



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

P5.3 Modelo de Simulación Social para Recomendación en Social Media basado en Big Data

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

Tabla de Contenidos

Sumario

1. Introducción.....	6
2. Modelos y Prototipo de simulación social para recomendación en social media.....	8
2.1 Modelos de Simulación Social de Recomendación.....	8
2.1.1 Baseline en el modelo de control.....	8
2.1.2. Nuevo modelo de control.....	8
2.2. Prototipo de Simulación.....	9
2.2.1 Extensión BigTweet.....	9
2.2.2. BigTweet javadoc.....	11
3. Pruebas.....	13
3.1. Instalación.....	13
3.1.1. Instalación de la máquina virtual.....	13
3.1.2. Ejecución del entorno BigTweet.....	16
3.1.2.1. Configuración de parámetros.....	17
3.1.2.1. Descripción básica de la interfaz.....	18
3.2. Simulación del escenario de recomendación en social media.....	18
3.2.1. Prueba inicial.....	19
3.3.2. Prueba aumentando el número de usuarios inicialmente infectados.....	21
4. Resumen final y conclusión.....	23

Índice de ilustraciones

Ilustración 1: Arquitectura global de BigMarket.....	9
Ilustración 2: Prototipo BigTweet extendido para incluir estrategias de control.....	10
Ilustración 3: Diagrama de clases UML de BigTweet.....	11
Ilustración 4: Página de descargas de VirtualBox.....	13
Ilustración 5: Ejecutables de VirtualBox.....	13
Ilustración 6: Asistente de VirtualBox.....	14
Ilustración 7: Asistente de VirtualBox para realizar la importación de OVF.....	14
Ilustración 8: Configuración de la máquina virtual.....	15
Ilustración 9: Proceso de importación de la máquina virtual.....	15
Ilustración 10: Resultado esperado tras importar la máquina virtual.....	16
Ilustración 11: consola para lanzar la aplicación.....	16
Ilustración 12: Pantalla principal del Prototipo.....	19
Ilustración 13: Resultado de la prueba.....	20
Ilustración 14: Evolución de la prueba.....	21
Ilustración 15: Grafo de la prueba.....	21
Ilustración 16: Evolución de los estados de los usuarios en la prueba.....	22

Resumen ejecutivo

Este entregable aplica la metodología de E5.1 y el modelo de rumores de P5.2, extendiéndolo para incluir un modelo de recomendación. El modelo de marketing resultante de la metodología tiene como objetivo recomendar acciones de marketing en redes sociales y más concretamente para el control de rumores maliciosos en Twitter. El motivo de basarse en Twitter es su naturaleza concisa que facilita el análisis de sentimientos, pero sobre todo que es una red eminentemente pública donde los datos pueden ser recogidos sin necesidad de consentimiento previo del usuario. El modelo resultante de P5.2 es la base para la evaluación y sugerencia de estrategias para el control de rumores. En concreto, las estrategias se centrarán en la consideración de usuarios especiales llamados “beacons” que representan o bien a la compañía que quiere controlar el rumor o bien a socios de esta. En concreto, representan a influenciadores de la red que pueden recomendar contenidos de marketing para mejorar las campañas virales. Específicamente, se discutirá: el número de beacons necesario; la posición recomendable de estos beacons en la red (que se puede conseguir siguiendo y siendo seguido por ciertos usuarios mediante incentivos); y las acciones realizables por estos beacons.

Información del Documento

Proyecto FIT Número	TSI-100102-2013-80	Acrónimo	Big Market
Título completo	Plataforma Big Data para simular y evaluar técnicas de marketing en entornos realistas		
URL	http://www.paradigmatecnologico.com/portfolio/bigmarket/		
URL del documento			

Entregable	Número	P5.3	Título	Prototipo de Modelo de Simulación Social para Recomendación en Social Media basado en Big Data
Paquete de Trabajo	Número	5	Título	Simulación social basada en Big Data
Tarea	Número	T5.3	Título	Modelo de Simulación Social para Recomendación en Social Media basado en Big Data

