกจากระบบต่อรอบมีในช่วง 0% ,1-25% และ 25-50



จากกราฟพบว่า EM Algorithm ที่มี Soft Penalty ใช้งานกับผู้ก่อวินาศกรรมระบบประเภทที่ 3 และมีการตัดผู้ใช้งานระบบออกในแต่ละรอบ จะเห็นว่าเปอร์เซ็นต์ที่ตัดผู้ใช้งานออกจากระบบมีผลต่อค่าความถูกต้องของระบบโดยรวมเล็กน้อย ถ้า Worker-prob ซึ่งถ้าเปลี่ยนจาก 100% มาเป็น 50% จะให้ค่าความถูกต้องของ

ระบบต่างกันอย่างเห็นได้ชัด และถ้ามีผู้ก่อวินประเพณีที่ 3 จำนวนมากจะทำให้ความถูกต้องของระบบลดลงเหลือ 0 ทันที

ปัญหาและอุปสรรค

1. อัลกอริทึมที่ใช้ในส่วน EM algorithm และ Hard Penalty นำมา Implementation ยากเนื่องจากต้องมีการเก็บค่าในแต่ละรอบ เพื่อใช้มาเป็นค่าตั้งต้นในรอบต่อไป
2. หาข้อมูลที่เปิดให้ใช้อย่างอิสระนำมาใช้ในส่วนจำลอง Problem มีความท้าทาย เนื่องได้ตอนแรกได้ข้อมูลมากจากสำนักการระบายน้ำของกรุงเทพมหานคร แต่ข้อมูลส่วนใหญ่ที่ได้มาไม่อยู่ในรูปที่สามารถใช้งานได้ถึงแม้จะทำ Data Cleaning แล้วก็ตาม ทำให้ต้องหาข้อมูลจากแหล่งใหม่ ทำการศึกษา ทำความเข้าใจข้อมูล และ Data Cleaning ใหม่
3. เครื่องมือที่ใช้ในการจำลองทางทีมผู้พัฒนาได้ทำการศึกษาค้นคว้า และทำความเข้าใจ NetLogo Programming Language ด้วยตนเอง
4. เนื่องจากมีตัวแปรหลายตัวแปรที่ใช้ในการจำลองขึ้น ทำให้การรันผลในแต่ละสมมติฐานใช้เวลานาน เช่น ถ้าสมมติฐานนั้นมี EM Algorithm ช่วยในการทำ Data fusion จะทำให้การรันสมมติฐานนั้นใช้เวลามากกว่า 4 ชั่วโมง

แนวทางในการพัฒนาและประยุกต์ใช้ร่วมกับงานอื่นๆ ในขั้นต่อไป

1. นำอัลกอริทึมที่ใช้ในการจัดการความน่าเชื่อถือไปใช้กับระบบ Crowdsourcing ที่มีอยู่ในปัจจุบัน
2. นำระบบ Crowdsourcing ไปแทนหรือช่วยในการทำงานของระบบที่รับข้อมูลจากอุปกรณ์ต่างๆ เช่น เซนเซอร์ เนื่องจากระบบที่รับข้อมูลจากอุปกรณ์ข้อจำกัดเรื่องพื้นที่ที่เข้าถึง และหากอุปกรณ์เสียหายจะทำให้ไม่สามารถส่งค่ามาที่ระบบได้
3. การจำลองบริเวณน้ำท่วมโดยสร้างมาจากข้อมูลจริงมาผ่านกระบวนการทำให้ Distribution ออกมาสามารถนำมาเป็น Model จำลองสถานการณ์น้ำท่วมรูปแบบต่างๆ เพื่อจะได้ตั้งมือรับสถานการณ์ที่ไม่คาดฝันได้

ข้อสรุปและข้อเสนอแนะ

จากผลการทดลองจะสรุปได้ว่าวิธีการรวมข้อมูลแบบ Majority Voting และ EM Algorithm จะให้ผลลัพธ์ที่ใกล้เคียงกันในหลายกรณี แต่ EM Algorithm สามารถปรับแก้ประคองความถูกต้องของระบบได้ดีกว่า Majority Voting ส่วนของวิธีการจัดการความน่าเชื่อถือของผู้ใช้ Soft Penalty สามารถเพิ่มความถูกต้องของระบบได้ประมาณ 0.5% และบางสถานการณ์จะทำให้ความถูกต้องของระบบลดลงอีกด้วย ซึ่งเป็นมาจากการวิธีการตัดข้อมูลของผู้ใช้ที่ยังไม่เหมาะสมกับระบบ

เอกสารอ้างอิง (Reference)

Jagabathula, S., Subramanian, L., & Venkataraman, A. (2014). *Reputation-based Worker Filtering in Crowdsourcing*. Retrieved from <http://papers.nips.cc/paper/5393-reputation-based-worker-filtering-in-crowdsourcing.pdf>

Buecheler, T., Lonigro, R., Fuchslin, R. M., & Pfeifer, R. (n.d.). *Modeling and Simulating Crowdsourcing as a Complex Biological System: Human Crowds Manifesting Collective Intelligence on the Internet*. Retrieved from <https://mitpress.mit.edu/sites/default>

Davami, E., & Sukthankar, G. (2015). *Improving the Performance of Mobile Phone*. Retrieved from http://www.aamas2015.com/en/AAMAS_2015_USB/aamas/p145.pdf

Harvey, N. J., Ladner, E. R., Lovasz, L., & Tamir, T. (n.d.). *Semi-Matchings for Bipartite Graphs*. Retrieved from

<http://www.cs.ubc.ca/~nickhar/Publications/SemiMatching/WADS-SemiMatching.pdf>

Jøsang, A., & Ismail, R. (2002). *The Beta Reputation System*. Retrieved from [https://domino.fov.unimb.si/proceedings.nsf/proceedings/d9e48b66f32a7dffc1256e9f00355b37/\\$file/josang.pdf](https://domino.fov.unimb.si/proceedings.nsf/proceedings/d9e48b66f32a7dffc1256e9f00355b37/$file/josang.pdf)

Karger, D. R., Oh, S., & Shah, D. (2011). Iterative learning for reliable crowdsourcing systems. *Netrual Infomation Processing System* .

Ladson, T. (2015, June 29). *100-year flood: Poisson distribution*. Retrieved from <https://tonyladson.wordpress.com/2015/06/29/100-year-flood-poisson-distribution/>

Xintong, G., Hongzhi, W., Yangqiu, S., & Hong, G. (2014, July 11). *Brief survey of crowdsourcing for data mining*. Retrieved from <http://romisatriawahono.net/lecture/dm/survey/Xintong%20%20Crowdsourcing%20for%20Data%20Mining%20-%202014.pdf>

ข้อมูลของผู้พัฒนา

ทีมพัฒนา

1. ชื่อ-สกุล: นางสาวนภวรรณ ดุษฎีเวชกุล
วัน/เดือน/ปีเกิด 17/10/2537 อายุ: 21 ปี ระดับการศึกษา ปริญญาตรี
สถานศึกษา มหาวิทยาลัยธรรมศาสตร์
ที่อยู่ตามทะเบียนบ้าน 99 หมู่ 18 ต.คลองหนึ่ง อ.คลองหลวง จ.ปทุมธานี 12121
สถานที่ติดต่อ 89/1588 หมู่ที่ 5 ต.บางแม่นาง อ.บางใหญ่ จ.นนทบุรี 11140
มือถือ 082-959-9494 e-mail napawan-bohotmail.com

ประวัติการศึกษา

ระดับมัธยมศึกษาตอนต้น : โรงเรียนเบญจมราชาลัยในพระบรมราชูปถัมภ์
ระดับมัธยมศึกษาตอนปลาย : โรงเรียนเบญจมราชาลัยในพระบรมราชูปถัมภ์
ระดับอุดมศึกษา : มหาวิทยาลัยธรรมศาสตร์ สาขา วิทยาการคอมพิวเตอร์

