



**ACCIÓN ESTRATÉGICA EN ECONOMÍA Y SOCIEDAD DIGITAL 2013**  
**“INDUSTRIAS DEL FUTURO. INFRAESTRUCTURAS DE INTERNET DEL FUTURO”**

**BIG MARKET**

**Plataforma Big Data para simular y evaluar técnicas de marketing en entornos realistas**

**P5.2 Prototipo de Modelo de Simulación Social para Marketing Viral**

Paradigma Tecnológico (beneficiario)  
Universidad Politécnica de Madrid (subcontratada)



## Tabla de contenidos

1. Introducción.....	7
2. Modelos de simulación social para marketing viral.....	12
2.1. Baseline en el modelo de propagación.....	12
2.2. Nuevo modelo de propagación.....	14
3. Desarrollo del prototipo de modelo de simulación social para marketing viral .....	17
4. Pruebas del prototipo de modelo de simulación social para marketing viral .....	19
4.1. Simulador BigTweet.....	19
4.2. Simulación del escenario de marketing viral.....	26
4.2.1. Pruebas con el modelo M1.....	27
4.2.2. Pruebas con el modelo M2.....	31
5. Resumen final y conclusión.....	36

## Índice de ilustraciones

Ilustración 1: A la izquierda, ejemplo de red sintética de escala libre según el modelo de Barabási Albert (BA). A la derecha, red de mundo pequeño de Watts Strogatz (WS). La oscuridad del color indica un coeficiente de clustering más alto.....	9
Ilustración 2: Diagrama de actividad UML del modelo de propagación de rumores según Tripathy et al. [Tripathy et al., 2010] que es el modelo base M1.....	14
Ilustración 3: Diagrama de actividad UML del modelo de difusión de un nuevo rumor, M2.....	16
Ilustración 4: Arquitectura general de BigMarket.....	17
Ilustración 5: Interfaz gráfica del prototipo BigTweet.....	19
Ilustración 6: Diagrama de clases UML de la implementación de los modelos de difusión.....	20
Ilustración 7: Pantalla principal del Prototipo.....	27
Ilustración 8: Grafo de la red con 1000 usuarios.....	28
Ilustración 9: Estado de los usuarios.....	29
Ilustración 10: Parámetros avanzados del modelo.....	29
Ilustración 11: Grafo resultante.....	30
Ilustración 12: Estados de los usuarios.....	31
Ilustración 13: Grafo obtenido.....	32
Ilustración 14: Usuarios curados y vacunados.....	32
Ilustración 15: Grafo menor número de usuarios.....	33
Ilustración 16: Prueba aumentando probabilidad de infección.....	34
Ilustración 17: Prueba con mayor probabilidad de denier.....	35
Ilustración 18: Estados de los usuarios resultantes.....	35

## Resumen ejecutivo

Este entregable aplica la metodología de E5.1 a un caso concreto. Vistas las oportunidades de negocio en simular fenómenos de marketing viral como la difusión de rumores en Twitter, se optó por desarrollar un modelo para modelar la propagación de un mensaje de marketing y ver su impacto en la captación de clientes. Algunos de los elementos concretos a modelar identificados en este caso son: los distintos estados por los que puede pasar un agente en el campo de la propagación de rumores; la evolución de estos estados en función del estado de los contactos en la red social; y cómo evolucionan métricas globales de la sociedad artificial tales como el número de agentes apoyando el rumor o negándolo. El prototipo resultante de la implementación del modelo y presentado aquí se llama BigTweet.

## Información del Documento

<b>Proyecto FIT Número</b>	TSI-100102-2013-80	<b>Acrónimo</b>	Big Market
<b>Título completo</b>	Plataforma Big Data para simular y evaluar técnicas de marketing en entornos realistas		
<b>URL</b>			
<b>URL del documento</b>			

<b>Entregable</b>	<b>Número</b>	P5.2	<b>Título</b>	Prototipo de Modelo de Simulación Social para Marketing Viral
<b>Paquete de Trabajo</b>	<b>Número</b>	5	<b>Título</b>	Simulación social basada en Big Data
<b>Tarea</b>	<b>Número</b>	T5.2	<b>Título</b>	Modelo de Simulación Social para Marketing Viral

Fecha de Entrega	Contractual	31/03/15	Entregado			
Estado	Versión final 27/03/2015		Final			
Tipo	Informe	Demo	X	Otro		
Nivel de Diseminación	Público				X	Consorcio
Resumen (para diseminación)	El objetivo de este documento es presentar el prototipo del modelo de propagación de rumores para evaluar técnicas de marketing viral sobre él.					
Palabras Clave	Justificación					

<b>Autores (Partner)</b>	UPM			
<b>Responsable de Autoría</b>	<b>Partner</b>	UPM	<b>Email</b>	cif@gsi.dit.upm.es

# 1. Introducción.

El objetivo de este entregable es presentar el prototipo PT5.1 de BigMarket, cuyo objetivo es el desarrollo de un modelo de simulación de marketing viral.

Como se ha explicado en el documento E5.1, las decisiones comunes en el diseño del modelo basado en agentes son [Rand and Rust, 2011]: alcance del modelo, definición de agentes, propiedades de los agentes, comportamientos del agente, entorno, paso del tiempo y salida. Éstos se detallan a continuación:

1. *Alcance del modelo.* Esta es la parte del sistema de destino en el que se enfoca el modelo y se determina qué aspectos pueden ser ignorados. La idea de desarrollar un modelo es que debe ser lo más simple posible para estudiarlo fácilmente, pero al mismo tiempo, el modelo debe describir la realidad. Por lo tanto, existe un equilibrio entre el enfoque de KISS ("Keep it Simple Stupid" o "Keep it Simple Stupid" de una manera más cortés) y el enfoque de KIDS ("Keep it Descriptive Stupid") [Serrano y Botia, 2013]. En la experiencia de los autores, un error importante al usar métodos de investigación basados en la simulación social basada en agentes (ABSS) es tratar de modelar el mundo en lugar de centrarse sólo en los aspectos relevantes de la investigación. En cuanto al caso de Twitter, hay una serie de fenómenos que pueden ser interesantes para fines de marketing: propagación de retweets, número de menciones, número de tweets con un hashtag específico, actividad por zona horaria, etcétera. No obstante, con respecto a la propagación de la información errónea, el enfoque principal no es el modelado de la evolución de los mensajes (o tweets), sino la evolución del estado de los usuarios con respecto a un rumor o rumor específico como se muestra en E5.1. Estos estados suelen seguir la terminología epidemiológica: infectados, curados, etcétera.

2. *Agentes.* Otra decisión importante es lo que los agentes representan en la ABSS. Obsérvese que los agentes no significan necesariamente agentes inteligentes [Nwana, 1996] capaces, entre otras cosas, de aprender; ni agentes deliberativos [Woolridge, 2001] que toman decisiones usando el razonamiento simbólico. El agente ABSS típico es un agente reactivo que interactúa autónomamente con otros, basado en un modelo de comportamiento que puede ir desde el uso de reglas de producción hasta el uso de modelos de aprendizaje automático tales como redes neuronales artificiales [Holubiec et al., 2012]. En el caso de Twitter, la decisión directa es tener agentes por cada usuario de Twitter. Por otra parte, en el caso de la difusión del chisme, hay

generalmente usuarios especiales capaces de ejecutar estrategias de comercialización tales como lanzar contrarumores.

3. *Propiedades*. Estos son los campos que describen cada agente. Una vez más, estos dependen completamente del alcance del modelo. Para el caso de difusión de rumores de Twitter, como se explica, las propiedades típicas incluyen: un identificador; una posición en el medio ambiente (explicado más adelante); el estado del agente con respecto al rumor (infectado, curado, vacunado, etcétera); y, si es necesario, el campo del tipo de agente que puede determinar el alcance de otras propiedades o el comportamiento del agente.

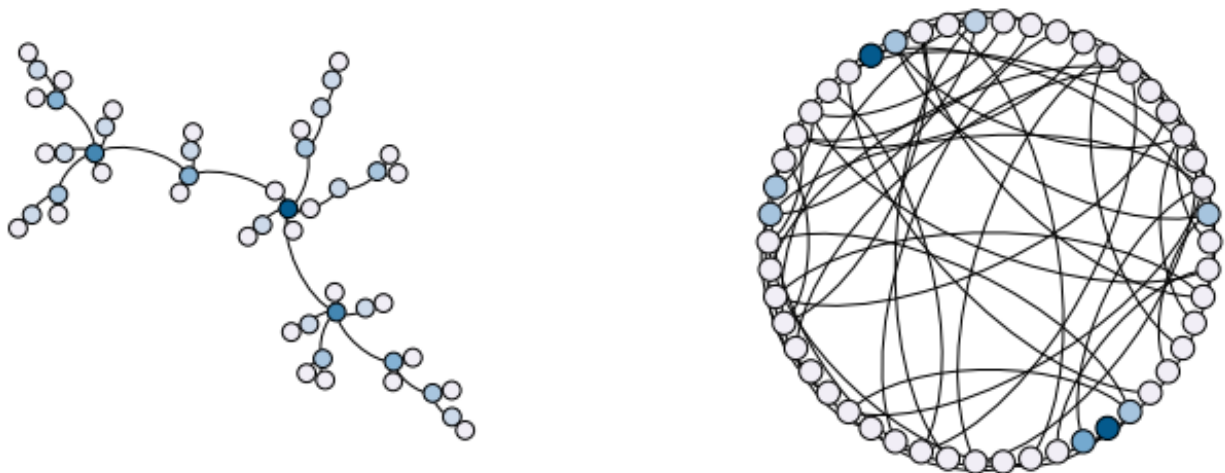
4. *Comportamientos*. Los agentes exhiben un comportamiento que implica interactuar con el medio ambiente y otros agentes cada paso del tiempo. Comúnmente, estos comportamientos son procesos estocásticos que dependen de probabilidades dadas. Existen innumerables maneras de definir comportamientos en ABSS: reglas de producción [Serrano et al., 2014b], modelos de aprendizaje mecánico [Serrano et al., 2013], funciones de densidad de probabilidad [Garcia-Valverde et al., 2012], etc. Para el caso de propagación de desinformación de Twitter, el enfoque principal observado en la literatura especializada (véase E5.1) es definir textualmente el comportamiento del agente. Esto siempre lleva a lagunas al programar el modelo especificado. Por lo tanto, los autores recomiendan el uso de pseudo-código o cualquier otra técnica general de modelado de software para software general como diagramas UML [Challenger et al., 2014]. El uso de diagramas de flujo, que corresponden aproximadamente a los diagramas de actividad de UML, son muy populares en la literatura de ABSS [Gilbert y Troitzsch, 2005].

5. *Entorno*. El entorno define la topología de interacción de los agentes. Para los trabajos de Twitter, el entorno se describe ampliamente como una red o un grafo<sup>1</sup> donde los nodos representan a los usuarios. Otros tipos de redes también son posibles dependiendo del alcance, como las redes de retweets. Los enlaces no tienen el mismo significado en todos los trabajos. Mientras que algunos autores representan la asimetría en Twitter (que el usuario u1 siga a u2 no significa que u2 siga a u1), otros consideran enlaces no dirigidos ya que Twitter tiene mecanismos para hacer que el flujo de información del seguidor a lo seguido (como respuestas, menciones, retweets y mensajes privados).

<sup>1</sup>Los términos grafo (compuesto de vértices y aristas o arcos) red (compuesto de nodos y enlaces) se usan de forma intercambiable en este documento. En la literatura de ciencias sociales, los términos actores y relaciones se usan ampliamente para los elementos de la red o grafo.



Otra decisión es si utilizar una red real o una sintética. En cuanto al uso de redes reales (las que tienen nodos y enlaces extraídos de Twitter), hay dos enfoques principales. El primero es el uso de redes egocéntricas que examinan sólo los vecinos inmediatos de un nodo de la semilla y sus interconexiones asociadas. El trabajo de Gatti et al. [Gatti et al., 2013], revisado en E5.1, sigue este enfoque usando la cuenta de Twitter del Presidente Obama como semilla. El segundo enfoque para el uso de redes reales es: recopilar tweets independientemente de sus autores, reunir a los seguidores y amigos<sup>2</sup>, y usar el componente conectado más grande en el gráfico resultante como red final. Como se explicó en E5.1, este enfoque se utiliza en el trabajo de Seo et al. [Seo et al., 2012].