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 de Modelo de Simulación Social para Recomendación en Social Media basado en Big Data sobre el modelo de marketing viral propuesto en P5.2			
Palabras Clave	Justificación			

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

1. Introducción

En relación con las posibles estrategias de marketing en Twitter, la mayoría de ellas dependen de tener uno o varios usuarios de Twitter que representen a su marca (es decir, su producto / servicio / empresa), sus colaboradores o personas después de recibir algún incentivo. Estos usuarios estratégicos o semilla tienen que iniciar el proceso de difusión de la conciencia mediante la propagación de la información a sus amigos a través de sus relaciones sociales [Long y Wong, 2014]. En consecuencia, el objetivo clásico es maximizar (caso de publicidad) o minimizar (caso de rumores maliciosos) la difusión de información mientras se crea o selecciona el mínimo de estos usuarios estratégicos. Algunos ejemplos en la literatura de propagación del rumor son los agentes de beacons propuestos por Tripathy et al. [Tripathy et al., 2010] o los monitores introducidos por Seo et al. [Seo et al., 2012].

El problema explicado suele conducir a la cuestión de cuáles son los usuarios más importantes de la red. Esto se puede estudiar con las métricas específicas de Twitter, tales como el número de menciones y retweets como se propone por Cha et al. [Cha et al., 2010]; o indicadores de centralidad [Abraham et al., 2010], indicadores que identifican los vértices más importantes dentro de un gráfico como el outdegree y el indegree (que serían seguidores o amigos en Twitter). En ambos casos, las mediciones precisas de "importancia" pueden no ser recuperables y algunas aproximaciones podrían ser usadas. Por ejemplo, la API de Twitter no permite recuperar el número de retweets o menciones para un usuario. Además, las métricas de centralidad como la cercanía requieren toda la red (y estática). En los ejemplos dados anteriormente de agentes estratégicos, Tripathy et al. [Tripathy et al., 2010] sólo consideran posiciones aleatorias para los beacons mientras que Seo et al. [Seo et al., 2012] estudian los efectos de tener monitores en posiciones con diferentes indicadores de centralidad como la intermediación.

Como se explicó anteriormente, una estrategia de marketing se ocupa de la creación de agentes especiales en este contexto; y, por lo tanto, las pautas para modelar a los agentes generales de Twitter con agentes de simulación social basada en agentes (ABSS) dado en P5.2 también son válidos para el modelado de estas estrategias. Este trabajo contempla estrategias de marketing como una tarea extra en el método; es decir, los escenarios hipotéticos que ABSS permite comprender, evaluar y predecir. La razón es que un único modelo del mercado debe ser válido para evaluar una serie de estrategias sobre él (y viceversa). Sin embargo, otro error común en la investigación de ABSS es asumir que primero la realidad es modelada y luego los expertos pueden

decidir cualquier tipo de hipótesis que se evalúe sobre el modelo. Como se indica en la decisión del modelo de alcance de la modelización P5.2 sólo aspectos relevantes del mercado es la clave en ABSS y, por lo tanto, hay un fuerte acoplamiento entre las estrategias de marketing y el mercado simulado.

2. Modelos y Prototipo de simulación social para recomendación en social media

2.1 Modelos de Simulación Social de Recomendación

Los modelos realizados en este entregable extienden los modelos realizados en el entregable P5.2.

2.1.1 Baseline en el modelo de control

El enfoque realizado para el experimento baseline considera una estrategia de control basada en incluir un número de usuarios especiales llamados *beacons* que representan una autoridad que detecta la propagación de información errónea y la combate. En las estrategias de marketing se corresponde con un influenciador que recomienda contenidos de la marca a sus seguidores. Por lo tanto, los estados de los agentes se amplían con: *beacon-off* (beacon antes de detectar un rumor en un vecino) y *beacon-on* (beacon después de detectar un rumor). Cuando un nodo infectado tiene un beacon inactivo como vecino, este último se activa en el siguiente paso de tiempo. Los beacon activos, como usuarios curados y vacunados, inician una propagación anti-rumor a sus vecinos, intentando curarlos o vacunarlos con una probabilidad (*probAcceptDeny*) cada vez que el paso t . Dado que los anti-rumores empiezan desde los beacons, el parámetro de retardo de tiempo explicado no se utiliza.