2. ชื่อ-สกุล: นายวงศธร ทองถาวร วัน/เดือน/ปีเกิด : 07/10/2538 อายุ: 20 ปี ระดับการศึกษา ปริญญาตรี
สถานศึกษา มหาวิทยาลัยธรรมศาสตร์
ที่อยู่ตามทะเบียนบ้าน : 99 หมู่ 18 ต.คลองหนึ่ง อ.คลองหลวง จ.ปทุมธานี 12121
สถานที่ติดต่อ : 7/114 ซ.พงษ์เพชร 6ถ.แจ้งวัฒนะ ต.บ้านใหม่ อ. ปากเกร็ด จ.นนทบุรี 11120
โทรศัพท์มือถือ 088-641-9994 e-mail circle_ng@hotmail.com

ประวัติการศึกษา

ระดับมัธยมศึกษาตอนต้น : โรงเรียนหอวัง
ระดับมัธยมศึกษาตอนปลาย : โรงเรียนหอวัง
ระดับอุดมศึกษา : มหาวิทยาลัยธรรมศาสตร์ สาขา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาโครงการ

1. ชื่อ-นามสกุล นางสาวประภาพร รัตนธ ราง
สังกัด/สถาบัน มหาวิทยาลัยธรรมศาสตร์ (รังสิต)
สถานที่ติดต่อ อาคารบรรยายรวม 2 เลขที่99 ถ.พหลโยธิน ม.18 ต.คลองหนึ่ง อ.คลองหลวง จ.ปทุมธานี 12120
โทรศัพท์ 02-989-9156 มือถือ 086-880-9343 e-mail rattanat@cs.tu.ac.th

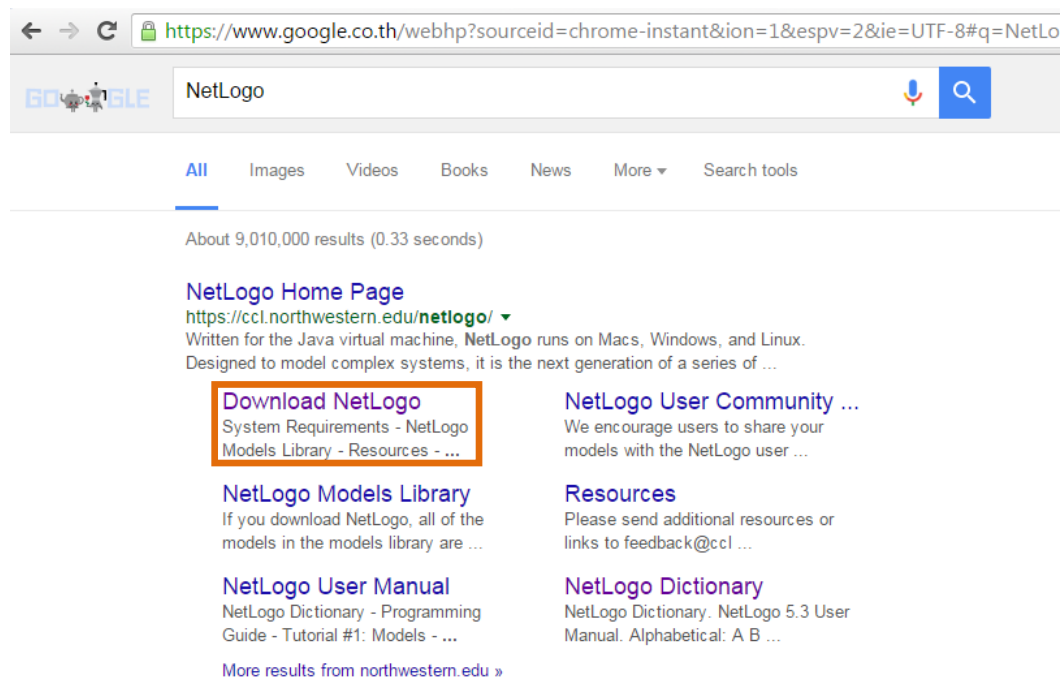
ภาคผนวก

1.คู่มือการติดตั้งเครื่องมือที่ใช้งาน

1.1 NetLogo

1) ค้นหา NetLogo ใน Search Engine เลือกที่ Download NetLogo หรือ

<https://ccl.northwestern.edu/netlogo/download.shtml>



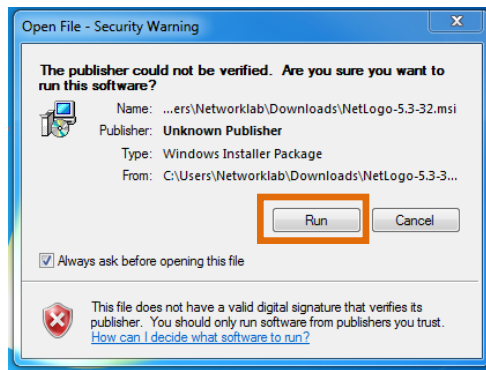
2) เลือก NetLogo Version 5.3 แล้วกด Download



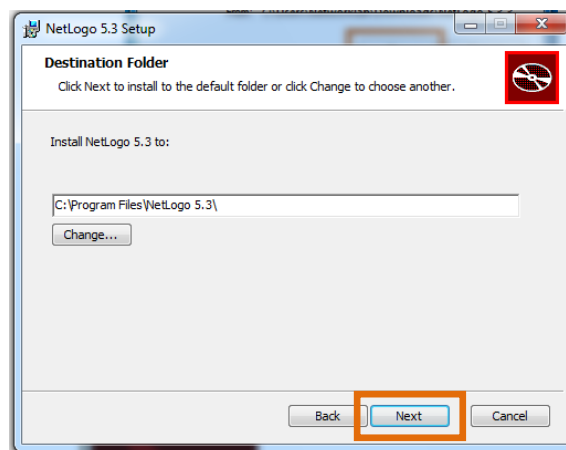
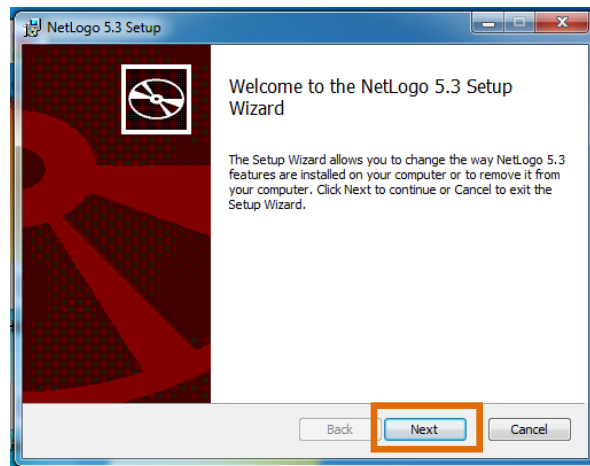
3) เลือก OS ให้ตรงกับคอมพิวเตอร์ที่มี