*Ilustración 1: A la izquierda, ejemplo de red sintética de escala libre según el modelo de Barabási Albert (BA). A la derecha, red de mundo pequeño de Watts Strogatz (WS). La oscuridad del color indica un coeficiente de clustering más alto.*

En cuanto al uso de redes sintéticas, Barabási Albert (BA) es la opción más popular al modelar las redes sociales [Liu y Chen, 2011]. Aunque la naturaleza de escala libre de un gran número de redes sigue siendo debatida por la comunidad científica, las redes sociales como Twitter se dice ampliamente que no tienen escala. En pocas palabras, la creación de estas redes se lleva a cabo bajo el supuesto de que la probabilidad de que un usuario  $u_1$  se conecte a otro usuario  $u_2$  depende del número de conexiones que  $u_2$  ya tiene. Esto hace que los hubs aparezcan, es decir, nodos con un grado que excede en gran medida el grado medio. Ver un ejemplo de gráfico BA en la figura a la izquierda. Otra opción para el caso del rumor es el uso de las redes de pequeños mundos de Watts Strogatz (WS), como en el trabajo presentado por Tripathy et al. [Tripathy et al., 2010], donde: *isi*  $u_1$  está conectado a  $u_2$  y  $u_2$  está conectado a  $u_3$ , es probable que  $u_1$  y  $u_3$  también están vinculados. Esto hace que estas redes tengan un alto coeficiente de agrupamiento, es decir, una

<sup>2</sup> La terminología de la API de Twitter tiene seguidores y amigos, donde los amigos son los usuarios seguidos.

medida del grado en que los nodos de un gráfico tienden a agruparse. Vea un ejemplo del gráfico WS en la figura a la derecha. Como en la descripción del comportamiento de los agentes, un inconveniente común en la literatura es no dar suficiente información para reproducir estas redes sintéticas. El modelo general, el algoritmo para generarlo, y los parámetros del algoritmo; son necesarios para garantizar la reproducibilidad.

Aunque el uso de redes realistas es siempre deseable, hay una clara razón para la línea hegemónica de usar redes sintéticas: las redes realistas restringen los usuarios estudiados; y esto restringe el número de tweets enviados por estos usuarios. Por lo tanto, si la investigación depende tanto de los tweets semánticos como en la propagación de rumores, el uso de redes realistas lleva a tener menos mensajes para trabajar. Como resultado, el realismo de los usuarios simulados es considerablemente inferior. E6.1 ilustrará este trade-off entre el realismo de la red y los usuarios por conjuntos de datos reales y públicos para el caso de difusión de rumores. Una razón adicional para utilizar las redes sintéticas explicadas para el caso de propagación de chismes, es que la información se extiende muy rápidamente en ellas [Abraham et al., 2010]. Más específicamente, las redes WS tienen una longitud de trayecto promedio muy baja, es decir, el número medio de pasos a lo largo de los caminos más cortos para todos los posibles pares de nodos de red; Y, la longitud promedio de las trayectorias de las redes de BA crece muy poco con el número de nodos debido a los centros. Como consecuencia, estas redes son fuertes adversarios para las estrategias de control de chismes y proporcionan a los investigadores una línea de base asumiendo que el rendimiento de las estrategias es mejor que en las redes reales.

6. *Salto temporal.* Los ABSS típicamente evolucionan en el tiempo usando un paso de tiempo. Se distinguen dos fases: la inicialización, cuando se crean los agentes y el entorno; e, iteración, donde los agentes actúan según su modelo de conducta. Además, dependiendo del alcance y de los datos reales disponibles, el paso del tiempo representará una unidad de tiempo física diferente. Para el caso de Twitter, la captura de datos y las tareas exploratorias de análisis de datos pueden dar una idea de esta decisión. Si los datos están muy dispersos (por ejemplo, los días pasan entre tweets relevantes), no hay suficiente información para un paso corto (por ejemplo, simulando horas).

7. *Entrada y salida.* Los parámetros y los valores observados en la ejecución de la simulación son otras decisiones para el modelo. Una de las entradas más importantes y comúnmente utilizadas en ABSS es la semilla aleatoria. Como se ve, una ABSS involucra una serie de procesos estocásticos: seleccionar el orden de ejecución de agentes en un paso de tiempo, crear un modelo de red, decidir entre posibles acciones en el modelo de comportamiento, etcétera. Un defecto importante en la

investigación ABSS es el no garantizar que todos estos procesos dependen de una única semilla aleatoria, perdiendo la repetibilidad de la simulación y la reproducibilidad. En cuanto a la salida en el caso de propagación de desinformación de Twitter, la salida común es el número de agentes por posible estado de rumor (infectado, curado, etcétera) y por paso de tiempo.

## 2. Modelos de simulación social para marketing viral

### 2.1. Baseline en el modelo de propagación

Esta sección revisa las decisiones explicadas para reproducir el enfoque de Tripathy et al. [Tripathy et al., 2010] para modelar la difusión de s en Twitter, llamémosla M1. El enfoque base se basa en el modelo en cascada [Weng et al., 2013]. Los agentes son usuarios de Twitter con una propiedad de estado que puede ser: neutral (estado inicial); infectados (crean la desinformación); vacunados (cree que el anti-rumor antes de ser infectado); O, curado (cree el anti-rumor después de ser infectado). El comportamiento básico implica: (1) inicializar un número de usuarios infectados; (2) cada usuario infectado en el tiempo  $t$  intenta infectar cada uno de sus vecinos no infectados con una probabilidad dada (*propInfect*); (3) después de un retardo dado (*timeLag*), un nodo infectado al azar inicia una propagación de un anti-rumor a sus vecinos, tratando de curar o vacunarlos con una probabilidad (*probAcceptDeny*) cada vez que el paso  $t$ ; y, finalmente, (4) los usuarios curados y vacunados también tratan de curar o vacunar a sus vecinos con una probabilidad (*probAcceptDeny*) cada vez que el paso  $t$ . Como se recomienda, la figura 2 muestra un diagrama de actividad UML para este comportamiento.

El entorno es una red sintética de escala libre de BA, ya que, como se explica en el epígrafe del entorno de E5.1, son fuertes adversarios de las estrategias de control de chismes y proporcionan a los investigadores una línea de base, ya que se supone que el rendimiento de las estrategias es mejor que en redes reales. Para dar más información para reproducir este entorno, este trabajo experimenta con nodos de 1K y un máximo de 10 enlaces inicialmente añadidos por cada nuevo nodo. Más específicamente, se ha empleado el generador gráfico de fijación preferencial Barabási-Albert del proyecto de flujo de gráficos<sup>3</sup>. En contraste con el ejemplo dado en la figura 1, los autores han experimentado con las redes de BA donde cada nodo añadido viene con varios enlaces porque esto no sólo es más realista (los usuarios de Twitter se ven obligados a seguir varias cuentas de Twitter al principio), pero también hace que la información se difunda más rápidamente.

Se supone un paso de tiempo de una hora, y la salida es el número de usuarios que endosan la desinformación (con infectado como estado) y el número de usuarios que lo niegan (con vacunados o curados como estados). Con los detalles dados, los parámetros de entrada son los siguientes: semilla aleatoria, número de usuarios, enlaces máximos por nodo (para la construcción

<sup>3</sup> Sitio web del proyecto GraphStream: <http://graphstream-project.org/>

de la red BA), usuarios inicialmente infectados, probabilidad de infección, probabilidad de aceptar una negación y retraso temporal.

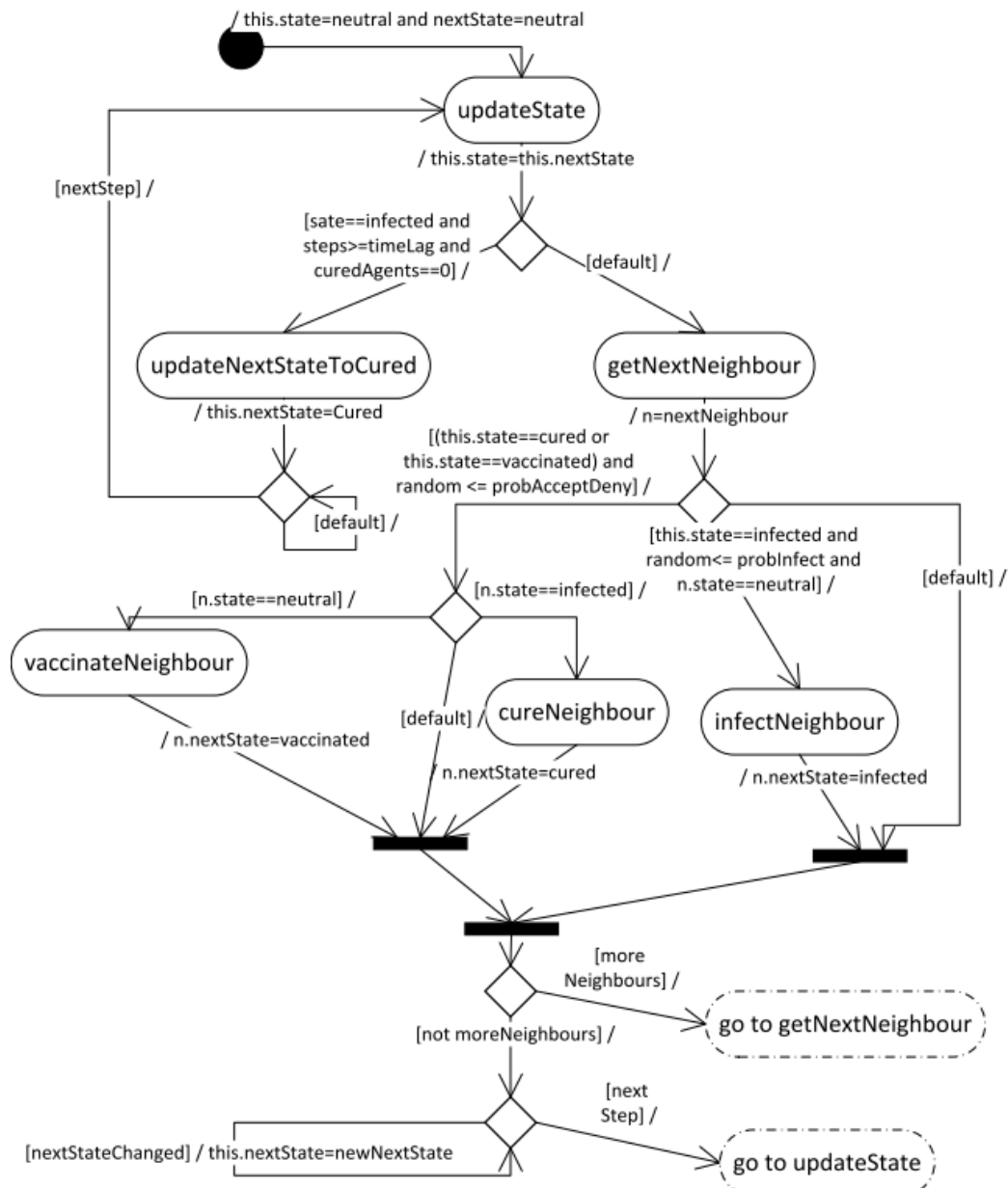


Ilustración 2: Diagrama de actividad UML del modelo de propagación de rumores según Tripathy et al. [Tripathy et al., 2010] que es el modelo base M1

## 2.2. Nuevo modelo de propagación

Después de estudiar las ejecuciones de M1 y datos reales de Twitter, los autores descubrieron que el concepto de "curado" apenas se valida en la propagación de rumores de Twitter. Por un lado, si la recuperación de los tweets sobre rumores (o anti-rumores) es de un tema específico, toda la información para la mayoría de los usuarios proviene de un solo tweet que dice si el usuario está

apoyando o negando la información errónea. Por otro lado, psicológicamente, los usuarios infectados que cometen un error, pueden no ser tan entusiastas como se asume en M1 sobre la difusión de sus equivocaciones con anti-rumores. M2 modificó el modelo M1 para incluir esta idea al permitir que los usuarios vacunados (los que no han sido infectados previamente) difundan anti-rumores.

Otra idea incluida en M2 es que, independientemente de cualquier desfase temporal, un nodo neutro que tenga un vecino infectado, puede convertirse en un usuario vacunado si este nodo sabía de cualquier información externa que la desinformación era falsa. Con esto en mente, M2 modifica M1 para incluir una probabilidad de hacer un *denier*, es decir, convertir un usuario neutro en un usuario vacunado al propagar un rumor.

Así, el comportamiento es: (1) inicializar un número de usuarios infectados; (2) cada usuario infectado en el tiempo  $t$  intenta infectar cada uno de sus vecinos no infectados con una probabilidad dada (*propInfect*); (3) en lugar de infectarlos, estos vecinos pueden ser vacunados si son neutrales con una probabilidad (*propMakeDenier*); Y, finalmente, (4) los usuarios vacunados (pero no los usuarios curados) intentan curar o vacunar a sus vecinos con una probabilidad (*probAcceptDeny*) cada vez que el paso  $t$ . Como se recomienda, la figura 3 muestra un diagrama de actividad UML para este comportamiento.