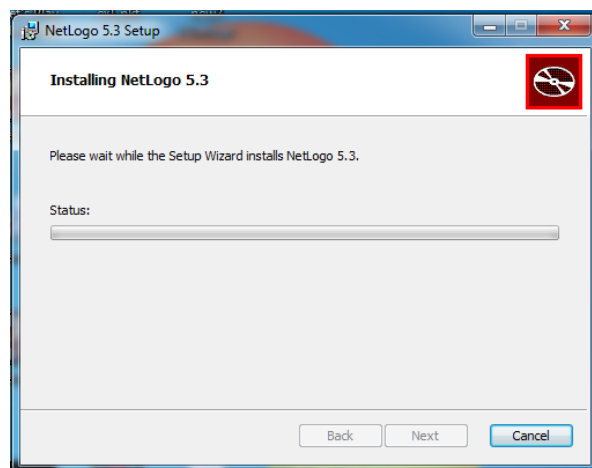
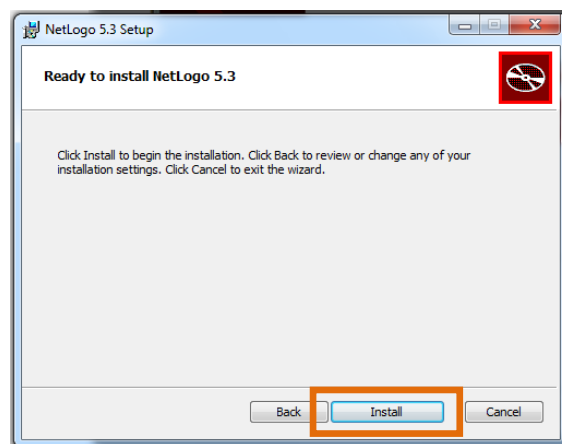
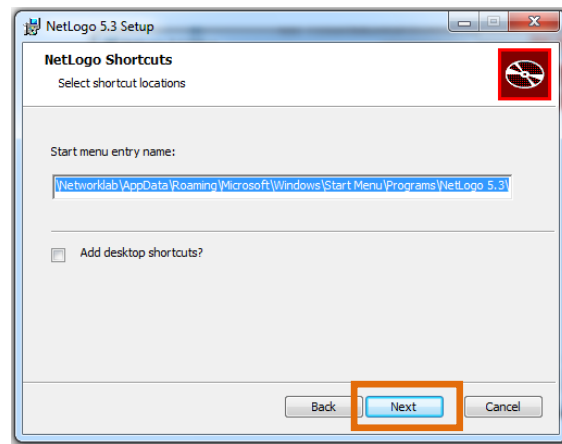


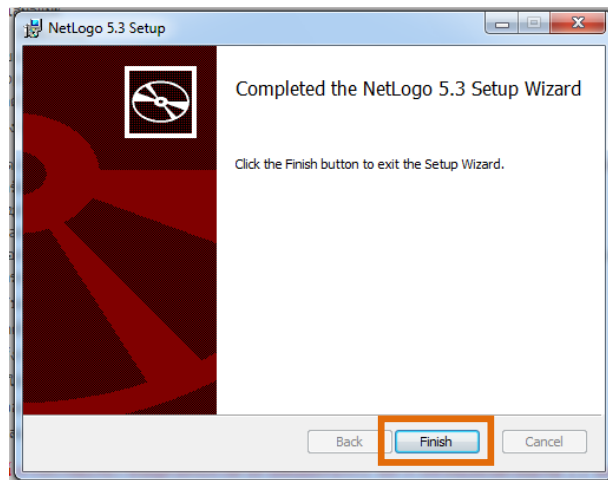
4) เมื่อ Download เสร็จเรียบร้อย คลิกตัวลงโปรแกรมขึ้นมา แล้วเลือก Run



5) เลือกที่ที่ต้องการให้ไฟล์อยู่ กด Next ไปเรื่อยๆ และกด Install จากนั้นรอติดตั้งโปรแกรมสำเร็จ

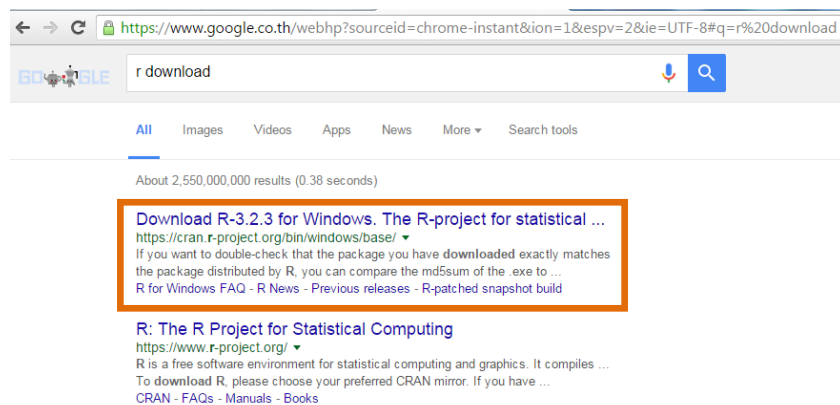




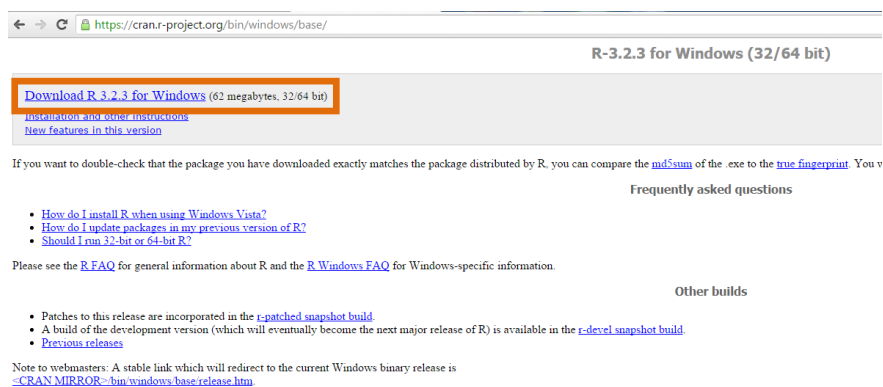


1.2 R

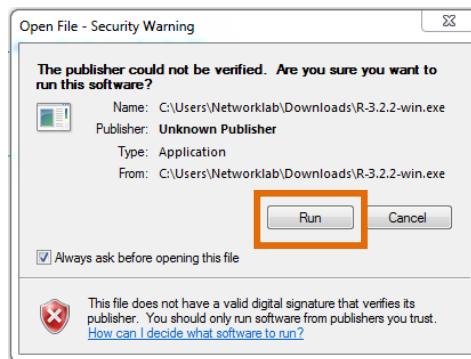
1) ค้นหา R Download ใน Search engine แล้วเข้าไปที่ link ของ cran.r หรือ <https://cran.r-project.org/bin/windows/base/>



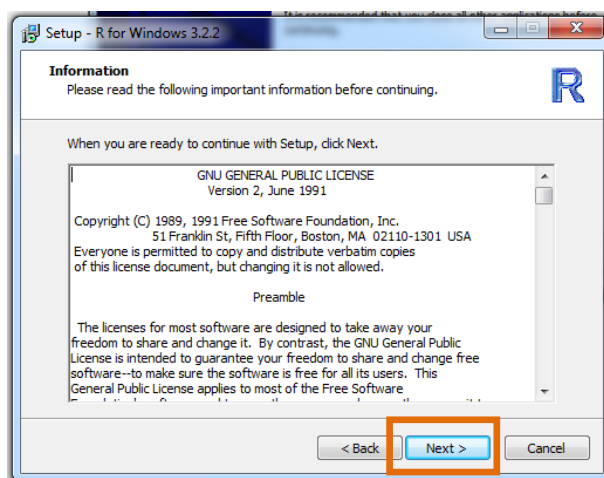
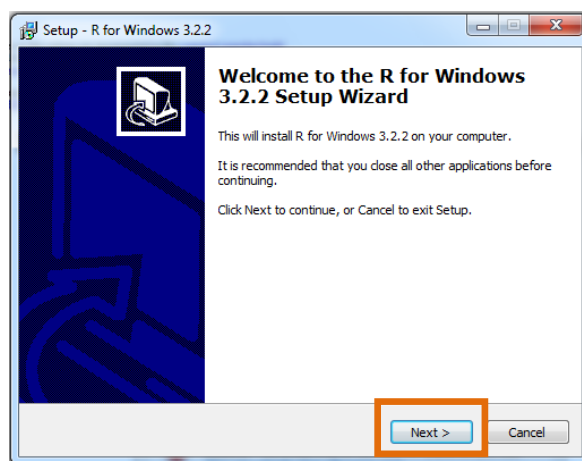
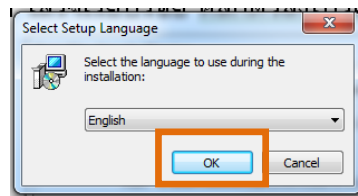
2) เลือก Download โปรแกรม