Con los detalles dados, los parámetros de entrada de M2 son los mismos de M1 excepto el retardo que se reemplaza con *propMakeDenier*: semilla aleatoria, número de usuarios, enlaces máximos por nodo (para la construcción de la red BA), usuarios inicialmente infectados, probabilidad de infección, Probabilidad de aceptar una negación, y probabilidad de hacer un *denier*. Con respecto a la producción, los agentes curados se cuentan en M2 como usuarios que avalan el chisme junto con los agentes infectados; y sólo los agentes vacunados se cuentan como usuarios que niegan la desinformación.



Ilustración 3: Diagrama de actividad UML del modelo de difusión de un nuevo rumor, M2

### 3. Desarrollo del prototipo de modelo de simulación social para marketing viral

El prototipo desarrollado en este entregable se corresponde con el modelo de retención que muestra la propagación de una campaña de márketing, basado en simulación social.

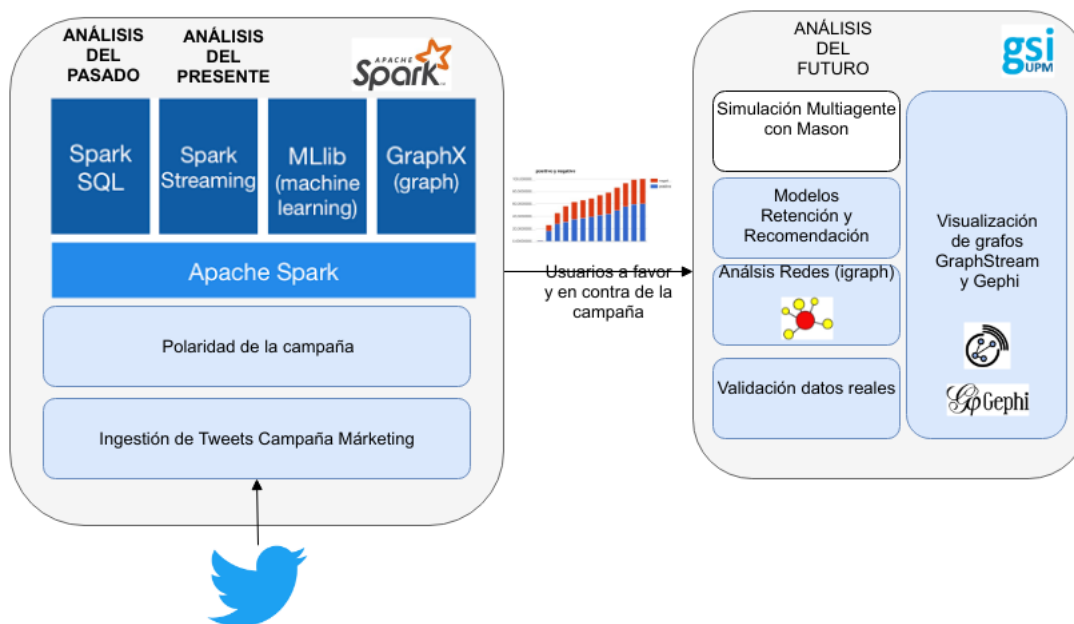


Ilustración 4: Arquitectura general de BigMarket

Como se explicó en E5.1, esta tarea consiste en traducir el modelo en algo que puede ser utilizado por un ordenador (es decir, la programación del modelo). Hay un gran número de frameworks de software de modelado basado en agentes [wik, a] que ayudan a los desarrolladores a implementar estos sistemas. Algunas de las características típicas incluyen: ejemplos de simulación; agentes predefinidos; algoritmos de programación; y, soporte para experimentos por lotes. Algunas de las plataformas más extendidas para el desarrollo de ABSS son NetLogo [Tisue y Wilensky, 2004], Mason [Luke et al., 2004] y Repast [North et al., 2006].

Aunque los marcos mencionados tienen soporte para modelar y mostrar redes, las herramientas de análisis de redes sociales (SNA) [Abraham et al., 2010] son claramente superiores en la visualización y el estudio de gráficos a gran escala. Esto se debe, entre otras cosas, a: las métricas de centralidad listas para usar, que identifican los vértices más importantes dentro de un gráfico; los métodos de detección de la comunidad, que pueden utilizarse para dividir la red en grupos bien definidos densamente conectados internamente; y, lo que es más importante, la variedad de



algoritmos de trazado de gráficos dirigidos por fuerzas que estos paquetes implementan, permitiendo una visualización de red más comprensible. Las pantallas de simulación comprensibles son cruciales porque proporcionan el mecanismo básico para verificar la ABSS, es decir, comprobar que cumple con la especificación del modelo y los requisitos. Algunos marcos de SNA populares [wik, b] son: Gephi, iGraph, GraphStream y NetworkX.

Uno de los requisitos de implementación de ABSS más importantes y comúnmente olvidados es ofrecer repetibilidad y reproducibilidad. En una ABSS, hay una serie de procesos estocásticos tales como: las posiciones y estados iniciales de los agentes; el orden de que los agentes ganen el turno de ejecución; el orden en que se visitan los vecinos de los agentes; o, la evolución del estado de los agentes. Al combinar diferentes paquetes de software como los marcos ABSS y SNA, los desarrolladores deben garantizar una única y parametrizada semilla aleatoria para controlar estos procesos. Una práctica recomendada es el uso de pruebas unitarias [Khamis y Nagi, 2013]. Esto permite a los desarrolladores probar automáticamente que cada unidad de simulación individual no devuelve salidas diferentes cuando se utiliza la misma semilla aleatoria como entrada.

## 4. Pruebas del prototipo de modelo de simulación social para marketing viral

### 4.1. Simulador BigTweet

Para el problema de difusión de rumores, se ha construido una implementación de los modelos de propagación presentados en la sección anterior. Este simulador se llama BigTweet. La Figura 4 muestra la GUI del simulador que incluye: la pantalla de red simulada; un gráfico con el número de agentes por estado posible; y, el marco de la consola. El marco de la consola permite controlar la ejecución de un experimento (iniciar, para, detener); y seleccionar parámetros (semilla aleatoria, modelo de propagación, probabilidad de infección, etc.).

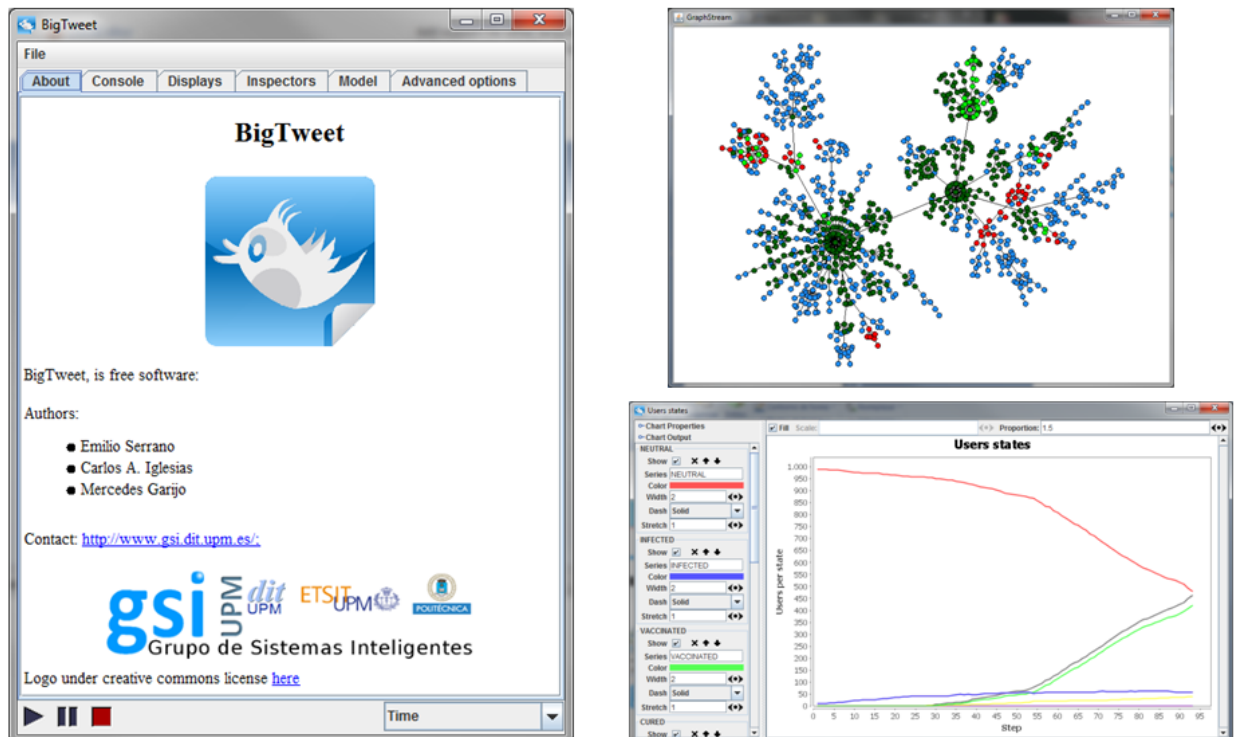


Ilustración 5: Interfaz gráfica del prototipo BigTweet

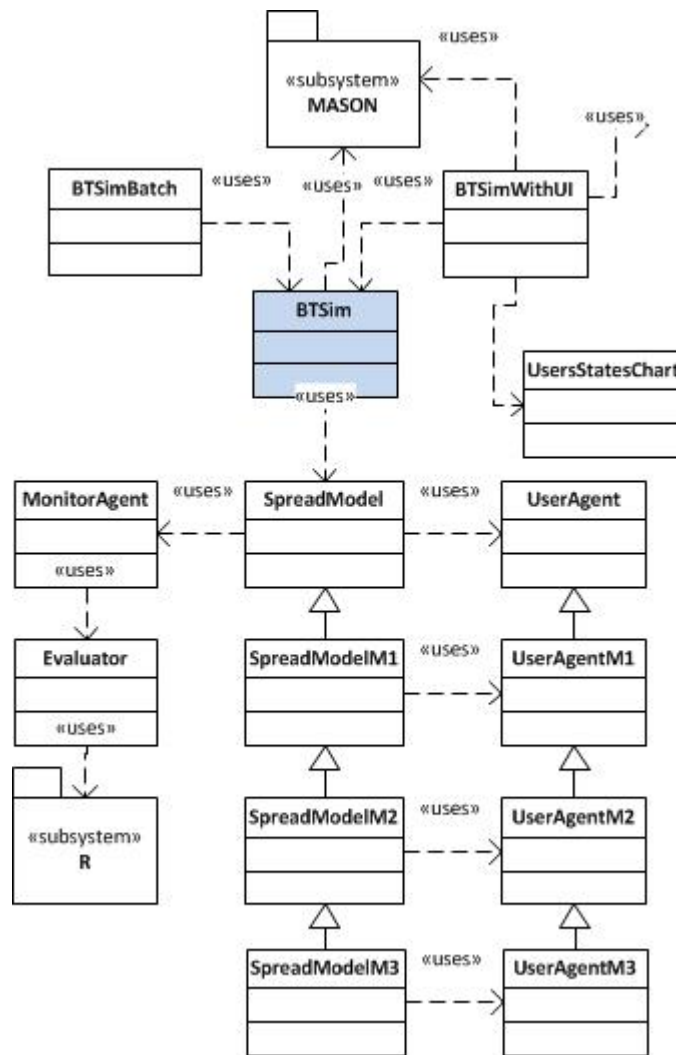


Ilustración 6: Diagrama de clases UML de la implementación de los modelos de difusión

Como se muestra en el diagrama de clases BigTweet UML que se muestra en la figura 5, BigTweet emplea el simulador social Mason [Luke et al., 2004]. BigTweet es un software libre y de código abierto y está disponible en línea bajo una licencia GPL (consulte el sitio web de material adicional [Serrano et al., B] o el sitio web de BigTweet GitHub [Serrano et al., A]).

#### 4.1.1. Instalación de la máquina virtual

En primer lugar deberá instalarse el software VirtualBox para poder ejecutar la máquina virtual proporcionada. Para ello accedemos a la página web oficial de VirtualBox () y seleccionaremos la versión acorde a nuestro sistema operativo.



Ilustración 7: Página de descargas de VirtualBox

## VirtualBox binaries

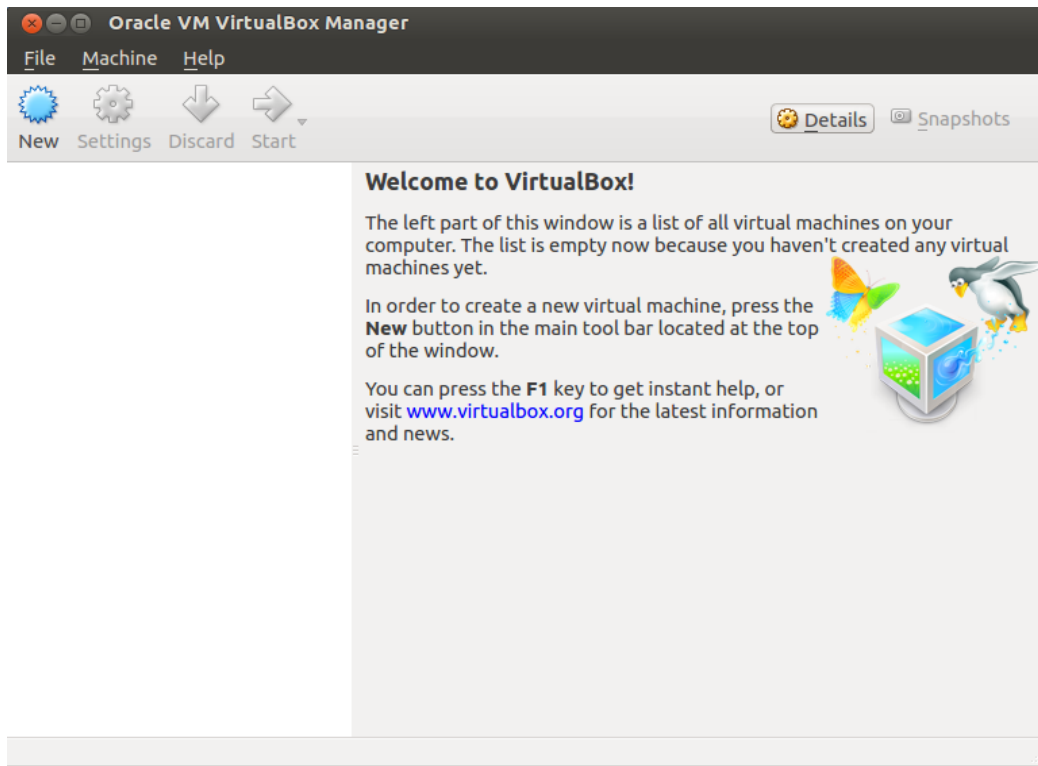
By downloading, you agree to the terms and conditions of the respective license.

- **VirtualBox platform packages.** The binaries are released under the terms of the GPL version 2.
  - **VirtualBox 5.1.8 for Windows hosts** → [x86/amd64](#)
  - **VirtualBox 5.1.8 for OS X hosts** → [amd64](#)
  - **VirtualBox 5.1.8 for Linux hosts**
  - **VirtualBox 5.1.8 for Solaris hosts** → [amd64](#)

Ilustración 8: Ejecutables de VirtualBox

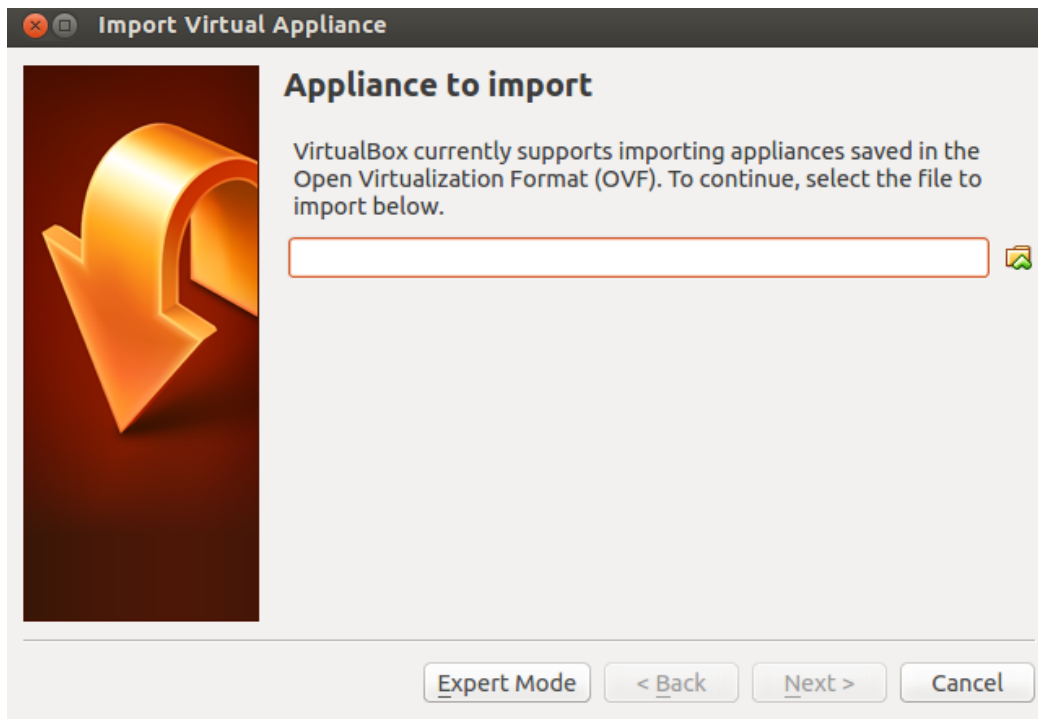
Una vez seleccionado el sistema operativo y su versión se iniciará la descarga. A continuación procederemos a la instalación de la herramienta ejecutando el fichero descargado previamente. Una vez completado el proceso de instalación ejecutamos el entorno VirtualBox.

Puede consultar los errores más frecuentes encontrados en este proceso de instalación en la sección FAQ - VirtualBox dentro de este documento, o bien en la propia web de VirtualBox en el .



*Ilustración 9: Asistente de VirtualBox*

Procedemos a importar la máquina virtual proporcionada *bigtwitter.ova* utilizando el menú de navegación superior File > Import Appliance



*Ilustración 10: Asistente de VirtualBox para realizar la importación de OVF*

Una vez seleccionado el fichero procederemos a configurar la máquina virtual de acuerdo con el rendimiento y especificaciones técnicas del equipo en el que se quiera instalar. Por defecto vienen asignados 4096MB de memoria RAM, pero en caso de ser necesario podrían reducirse hasta 2048 MB.

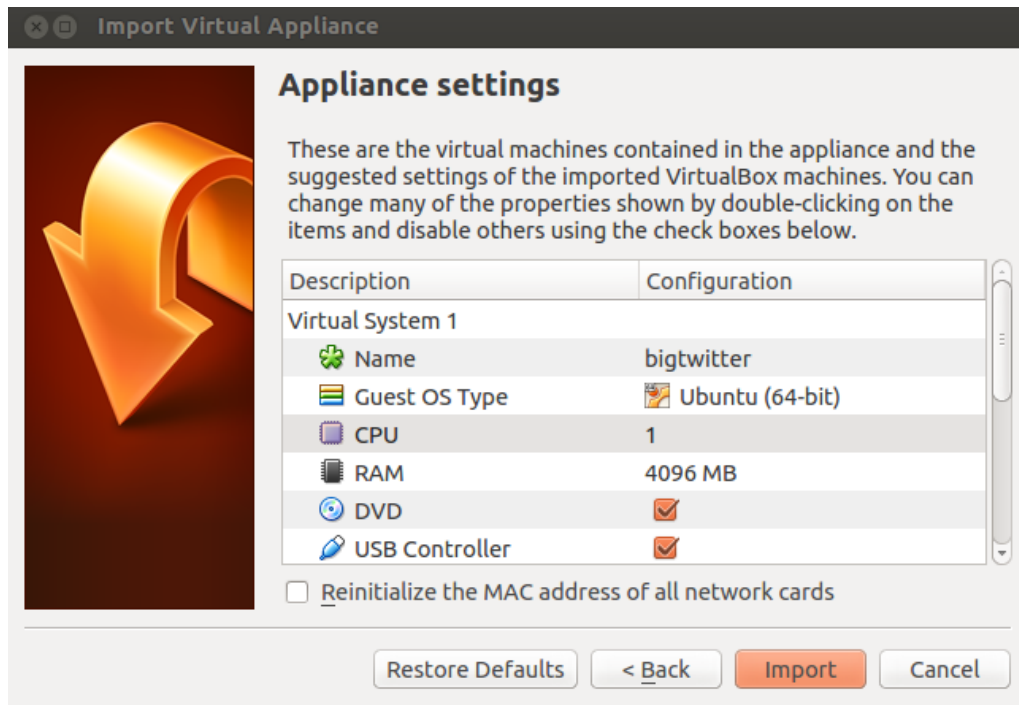


Ilustración 11: Configuración de la máquina virtual

Una vez adaptada la configuración estándar de la máquina virtual le damos a Import. Esta operación puede tardar unos minutos.

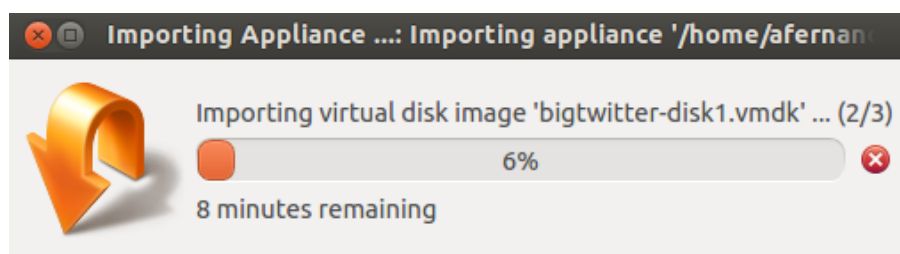


Ilustración 12: Proceso de importación de la máquina virtual

Una vez importada la máquina virtual correctamente debería aparecernos en el gestor de VirtualBox.

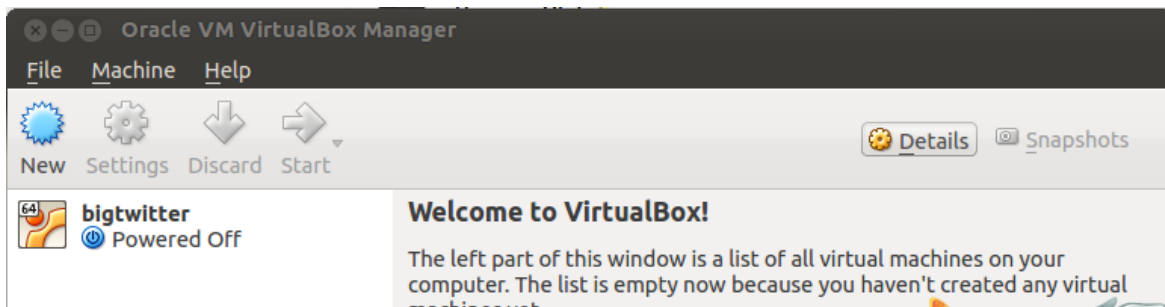


Ilustración 13: Resultado esperado tras importar la máquina virtual

Ejecutamos la máquina virtual haciendo doble click o bien Click derecho > Start > Normal Start y se abrirá una nueva ventana con la máquina virtual iniciada. Una vez finalice su configuración observará el Escritorio mostrado en la imagen. En caso de preguntarle los credenciales de usuario deberá introducir Usuario: **bigtweet**, Contraseña: **bigtweet**

#### 4.1.2. Ejecución del entorno BigTweet

Una vez arrancada la máquina virtual pasamos a detallar los pasos a seguir para ejecutar la aplicación BigTweet. Simplemente haga doble click en el icono del Escritorio “LXTerminal”, pegue la siguiente línea y presione la tecla **Enter**.

```
cd Escritorio/BigTweet && java -jar bigtwitter.jar
```

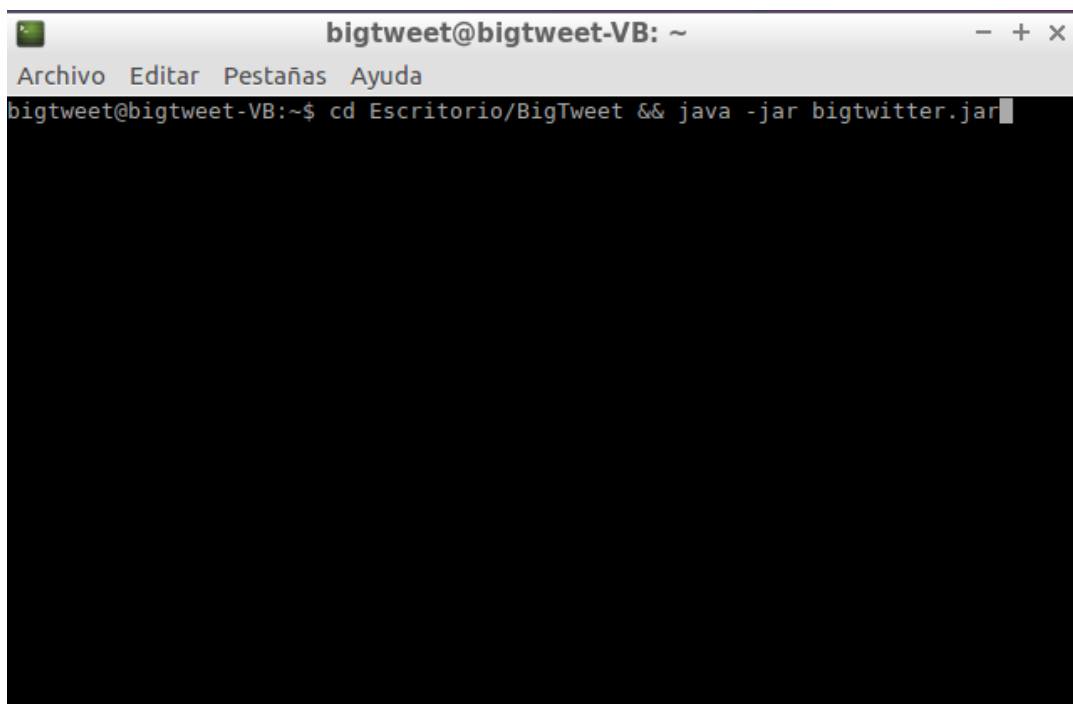


Ilustración 14: consola para lanzar la aplicación

Al presionar la tecla Enter visualizará el entorno BigTweet en su pantalla.