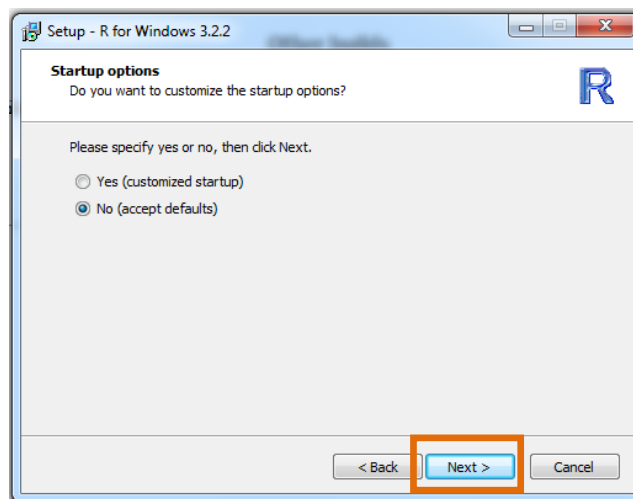
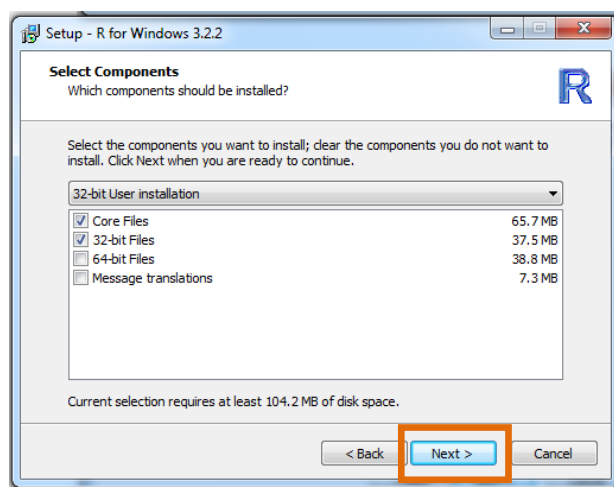
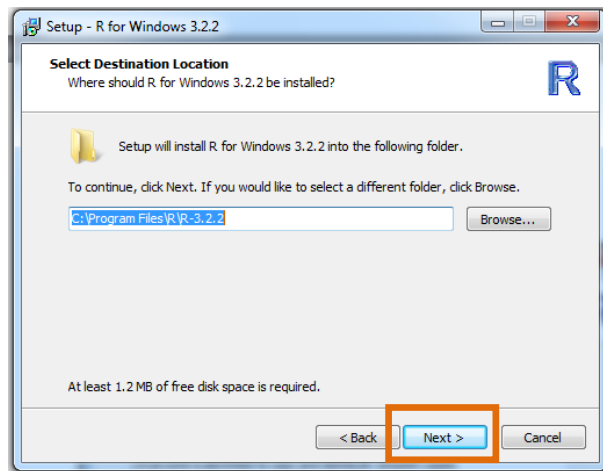


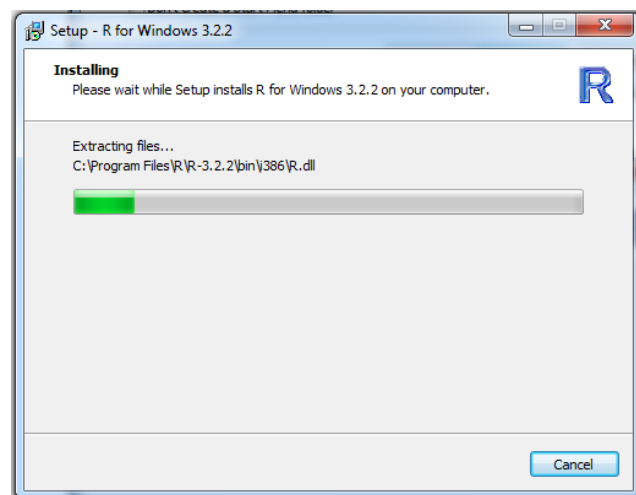
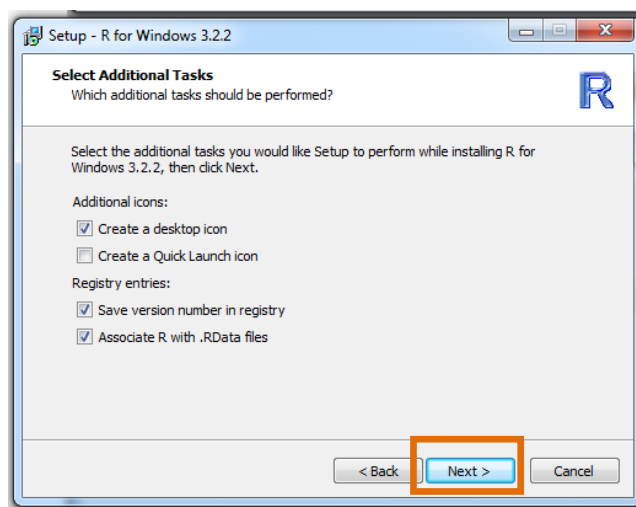
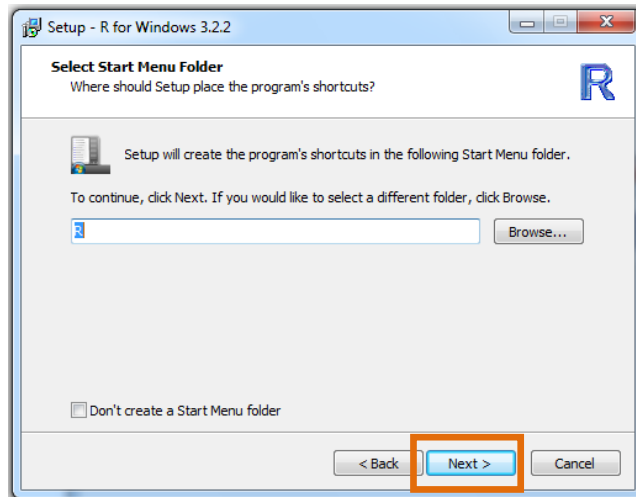
3) เมื่อ Download เสร็จเรียบร้อย คลิกตัวลงโปรแกรมขึ้นมา แล้วเลือก Run

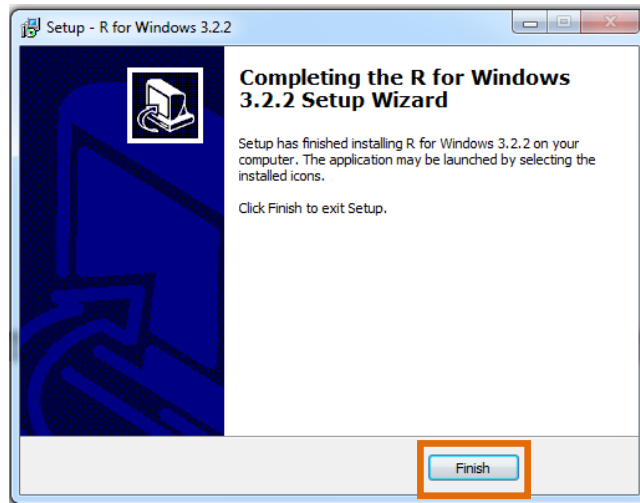


4) เลือกภาษาที่ต้องการใช้ในโปรแกรม เลือกที่ที่ต้องการให้ไฟล์อยู่ กด Next ไปเรื่อยๆ และ กด Install จากนั้นรอติดตั้งโปรแกรมสำเร็จ