#### 4.1.2.1. Configuración de parámetros

Muchos de los parámetros utilizados para la ejecución y posterior análisis de la simulación no pueden ser modificados directamente desde la interfaz de usuario. Para ello es posible cambiarlos manualmente desde el fichero de configuración situado en la ruta:

“Escritorio/BigTweet/configuration/config.properties”

A continuación se muestra una descripción más en profundidad de los parámetros más relevantes. Para que una vez modificados se apliquen al modelo implementado correctamente deberá reinizarse la aplicación BigTweet.

Parámetro	Descripción
comparingdata	Indica la ruta de ficheros donde se encuentran los datos reales con los que se quiere comparar la simulación. Por defecto es el fichero palinUsers.txt
stopcondition	Indica la condición de parada de la simulación. Por defecto, se detiene si en 2 días no ha habido cambios en los estados.

El resto de parámetros del archivo hacen referencia a rutas de archivos y parámetros preestablecidos por defecto, los cuales son irrelevantes para la simulación o bien pueden ser modificados desde la propia interfaz.

En caso de querer modificar los parámetros internos de cada uno de los 'Spread Model', se deberá acceder al fichero de configuración de los mismos situado en la ruta:

“Escritorio/BigTweet/configuration/SpreadModelsParameters.json”

En este archivo se pueden cambiar los parámetros de cada uno de los modelos M1, M2 y M3. Estos parámetros son ***initiallyInfected***, que indica el número inicial de usuarios infectados, ***probInfect***, que refleja la probabilidad de que un usuario se infecte por otro, ***timeLag***, referencia al



tiempo de retardo, *probAcceptDeny*, probabilidad de curar o vacunar a un vecino y *beacons*, siendo el número de balizas. Al igual que en la configuración de los parámetros de la simulación, para que las modificaciones tengan efecto inmediato deberá reiniciarse la aplicación BigTweet.

#### 4.1.2.1. Descripción básica de la interfaz

Para acceder al panel de administración principal deberá pulsar en la pestaña 'Advanced options'. Desde esta sección se seleccionará el 'Spread model', con valores M1 M2 y M3. Una vez ejecutada la simulación dándole al play situado en la barra inferior, podrán exportarse los resultados con la extensión gefx. Podrá añadirse una nueva baliza en la pantalla del grafo, o destacar uno en concreto. Por último se puede comparar la simulación realizada con datos reales (fichero especificado en *config.properties*). En el panel derecho “Important nodes” se pueden filtrar los nodos por su centralidad (betweenness, closeness, degree y random). Al cambiar el tipo de centralidad se debe pulsar el botón get nodes para guardar los cambios. En caso de querer ocultar alguno de los nodos simplemente hay que desmarcar la casilla del mismo y pulsar el botón “beacon towards selection”. Para resaltar en el grafo aquellos nodos seleccionados con un círculo más grande simplemente hay que pulsar el botón “highlight selection”.

En la pestaña model se encuentran algunos parámetros iniciales de la simulación como el número de usuarios, número máximo de enlaces por nodo etc. Deberán cambiarse antes de ejecutar el modelo.

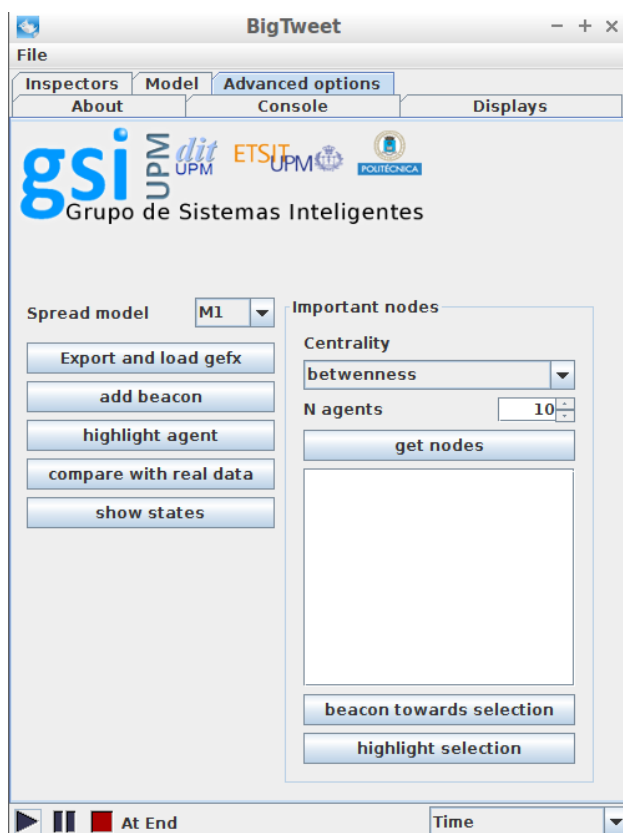
La pestaña “Displays” muestra los gráficos de Social Network, visualizándose el grafo de distribución de los nodos, y Users States, viendo reflejada gráficamente la evolución de los usuarios en cada paso de la simulación.

Por último desde la pestaña “Console” pueden modificarse las condiciones de ejecución de la simulación, sus tiempos de parada, retardo entre saltos y diferentes campos para detener la simulación automáticamente en caso de querer analizar en profundidad el comportamiento del escenario en diferentes estados.

## 4.2. Simulación del escenario de marketing viral

A continuación, se va a proceder a explicar detalladamente algunas de las pruebas realizadas para la evaluación del prototipo. Las simulaciones se harán utilizando un dataset creado específicamente para el escenario contemplado, aunque sería posible modificar este dataset para ajustarlo al escenario que se requiera.

El primer paso será la instalación del software BigTweet. Una vez instalado y ejecutado el simulador, se muestra la siguiente pantalla, en la que se puede navegar por las diferentes pestañas para modificar algunos parámetros de la configuración de la simulación.



*Ilustración 15: Pantalla principal del Prototipo*

#### 4.2.1. Pruebas con el modelo M1

El escenario considerado está definido de acuerdo con el modelo de simulación social para marketing viral. En este primer modelo, se ha establecido que los usuarios pueden publicar opiniones negativas sobre la campaña de marketing de una empresa determinada, y se procede a estudiar la propagación de estas opiniones entre el resto de usuarios de la red.

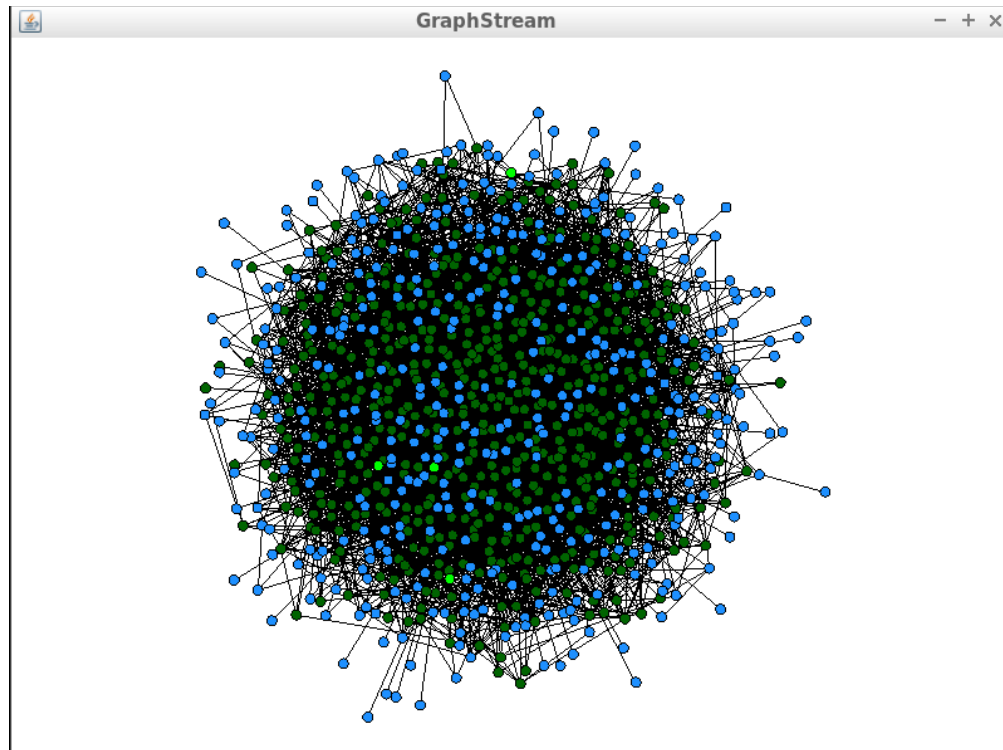
##### 4.2.1.1. Prueba inicial

En este escenario se han tomado 1.000 usuarios, con un número máximo de enlaces por usuario de 10 y un número de usuarios con opinión negativa inicial de 2. Además, la probabilidad de *contagio* de la opinión negativa de un usuario a otro es de 0.01 y la probabilidad de vacunación es de 0.065.

El primer parámetro que se debe ajustar, es la selección del modelo utilizado en la simulación. Para ello, en la pestaña *Advanced options*, se debe pulsar sobre el desplegable *spread model*, y seleccionar el modelo deseado, en este caso, M1. Los parámetros usados en esta primera prueba son los siguientes:

```
"users": 1000,  
"maxLinkPerNode": 10,  
"initiallyInfected":2,  
"probInfect": 0.01,  
"probAcceptDeny": 0.065,
```

Una vez configurados los parámetros deseados, se procederá a iniciar la simulación. Para ello, se pulsará el botón *play*, alojado en la esquina inferior izquierda. Al iniciarse la simulación, se abrirán varias ventanas. La mostrada a continuación, muestra un grafo de la red, en el que están representados los usuarios con distintos colores según su naturaleza. Para esta prueba, podemos comprobar como la red es muy extensa, de 1.000 usuarios, algunos de los cuales han quedado en estado neutro (azules) y otros en estado *vacunado*, que serían aquellos que no se creen la opinión negativa pues tienen una positiva.



*Ilustración 16: Grafo de la red con 1000 usuarios*

La ventana mostrada en la siguiente captura muestra la evolución del estado de los usuarios a lo largo del tiempo. En esta ventana se puede comprobar como cada vez hay más usuarios *vacunados*.