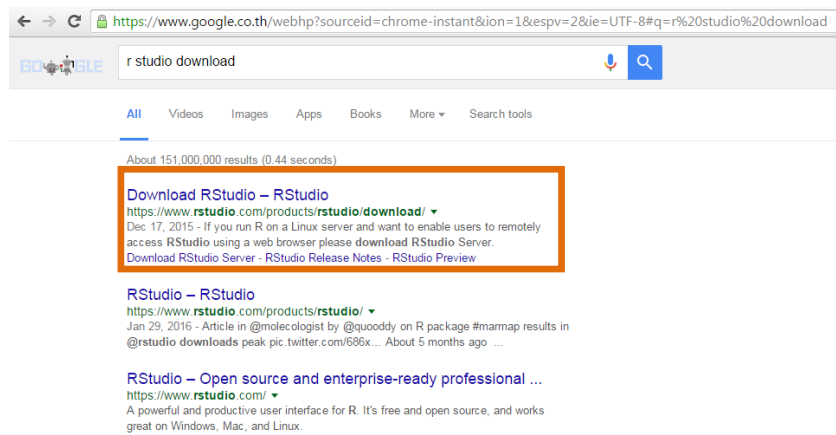







1.3 RStudio

1) ค้นหา RStudio Download ใน Search engine แล้วเข้าไปที่ link ของ rstudio หรือ <https://www.rstudio.com/products/rstudio/download/>



2) เลือก Download โปรแกรม ให้เหมาะสมกับคุณสมบัติของเครื่อง



[Products](#)
[Resources](#)
[Pricing](#)
[About Us](#)
[Blog](#)

Download RStudio

Home / Overview / RStudio / Download RStudio

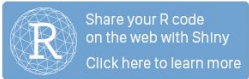
RStudio is a set of integrated tools designed to help you be more productive with R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

If you run R on a Linux server and want to enable users to remotely access RStudio using a web browser please download RStudio Server.

Do you need support or a commercial license? Check out our commercial offerings

RStudio Desktop 0.99.879 — Release Notes

RStudio requires R 2.11.1 (or higher). If you don't already have R, you can download it [here](#).



Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 0.99.879 - Windows Vista/7/8/10	77.1 MB	2016-02-13	c407ebf36c76a42b19b86c9cefaad14
RStudio 0.99.879 - Mac OS X 10.6+ (64-bit)	60 MB	2016-02-13	a3015e9c8fafe0314da908030336925
RStudio 0.99.879 - Ubuntu 12.04+/Debian 6+ (32-bit)	81.6 MB	2016-02-13	afde330661f9565445c522983c2caa8f
RStudio 0.99.879 - Ubuntu 12.04+/Debian 6+ (64-bit)	88.2 MB	2016-02-13	64aa9c53f930d415e4a3dac019cf2c5f
RStudio 0.99.879 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	80.9 MB	2016-02-13	32e35ed218257044385af85b525b6fd
RStudio 0.99.879 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	81.9 MB	2016-02-13	1caa759fd90179930e3840ab5052f89c

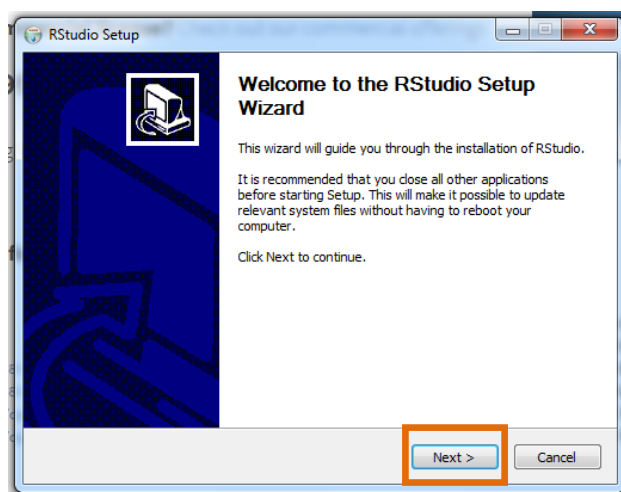
Zip/Tarballs

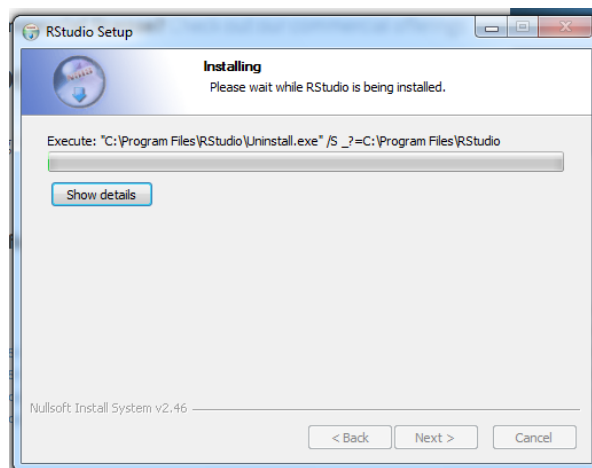
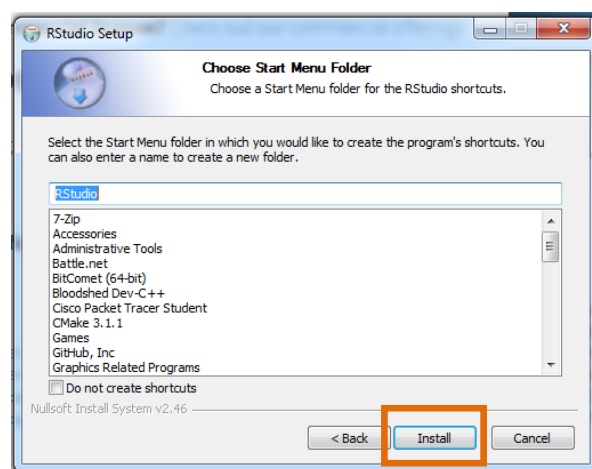
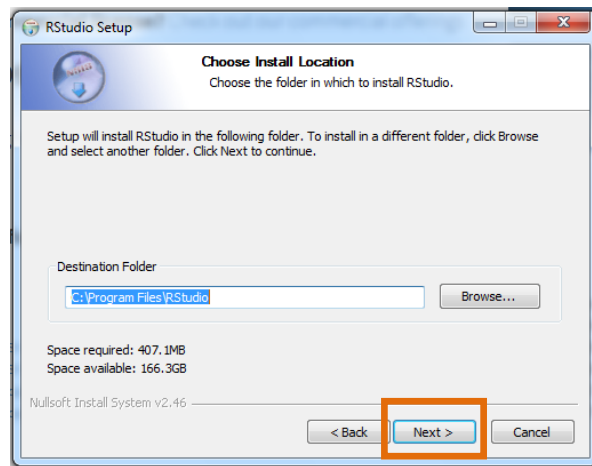
Zip/tar archives	Size	Date	MD5
RStudio 0.99.879 - Windows Vista/7/8/10	110.5 MB	2016-02-13	447fb39b63eb8972375baa5aa18e49d3
RStudio 0.99.879 - Ubuntu 12.04+/Debian 6+ (32-bit)	82.3 MB	2016-02-13	b504cd358c9d3f70312c78772a9e0e1d
RStudio 0.99.879 - Ubuntu 12.04+/Debian 6+ (64-bit)	89.2 MB	2016-02-13	901fb540048a9642067e6f5b17f8073d
RStudio 0.99.879 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	81.6 MB	2016-02-13	08fd5de9e50c21deb59caed80d0e5d96
RStudio 0.99.879 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	82.8 MB	2016-02-13	841c7abb43d4199f2df90c2b2b09c80

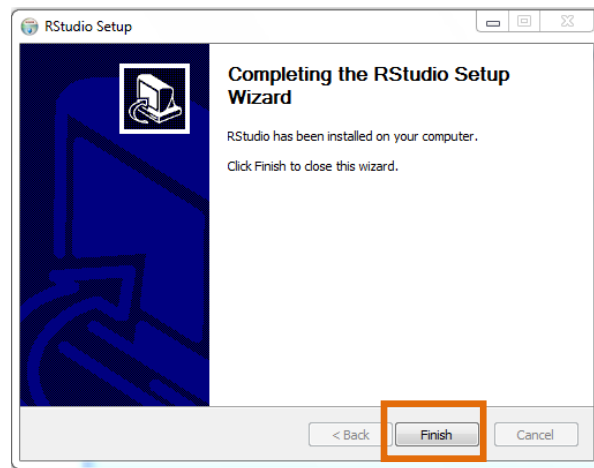
Source Code

A tarball containing source code for RStudio v0.99.879 can be downloaded from [here](#)