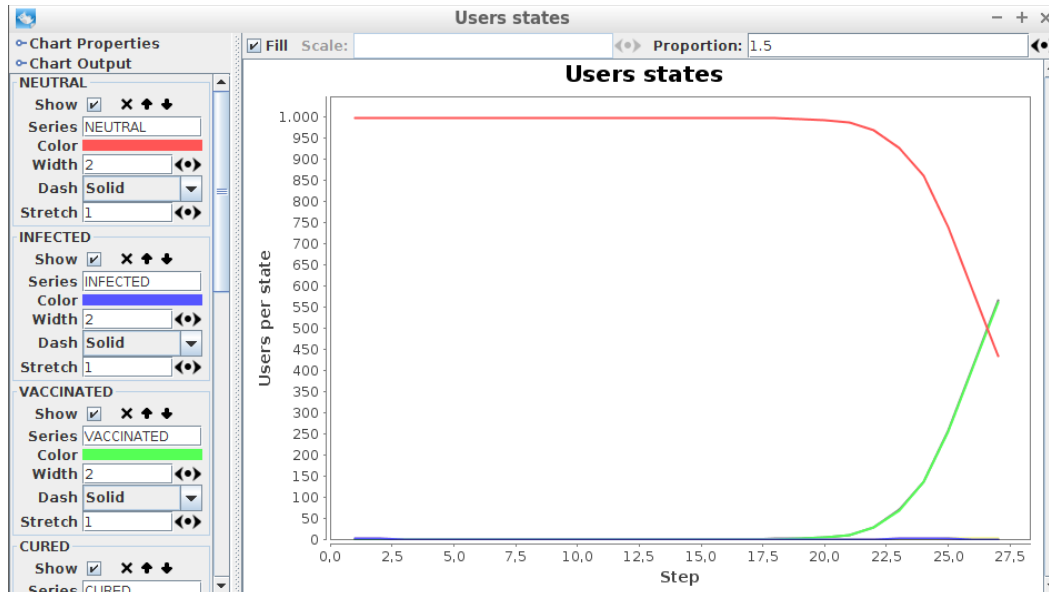


Ilustración 17: Estado de los usuarios

En esta prueba, al ser el número de usuarios tan elevado, cuesta ver de manera gráfica el número de usuarios infectados. Sin embargo, esto se puede observar desde la pestaña *Model*.

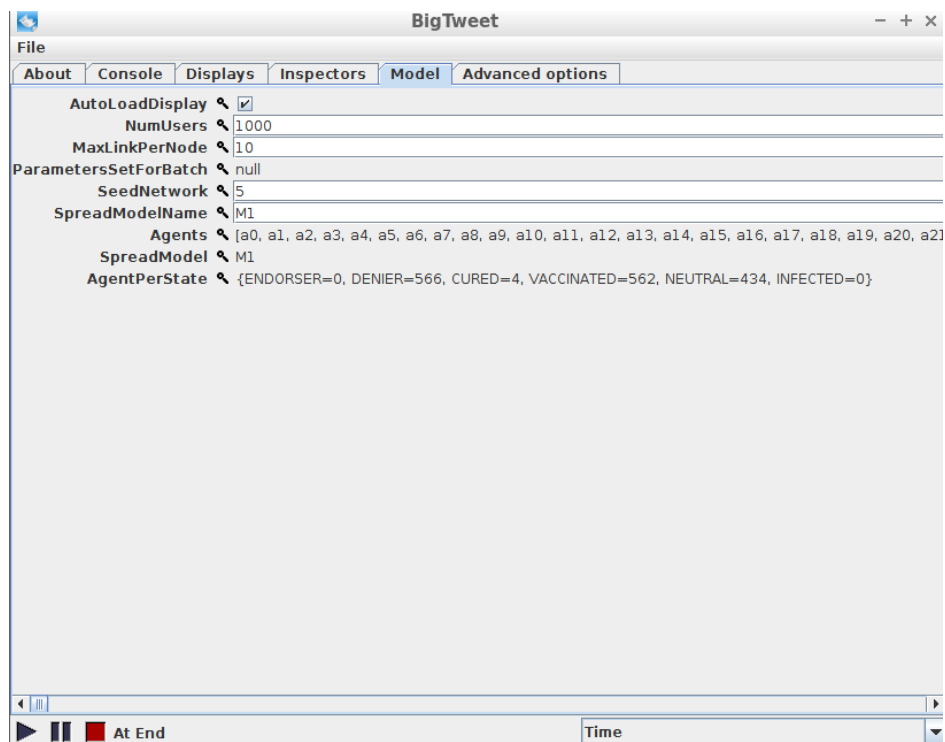
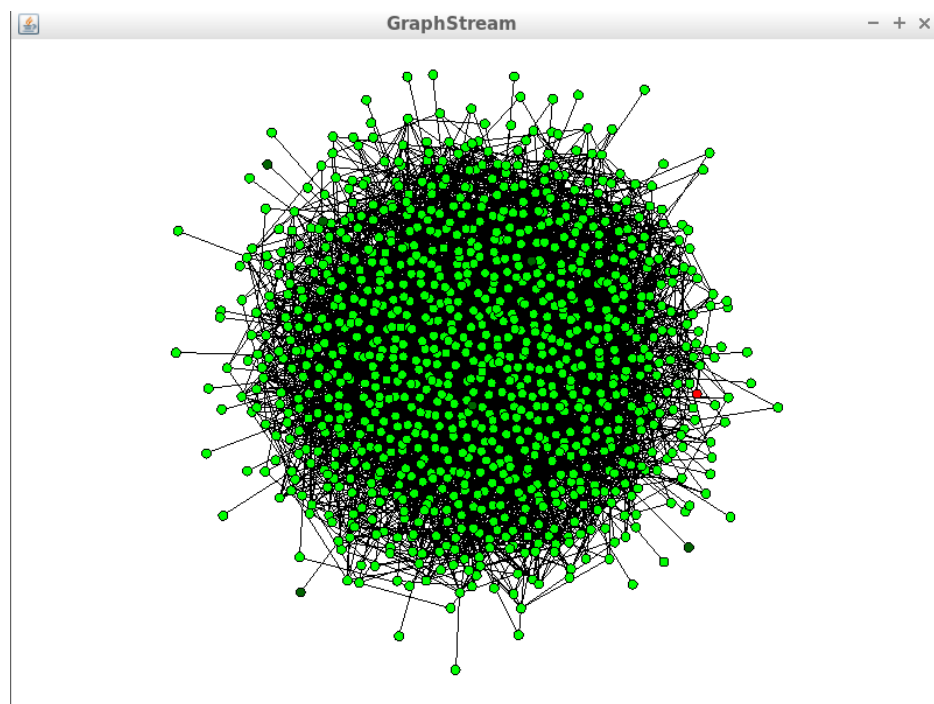


Ilustración 18: Parámetros avanzados del modelo

Como vemos, en esta prueba al final no queda ningún usuario infectado (que propague la opinión negativa), ya que los que ha habido han sido *curados* ante el rumor, y otros han sido *vacunados* ante el mismo.

#### 4.2.1.2. Aumento de la probabilidad de infección

En esta prueba se ha modificado la probabilidad de infección del valor inicial (que era 0.1) a un valor de 0.5. Comprobamos en el grafo de la red, que en este caso la mayoría de los usuarios acaban siendo curados en vez de vacunados como en el caso anterior. Esto es así ya que, como veremos a continuación, la probabilidad de infección tal alta hace que los usuarios se contagien de la opinión negativa de una forma exponencial.



*Ilustración 19: Grafo resultante*

Es el gráfico de la evolución temporal de los usuarios el que en este caso aporta más información, ya que como se puede observar, en muy poco tiempo casi la totalidad de los usuarios de la red han quedado infectados por la opinión negativa. Sin embargo, progresivamente con el paso del tiempo, han quedado curados de ella.

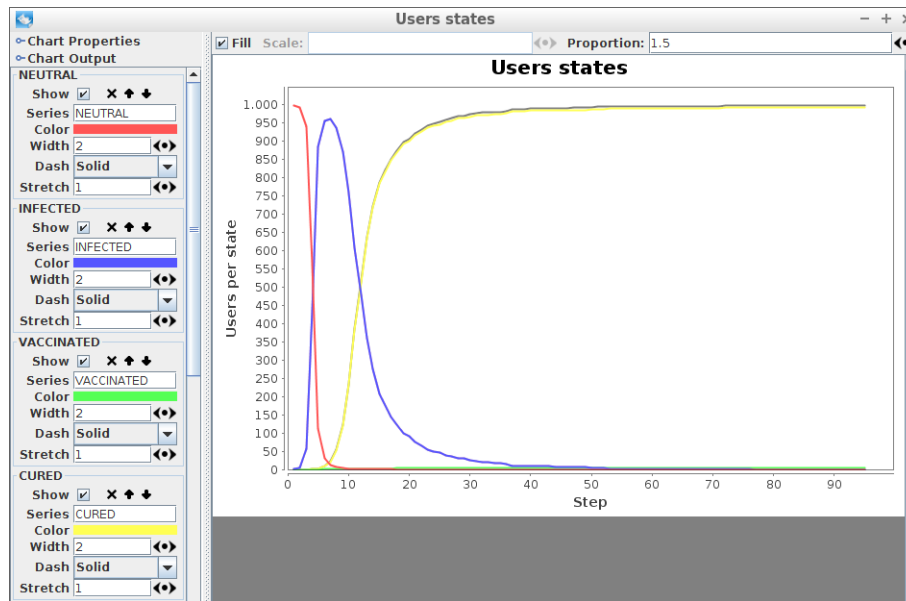


Ilustración 20: Estados de los usuarios

#### 4.2.2. Pruebas con el modelo M2

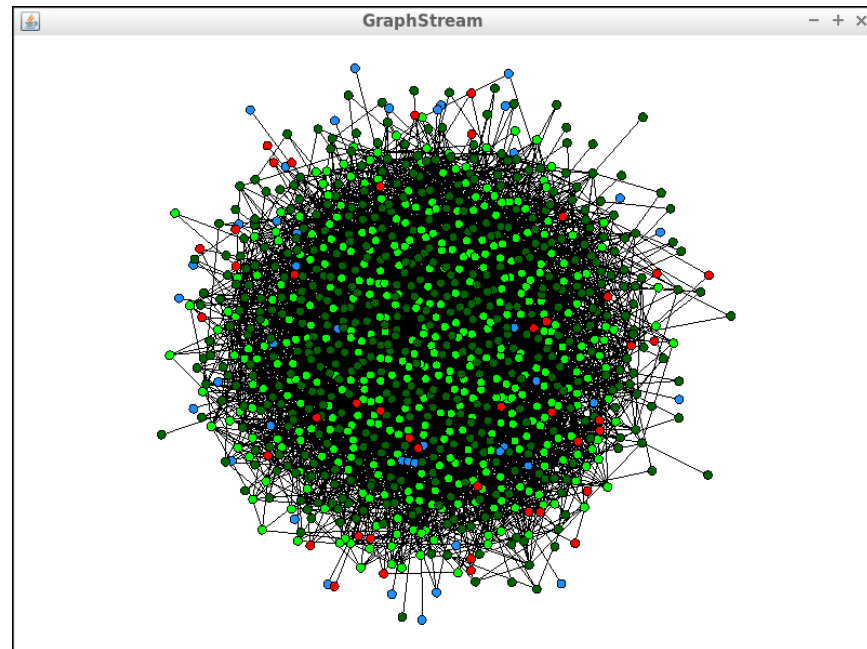
Como se ha explicado previamente, M2 es una modificación del modelo M1 para incluir la idea de que los usuarios vacunados difundan anti-rumores. Además, un nodo neutro que tenga un vecino infectado, puede convertirse en un usuario vacunado si este nodo sabía de cualquier información externa que la desinformación era falsa. Con esto en mente, M2 modifica M1 para incluir una probabilidad de hacer un *denier*, es decir, convertir un usuario neutro en un usuario vacunado al propagar un rumor. Para estas pruebas, debemos cambiar el modelo seleccionado desde la pestaña *Advanced Options*, para seleccionar M2.