3) เลือกที่ต้องการให้ไฟล์อยู่ กด Next ไปเรื่อยๆ และกด Install จากนั้นรอติดตั้งโปรแกรมสำเร็จ

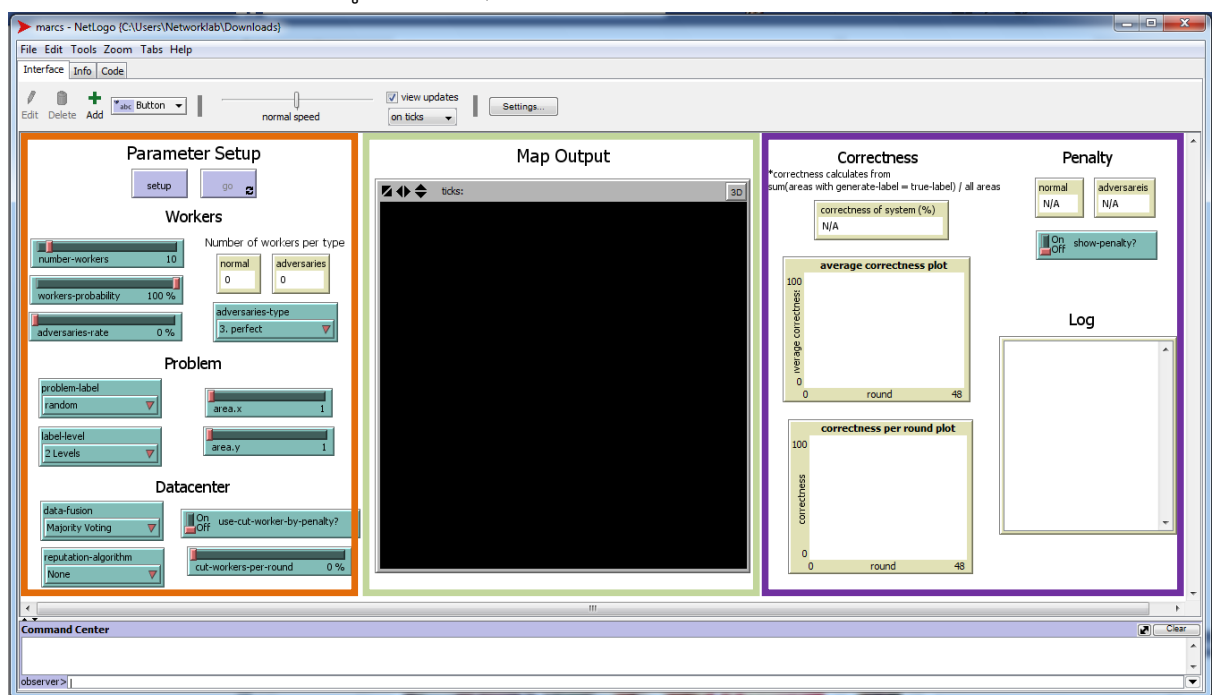




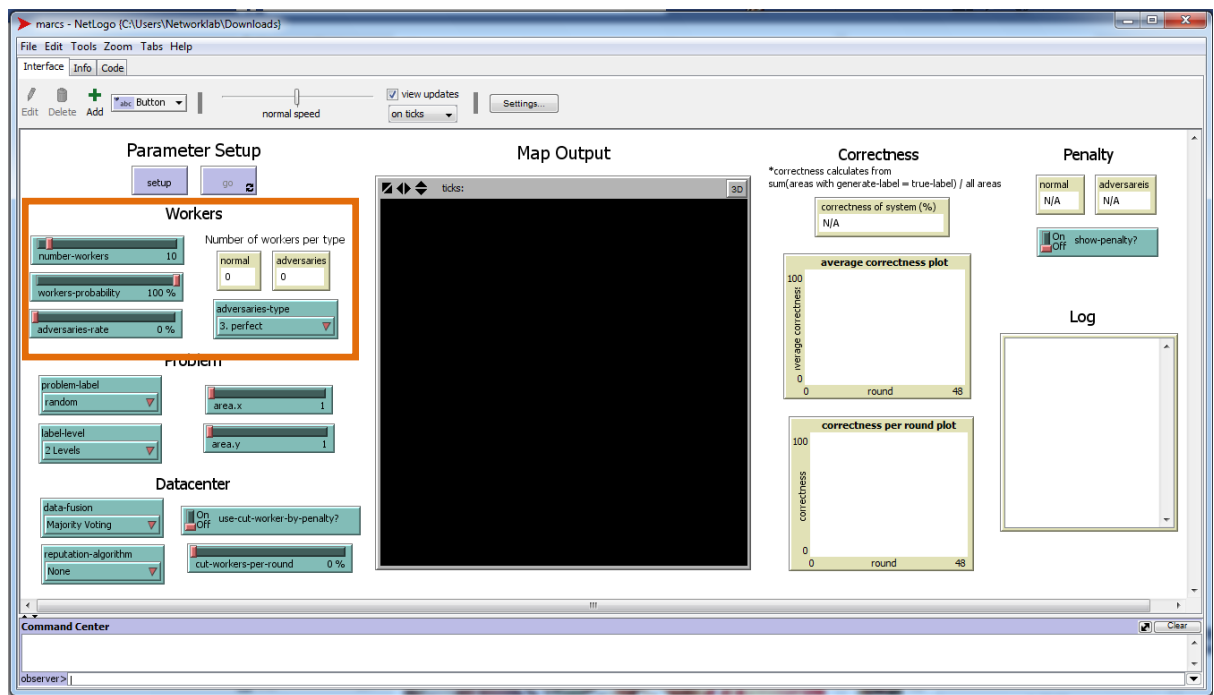


2. คู่มือการใช้โปรแกรมงานอย่างละเอียด

1) เมื่อเปิด source code จะพบกับหน้าจอ ซึ่งด้านซ้าย(กรอบสีส้ม) จะเป็นในส่วนตั้งค่า Parameter หน้าจอตรงกลาง(กรอบสีเขียว) เป็นส่วนที่แสดงให้เห็น Simulation และด้านขวา(กรอบสีม่วง) เป็นส่วนของการประมวลออกมาในรูปแบบต่างๆ



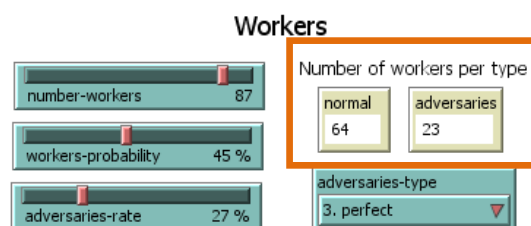
2) ตั้งค่าในส่วนของ Worker Parameter



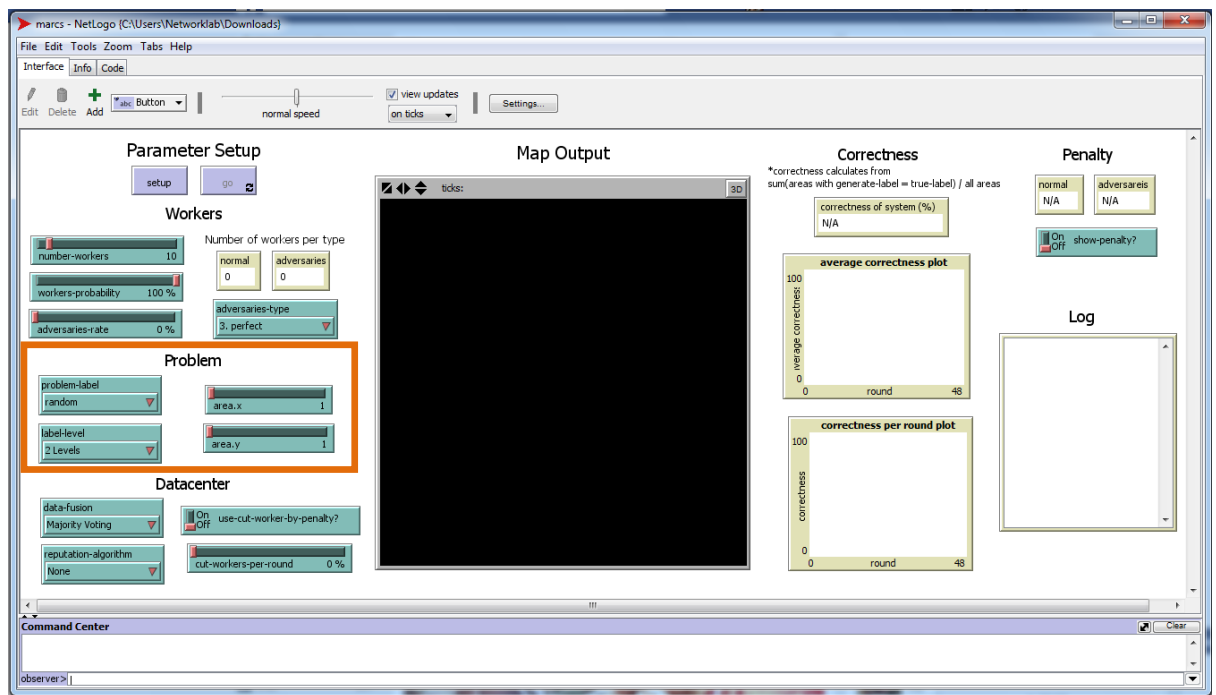
- number-worker คือส่วนกำหนดจำนวน Worker ในระบบเป็นไปได้ตั้งแต่ 1-100 คน
- worker-probability คือเป็นส่วนที่จะระบุ ว่า Normal Worker มีโอกาสที่จะตอบถูกต้อง เป็นกี่เปอร์เซ็นต์
- adversaries-rate คือส่วนที่กำหนดว่าจาก Worker ในระบบที่มีทั้งหมด จะเป็น Adversary Worker ทั้งหมดกี่เปอร์เซ็นต์
- adversaries-type คือการกำหนดประเภทของ Adversary Worker สามารถเลือกได้ ทั้งหมด 4 ประเภท คือ fixed : flood (ก่อกวนโดยรายงานสถานะถนนไม่สามารถใช้งานได้ตลอดเวลา), fixed : normal (ก่อกวนโดยรายงานสถานะถนนสามารถใช้งานได้ตลอดเวลา), random (ก่อกวนโดยรายงานสถานะถนนโดยการสุ่ม) และ perfect (ก่อกวนโดยรายงานสถานะถนนตรงข้ามกับความจริงตลอดเวลา)



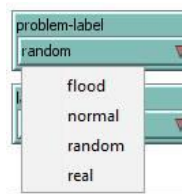
เมื่อตั้งค่าเรียบร้อยแล้ว และกด Set-up จะขึ้นจำนวนของ Worker แต่ละประเภทในช่อง ดังนี้



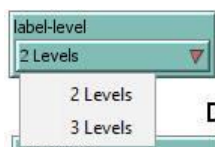
3) ตั้งค่าในส่วน Problem Parameter



- problem-label คือ ส่วนลักษณะปัญหาของ Problem มี 4 ประเภท คือ Flood(Problem มีสถานะน้ำท่วมตลอดเวลา) ,Normal(Problem มีสถานะปกติตลอดเวลา) ,Random(Problem มีสถานะแบบสุ่มตลอดเวลา) และReal(Problem มีสถานะมาจากข้อมูลจริง)

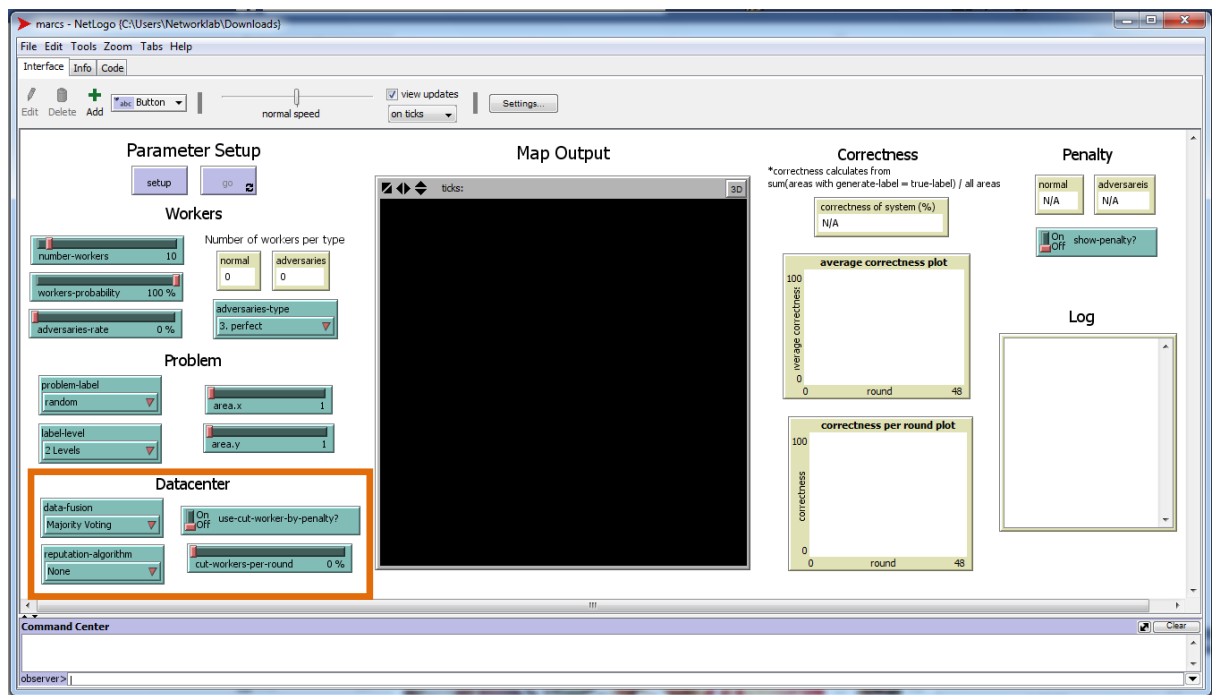


- label-level คือ ในส่วนของ Problem จะมีสถานะที่เป็นไปได้ก็สถานะ มี 2 ประเภทด้วยกัน คือ 2 Level(ถนนสามารถใช้งานได้ และถนนไม่สามารถใช้งานได้) และ 3 Level (ถนนสามารถใช้งานได้ ถนนพอใช้งานได้ และถนนไม่สามารถใช้งานได้)

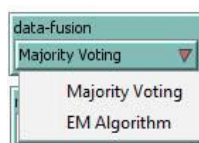


- area คือการกำหนดจำนวนพื้นที่ของ Problem ในการรายงานข้อมูล
area.x คือ การกำหนดด้านย่อยของด้าน x มีค่าตั้งแต่ 1 ถึง 20
area.y คือ การกำหนดด้านย่อยของด้าน y มีค่าตั้งแต่ 1 ถึง 20

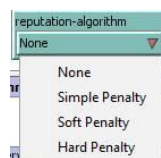
4) ตั้งค่าในส่วน Datacenter Parameter



- Data-fusion คือส่วนของการรวบรวมข้อมูล มี 2 ประเภท คือ Majority Voting (วิธีระบบเสียงข้างมาก) และ EM Algorithm (วิธีค่าคาดหวังสูงสุด)