##### 4.2.2.1. Prueba inicial

En este escenario se han tomado 1.000 usuarios, con un número máximo de enlaces por usuario de 10 y un número de usuarios con opinión negativa inicial de 2. Además, la probabilidad de *contagio* de la opinión negativa de un usuario a otro es de 0.02, la probabilidad de hacer un *denier* 0.01, y la probabilidad de vacunación 0.01

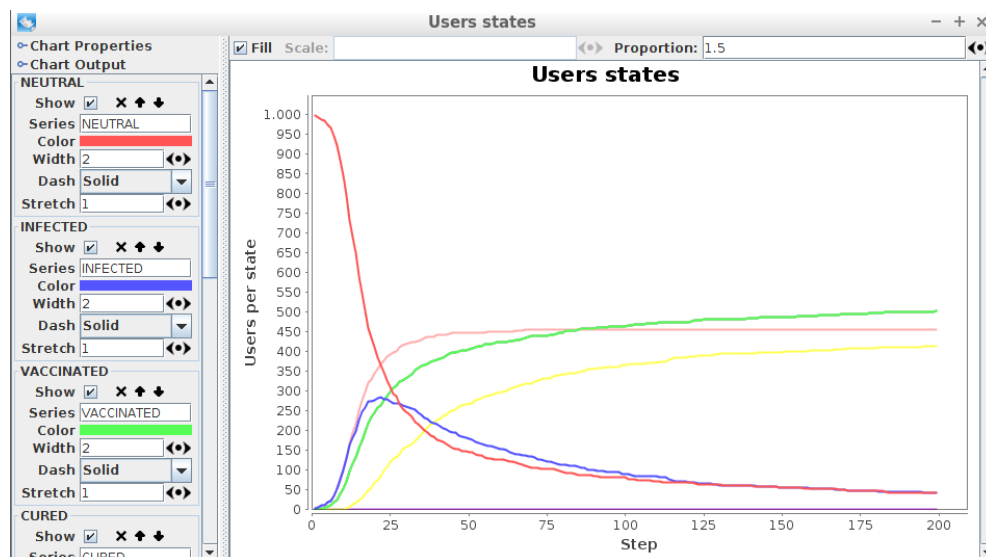
```
"users": 1000,
"maxLinkPerNode": 10,
"initiallyInfected": 2,
"probInfect": 0.02,
"probAcceptDeny": 0.01,
"probMakeDenier": 0.01,
```

Los resultados obtenidos son los siguientes. El grafo de la red quedaría como se muestra en la captura de pantalla a continuación.



*Ilustración 21: Grafo obtenido*

Como vemos, una mayoría de usuarios están curados o vacunados, dato que se confirma con el gráfico de evolución temporal que mostramos a continuación:

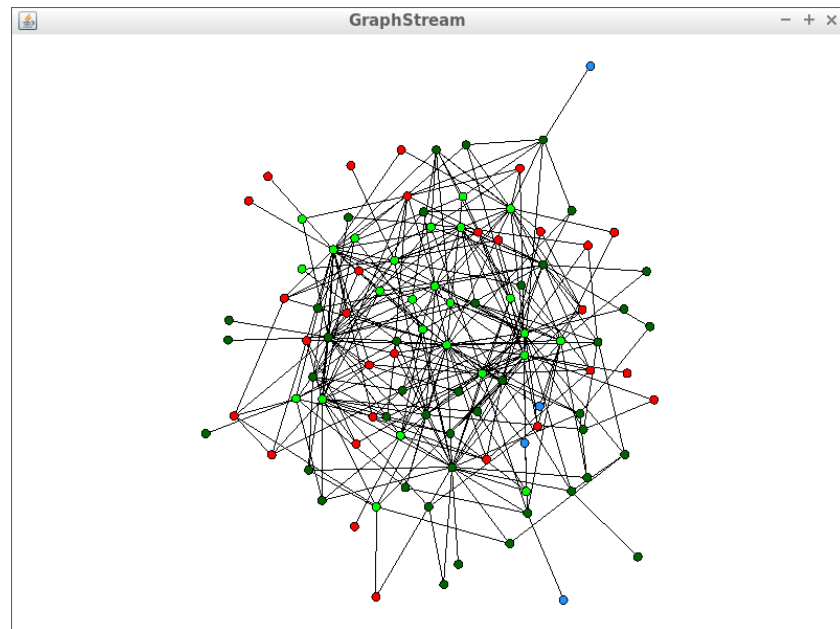


*Ilustración 22: Usuarios curados y vacunados*

Como se puede observar, inicialmente crece el número de infectados, que luego va bajando mientras el número de curados sube. Además, el número de usuarios vacunados también resulta bastante grande.

#### 4.2.2.2. Prueba disminuyendo el número de usuarios

Con el fin de observar los usuarios en el grafo de la red de una manera más clara, en esta prueba se ha reducido el número de usuarios a 100, y el número máximo de enlaces por nodo a 5.



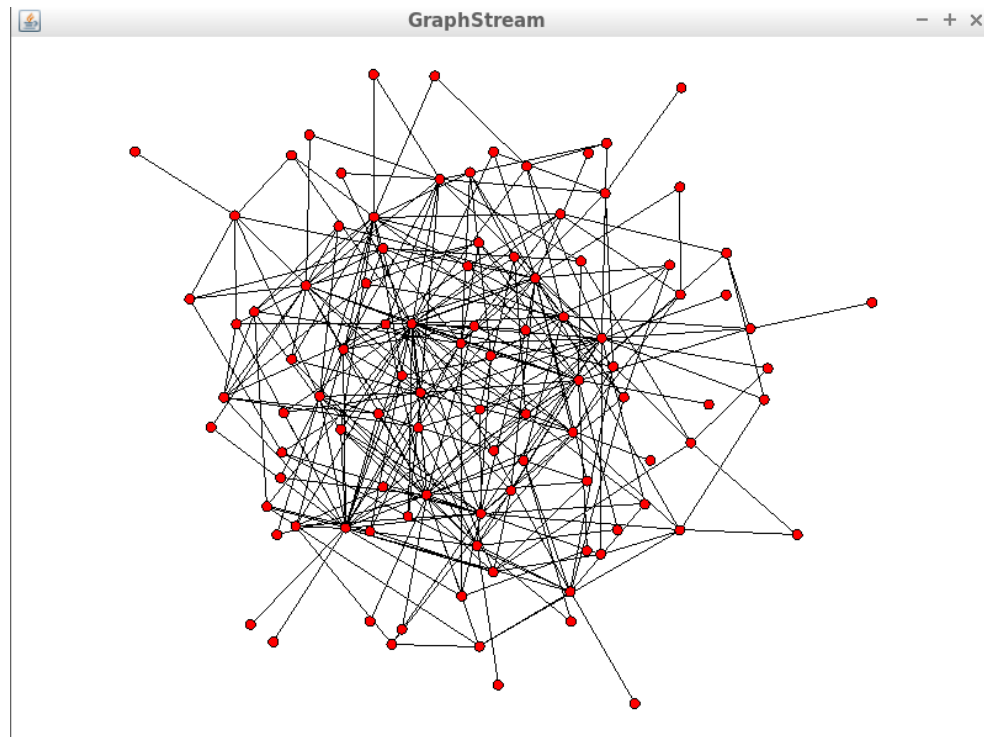
*Ilustración 23: Grafo menor número de usuarios*

Ahora el grafo de la red es mucho más sencillo y es posible distinguir con claridad el estado de cada usuario. Los rojos en este caso serían los usuarios infectados con la opinión negativa, los azules serían los usuarios neutros, los de color verde claro serían los curados y los de color verde oscuro los vacunados.

#### 4.2.2.3. Prueba aumentando la probabilidad de infección

En este caso, se ha modificado la probabilidad de infección de 0.02 a 0.5. Lo que se debe apreciar es un aumento de los usuarios infectados. Esto lo podemos confirmar con los resultados, mostrados en las siguientes capturas.



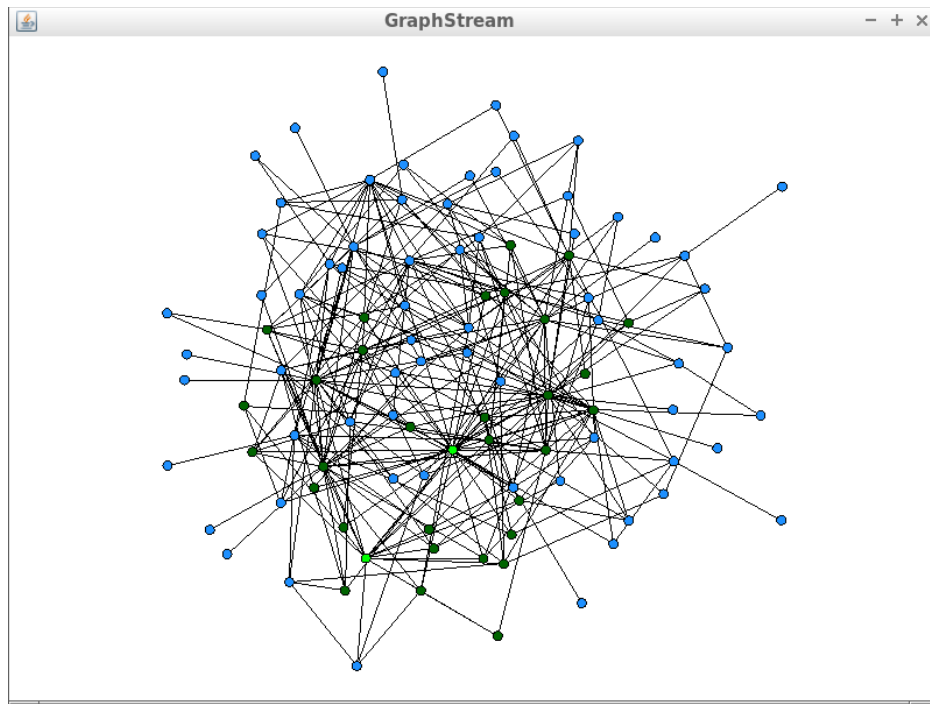


*Ilustración 24: Prueba aumentando probabilidad de infección*

Como podemos ver, para una probabilidad tan alta de infección, al final todos los usuarios terminan infectados con la probabilidad negativa.

#### 4.2.2.4. Prueba aumentando la probabilidad de hacer un denier

En este caso, se vuelve la probabilidad de infección a su valor inicial de 0.02, y se aumenta la probabilidad de hacer un denier a 0.3. Esto debería dar lugar a una red con mayoría de usuarios vacunados, y muy pocos usuarios infectados. Esto se puede confirmar observando los resultados.



*Ilustración 25: Prueba con mayor probabilidad de denier*

A continuación, se muestra también la evolución de los estados de los usuarios. Como se puede comprobar, los usuarios infectados iniciales terminan curándose, mientras que no se infectan nuevos usuarios al crecer el número de vacunados.



*Ilustración 26: Estados de los usuarios resultantes*

## 5. Resumen final y conclusión

Este documento revisa las principales consideraciones metodológicas para diseñar un modelo de difusión de rumores, un modelo de propagación de rumores de línea de base, un nuevo modelo de difusión de rumores, las principales consideraciones en la implementación de estos modelos y el prototipo BigTweet.

El documento describe el modelo realizado de simulación multiagente de marketing viral basado en modelos de contagio, y se ilustra mediante el presente prototipo cómo se emplean para validar campañas de marketing, y simular su comportamiento futuro.

## Referencias

[wik, a] Agent-based modeling software frameworks list. Accessed February 1, 2015. .

[wik, b] Social network analysis software. Accessed February 1, 2015. .

[twi, a] Twitter REST API console website.. Accessed February 1, 2015.

[twi, b] Twitter REST API documentation website.. Accessed February 1, 2015.

[twi, c] Twitter terms of use website.. Accessed February 1, 2015.

[Abraham et al., 2010] Abraham, A., Hassanien, A.-E., and Snasel, V. (2010). Computational social network analysis. *Computational Social Network Analysis, Computer Communications and Networks*, 1.

[Andrienko and Andrienko, 2005] Andrienko, N. and Andrienko, G. (2005). *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Beck et al., 2001] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001). *Manifesto for Agile Software Development*. Online version: <http://agilemanifesto.org/>; letzter Zugriff 08.07.2010).

[Bothos et al., 2010] Bothos, E., Apostolou, D., and Mentzas, G. (2010). Using social media to predict future events with agent-based markets. *IEEE Intelligent Systems*, 25(6):50–58.

[Buchanan, 2009] Buchanan, M. (2009). Economics: Meltdown modelling. *Nature*, 460(7256):680.

[Campuzano et al., 2015] Campuzano, F., Garcia-Valverde, T., Botia, J. A., and Serrano, E. (2015). Generation of human computational models with machine learning. *Information Sciences*, 293(0):97 – 114.

[Campuzano et al., 2014] Campuzano, F., Garcia-Valverde, T., Serrano, E., and Botía, J. A. (2014). Generation of human computational models with knowledge engineering. *Engineering Applications of Artificial Intelligence*, 35(0):259 – 276.

[Cha et al., 2010] Cha, M., Haddadi, H., Benevenuto, F., and Gummadi, K. (2010). Measuring user influence in twitter: The million follower fallacy. In *4th International AAAI Conference on Weblogs and Social Media (ICWSM)*.

[Challenger et al., 2014] Challenger, M., Demirkol, S., Getir, S., Mernik, M., Kardas, G., and Kosar, T. (2014). On the use of a domain-specific modeling language in the development of multiagent systems. *Engineering Applications of Artificial Intelligence*, 28(0):111 – 141.

[Cohen, ] Cohen, H. 72 marketing definitions. . Accessed February 1, 2015.