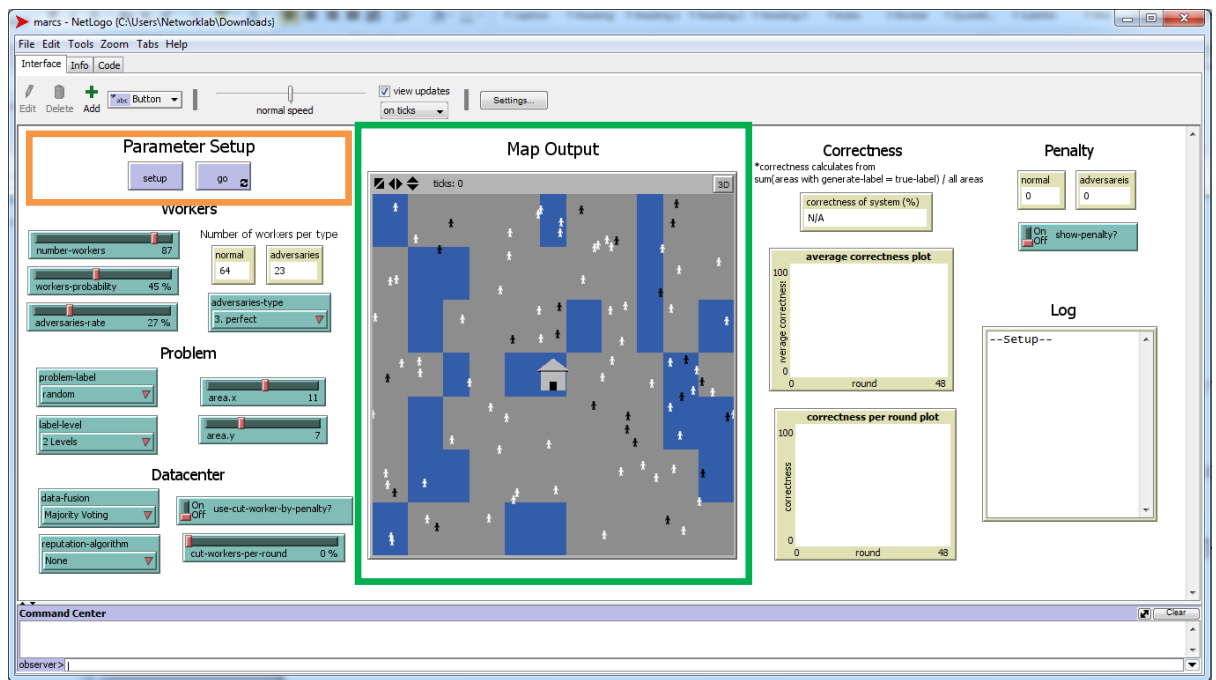


- Reputation-management คือ ส่วนที่จัดการความน่าเชื่อถือของผู้ใช้ มี 4 ประเภท คือ None(ไม่มี Reputation Management) ,Simple Penalty (วิธีการลงโทษอย่างง่าย) ,Soft Penalty (วิธีการลงโทษแบบเบา) และ Hard Penalty (วิธีการลงโทษแบบหนัก)

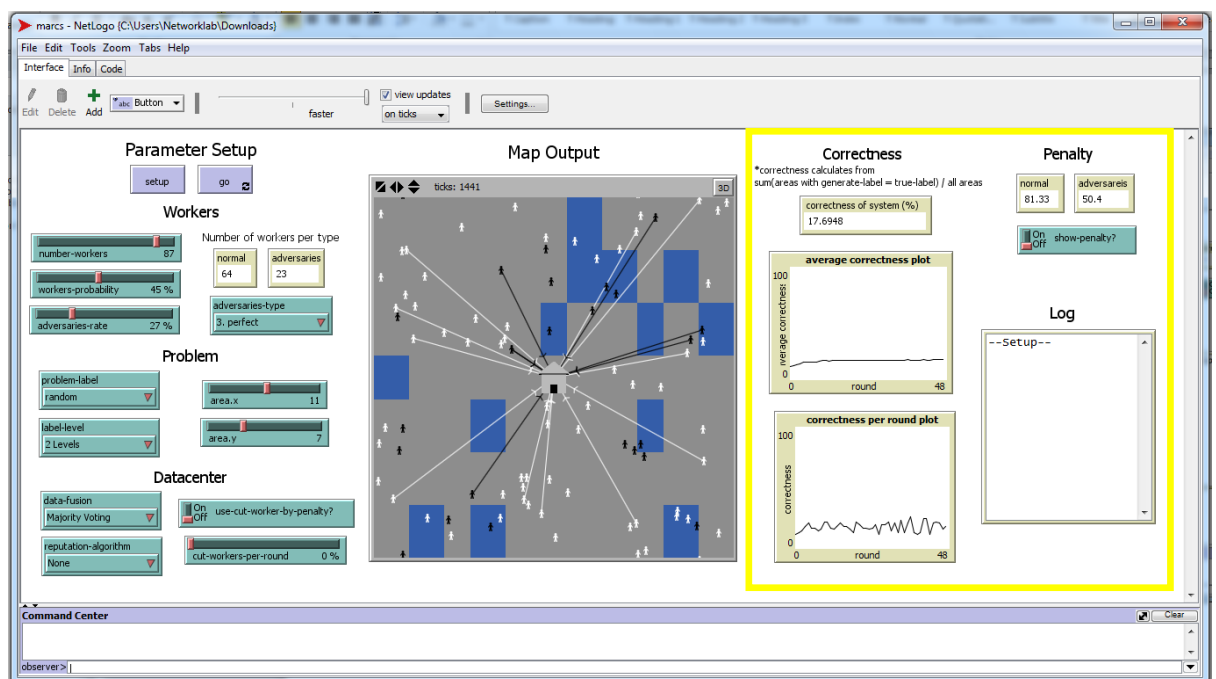


- Use-cut-worker-by-penalty? คือ ต้องการตัด Worker ที่ไม่มีความน่าเชื่อถือออกจากระบบหรือไม่
- cut-worker-per-round คือ เปอร์เซ็นที่ต้องการตัด Worker ที่ไม่มีความน่าเชื่อถือออกจากระบบต่อรอบ

5) หลังจากที่ตั้งค่า Parameter ต่างๆเรียบร้อยแล้ว ให้กด Setup (กรอบสีส้ม) เพื่อให้ Worker และ Problem สุ่มตำแหน่งใน Map Output(กรอบสีเขียว) และกด Go (กรอบสีส้ม) เพื่อเริ่มทำการ Simulation



6) ในส่วนกรอบสี่เหลี่ยมจะเป็นส่วนของการแสดงผลหลังจากการประมวลผลของระบบ



- Correctness of System คือ ค่าความถูกต้องของระบบ โดยความถูกต้องคำนวณจากข้อมูล น้ำท่วมที่แสดงผลออกมาจากระบบมีค่าตรงกับสภาพน้ำท่วมจริงในแต่ละพื้นที่ และคำนวณ ค่าเฉลี่ยความถูกต้องของระบบจากพื้นที่ทั้งหมด
- Average Correctness Plot คือ กราฟแสดงผลค่าเฉลี่ยความถูกต้องของระบบเมื่อเวลาผ่านไป
- Correctness Plot คือ กราฟแสดงผลความถูกต้องของระบบในแต่ละรอบของการประมวลผลข้อมูล

- Penalty คือ ใช้ตรวจสอบค่าความไม่น่าเชื่อถือของผู้ใช้ว่าระบบสามารถให้ค่า Penalty ได้ถูกต้องกับประเภทของ Worker หรือไม่ ถ้าวิธีการให้ค่า Penalty นั้นมีประสิทธิภาพ Normal Worker จะต้องมียค่า Penalty ที่ต่ำกว่า Adversaries
- Show-Penalty? คือ ต้องการแสดงผล Penalty ของ Worker ทุกคนหรือไม่
- Log คือ แสดงผล Flow การทำงานของระบบ