[De Domenico et al., 2013] De Domenico, M., Lima, A., Mougél, P., and Musolesi, M. (2013). The Anatomy of a Scientific Rumor. *Scientific Reports*, 3.

[Farmer and Foley, 2009] Farmer, J. D. and Foley, D. (2009). The economy needs agent-based modelling. *Nature*, 460(7256):685–686.

[Flentge et al., ] Flentge, F., Polani, D., and Uthmann, T. Modelling the emergence of possession norms using memes. *J. Artificial Societies and Social Simulation*, (4).

[Frias-Martinez and Frias-Martinez, 2014] Frias-Martinez, V. and Frias-Martinez, E. (2014). Spectral clustering for sensing urban land use using twitter activity. *Engineering Applications of Artificial Intelligence*, 35(0):237 – 245.

[Garcia-Valverde et al., 2012] Garcia-Valverde, T., Campuzano, F., Serrano, E., Villa, A., and Botia, J. A. (2012). Simulation of human behaviours for the validation of ambient intelligence services: A methodological approach. *Journal of Ambient Intelligence and Smart Environments*, 4(3):163–181.

[Gatti et al., 2013] Gatti, M. A. d. C., Appel, A. P., dos Santos, C. N., Pinhanez, C. S., Cavalin, P. R., and Neto, S. B. (2013). A simulation-based approach to analyze the information diffusion in microblogging online social network. In *Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World*, WSC '13, pages 1685–1696, Piscataway, NJ, USA. IEEE Press.

- [Gilbert and Troitzsch, 2005] Gilbert, N. and Troitzsch, K. G. (2005). *Simulation for the Social Scientist*. Open University Press.
- [Gupta et al., 2013a] Gupta, A., Lamba, H., and Kumaraguru, P. (2013a). \$1.00 per RT #BostonMarathon #PrayForBoston: Analyzing fake content on twitter. San Francisco, CA.
- [Gupta et al., 2013b] Gupta, A., Lamba, H., Kumaraguru, P., and Joshi, A. (2013b). Faking sandy: Characterizing and identifying fake images on twitter during hurricane sandy. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 729–736, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.
- [Gutiérrez et al., 2013] Gutiérrez, C., García-Magariño, I., Serrano, E., and Botía, J. A. (2013). Robust design of multi-agent system interactions: A testing approach based on pattern matching. *Engineering Applications of Artificial Intelligence*, 26(9):2093 – 2104.
- [Hethcote, 2000] Hethcote, H. W. (2000). The mathematics of infectious diseases. *SIAM Review*, 42:599–653.
- [Holubiec et al., 2012] Holubiec, J., Szkatula, G., and Wagner, D. (2012). A knowledge-based model of parliamentary election. *Information Sciences*, 202:24–40.
- [Jin et al., 2013] Jin, F., Dougherty, E., Saraf, P., Cao, Y., and Ramakrishnan, N. (2013). Epidemiological modeling of news and rumors on twitter. In *Proceedings of the 7th Workshop on Social Network Mining and Analysis*, SNAKDD '13, pages 8:1–8:9, New York, NY, USA. ACM.
- [Khamis and Nagi, 2013] Khamis, M. A. and Nagi, K. (2013). Designing multi-agent unit tests using systematic test design patterns-(extended version). *Engineering Applications of Artificial Intelligence*, 26(9):2128 – 2142.
- [Kilikki et al., 2014] Kilikki, O., Kangasrääsio, A., Nikkilä, R., Alahäivälä, A., and Seilonen, I. (2014). Agent-based modeling and simulation of a smart grid: A case study of communication effects on frequency control. *Engineering Applications of Artificial Intelligence*, 33(0):91 – 98.
- [Kitchenham et al., 2009] Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., and Linkman, S. (2009). Systematic literature reviews in software engineering - a systematic literature review. *Inf. Softw. Technol.*, 51(1):7–15.

[Kostka et al., 2008] Kostka, J., Oswald, Y. A., and Wattenhofer, R. (2008). Word of Mouth: Rumor Dissemination in Social Networks Structural Information and Communication Complexity. In Shvartsman, A. A. and Felber, P., editors, *Structural Information and Communication Complexity*, volume 5058 of *Lecture Notes in Computer Science*, chapter 16, pages 185–196. Springer Berlin / Heidelberg, Berlin, Heidelberg.

[Kwon et al., 2013a] Kwon, S., Cha, M., Jung, K., Chen, W., and Wang, Y. (2013a). Aspects of rumor spreading on a microblog network. In Jatowt, A., Lim, E.-P., Ding, Y., Miura, A., Tezuka, T., Dias, G., Tanaka, K., Flanagin, A., and Dai, B., editors, *Social Informatics*, volume 8238 of *Lecture Notes in Computer Science*, pages 299–308. Springer International Publishing.

[Kwon et al., 2013b] Kwon, S., Cha, M., Jung, K., Chen, W., and Wang, Y. (2013b). Prominent features of rumor propagation in online social media. In Xiong, H., Karypis, G., Thuraisingham, B. M., Cook, D. J., and Wu, X., editors, *2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013*, pages 1103–1108. IEEE Computer Society.

[Landwehr and Carley, 2014] Landwehr, P. and Carley, K. (2014). Social media in disaster relief. In Chu, W. W., editor, *Data Mining and Knowledge Discovery for Big Data*, volume 1 of *Studies in Big Data*, pages 225–257. Springer Berlin Heidelberg.

[Li et al., 2008] Li, X., Mao, W., Zeng, D., and Wang, F.-Y. (2008). Agent-based social simulation and modeling in social computing. In Yang, C., Chen, H., Chau, M., Chang, K., Lang, S.-D., Chen, P., Hsieh, R., Zeng, D., Wang, F.-Y., Carley, K., Mao, W., and Zhan, J., editors, *Intelligence and Security Informatics*, volume 5075 of *Lecture Notes in Computer Science*, pages 401–412. Springer Berlin Heidelberg.

[Liu and Chen, 2011] Liu, D. and Chen, X. (2011). Rumor propagation in online social networks like twitter – a simulation study. In *Proceedings of the 2011 Third International Conference on Multimedia Information Networking and Security*, MINES '11, pages 278–282, Washington, DC, USA. IEEE Computer Society.

[Long and Wong, 2014] Long, C. and Wong, R. C.-W. (2014). Viral marketing for dedicated customers. *Information Systems*, 46(0):1 – 23.

[Luke et al., 2004] Luke, S., Cioffi-Revilla, C., Panait, L., and Sullivan, K. (2004). Mason: A new multi-agent simulation toolkit. In *Proceedings of the 2004 Swarmfest Workshop*.

- [Mavridis and Symeonidis, 2014] Mavridis, T. and Symeonidis, A. L. (2014). Semantic analysis of web documents for the generation of optimal content. *Engineering Applications of Artificial Intelligence*, 35(0):114 – 130.
- [Mendoza et al., 2010] Mendoza, M., Poblete, B., and Castillo, C. (2010). Twitter under crisis: Can we trust what we rt? In *Proceedings of the First Workshop on Social Media Analytics, SOMA '10*, pages 71–79, New York, NY, USA. ACM.
- [Nekovee et al., 2007] Nekovee, M., Moreno, Y., Bianconi, G., and Marsili, M. (2007). Theory of rumour spreading in complex social networks. *Physica A: Statistical Mechanics and its Applications*, 374(1):457 – 470.
- [North et al., 2006] North, M. J., Collier, N. T., and Vos, J. R. (2006). Experiences creating three implementations of the repast agent modeling toolkit. *ACM Trans. Model. Comput. Simul.*, 16(1):1–25.
- [Nwana, 1996] Nwana, H. S. (1996). Software agents: An overview. *Knowledge Engineering Review*, 11:205–244.
- [Placzek, 2014] Placzek, B. (2014). A self-organizing system for urban traffic control based on predictive interval microscopic model. *Engineering Applications of Artificial Intelligence*, 34(0):75 – 84.
- [Qazvinian et al., 2011] Qazvinian, V., Rosengren, E., Radev, D. R., and Mei, Q. (2011). Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11*, pages 1589–1599, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Rand and Rust, 2011] Rand, W. and Rust, R. T. (2011). Agent-based modeling in marketing: Guidelines for rigor. *International Journal of Research in Marketing*, 28(3):181–193.
- [Rolla and Curado, 2013] Rolla, V. G. and Curado, M. (2013). A reinforcement learning-based routing for delay tolerant networks. *Engineering Applications of Artificial Intelligence*, 26(10):2243 – 2250.
- [Rossant, 2013] Rossant, C. (2013). *Learning IPython for Interactive Computing and Data Visualization*. Packt Publishing.



[Russell, 2011a] Russell, M. A. (2011a). *21 Recipes for Mining Twitter*. O'Reilly Media, 1 edition.

[Russell, 2011b] Russell, M. A. (2011b). *Mining the Social Web*. O'Reilly Media, 1 edition.

[Seo et al., 2012] Seo, E., Mohapatra, P., and Abdelzaher, T. (2012). Identifying rumors and their sources in social networks.

[Serrano and Botia, 2013] Serrano, E. and Botia, J. (2013). Validating ambient intelligence based ubiquitous computing systems by means of artificial societies. *Information Sciences*, 222(0):3 – 24.

[Serrano et al., a] Serrano, E., Iglesias, C. A., and Garijo, M. BigTweet website, simulator for rumor spreading models and rumor control strategies in Twitter with support for Big Data technologies. . Accessed February 1, 2015.

[Serrano et al., b] Serrano, E., Iglesias, C. A., and Garijo, M. Paper additional material: datasets, simulations code, and extended experiments. . Accessed February 1, 2015.

[Serrano et al., 2014a] Serrano, E., Moncada, P., Garijo, M., and Iglesias, C. A. (2014a). Evaluating social choice techniques into intelligent environments by agent based social simulation. *Information Sciences*, 286(0):102–124.

[Serrano et al., 2014b] Serrano, E., Poveda, G., and Garijo, M. (2014b). Towards a holistic framework for the evaluation of emergency plans in indoor environments. *Sensors*, 14(3):4513–4535.

[Serrano et al., 2013] Serrano, E., Rovatsos, M., and Botía, J. A. (2013). Data mining agent conversations: A qualitative approach to multiagent systems analysis. *Information Sciences*, 230(0):132 – 146.

[Shah and Zaman, 2011] Shah, D. and Zaman, T. (2011). Rumors in a network: Who's the culprit? *IEEE Transactions on Information Theory*, 57(8):5163–5181.

[Shamshirband et al., 2013] Shamshirband, S., Anuar, N. B., Kiah, M. L. M., and Patel, A. (2013). An appraisal and design of a multi-agent system based cooperative wireless intrusion detection computational intelligence technique. *Engineering Applications of Artificial Intelligence*, 26(9):2105 – 2127.

- [Starbird et al., 2014] Starbird, K., Maddock, J., Orand, M., Achterman, P., and Mason, R. M. (2014). Rumors, false flags, and digital vigilantes: Misinformation on twitter after the 2013 boston marathon bombing. In *iConference 2014 Proceedings* (p. 654 - 662).
- [Statell, ] Statell, G. Big data and simulations are transforming marketing. . Accessed February 1, 2015.
- [Tisue and Wilensky, 2004] Tisue, S. and Wilensky, U. (2004). NetLogo: A Simple Environment for Modeling Complexity.
- [Tripathy et al., 2010] Tripathy, R. M., Bagchi, A., and Mehta, S. (2010). A study of rumor control strategies on social networks. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10*, pages 1817–1820, New York, NY, USA. ACM.
- [Valecha et al., 2013] Valecha, R., Oh, O., and Rao, H. R. (2013). An exploration of collaboration over time in collective crisis response during the haiti 2010 earthquake. In Baskerville, R. and Chau, M., editors, *Proceedings of the International Conference on Information Systems, ICIS 2013, Milano, Italy, December 15-18, 2013*. Association for Information Systems.
- [Weng et al., 2013] Weng, L., Menczer, F., and Ahn, Y.-Y. (2013). Virality prediction and community structure in social networks. *Scientific Reports*, 3.
- [Woolridge, 2001] Woolridge, M. (2001). *Introduction to Multiagent Systems*. John Wiley & Sons, Inc., New York, NY, USA.
- [Yang et al., ] Yang, S. Y., Liu, A., and Mo, S. Y. K. Twitter financial community modeling using agent based simulation. SSRN scholarly paper, Rochester, NY. IEEE Computational Intelligence in Financial Engineering and Economics, London, 2013.
- [Zhao et al., 2013] Zhao, L., Cui, H., Qiu, X., Wang, X., and Wang, J. (2013). {SIR} rumor spreading model in the new media age. *Physica A: Statistical Mechanics and its Applications*, 392(4):995 – 1003.