Una modificación en el diseño original de la estrategia de control es que en lugar de seleccionar nodos de beacon al azar, se incluye un parámetro de centralidad para intentar hacer que la baliza siga los nodos más importantes de acuerdo con: los indicadores de centralidad, intermediación o proximidad [Abraham et al ., 2010]. Como resultado, los parámetros para este modelo de control explicado son: número de balizas y función de centralidad para incluirlas en el entorno.

2.1.2. Nuevo modelo de control

Como se ha explicado, M2 se puede combinar con la estrategia de control detallada en la sección 2. Sin embargo, esta sección propone un nuevo modelo de control basado en la estrategia de control original. Más específicamente, ya que no hay costo ni restricción en seguir a los usuarios de Twitter, este modelo propone que el beacon no sólo tiene que difundir anti-rumores a sus vecinos, sino también seguir los contactos de estos vecinos. De esta manera, incluso si los vecinos del beacon no difunden el anti rumor, la baliza puede hacerlo por sí misma en el siguiente paso del tiempo. Esta estrategia pretende minimizar el efecto observado en agentes curados: no pueden propagar anti-rumores porque implicaría admitir un error en tweets anteriores.

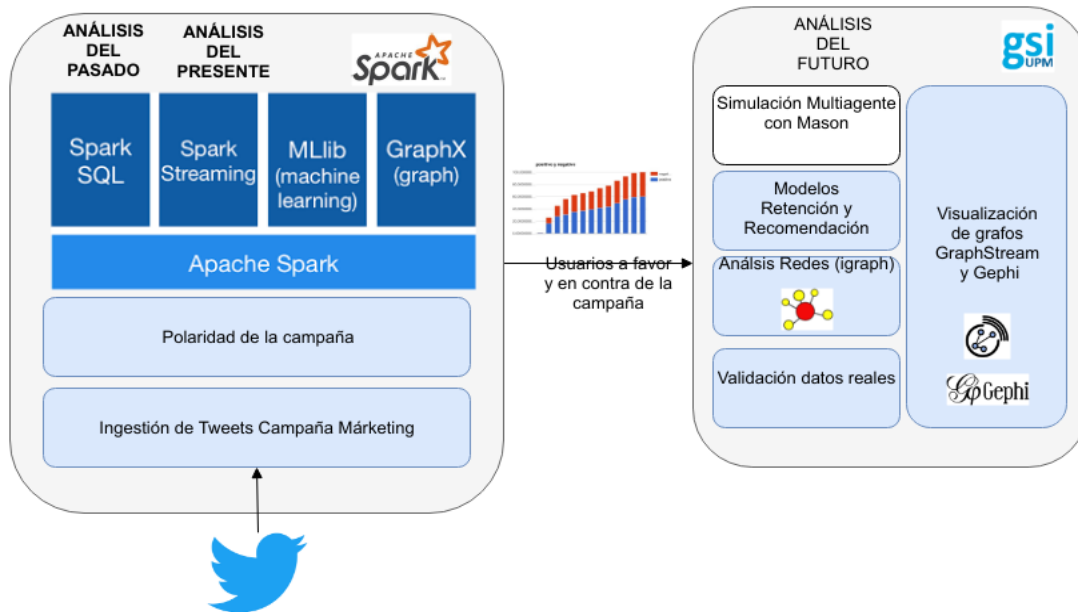


Ilustración 1: Arquitectura global de BigMarket

2.2. Prototipo de Simulación

Los modelos se integran en la arquitectura global de BigMarket como se muestra en la siguiente figura.

2.2.1 Extensión BigTweet

El prototipo BigTweet presentado en P5.2 se ha ampliado con estas estrategias. Para ello, se han modificado varias clases: UserAgentM1, UserAgentM2, UserAgentM3, SpreadModelM1, SpreadModelM2 y SpreadModelM3. Además, la GUI se ha ampliado para permitir a los usuarios ejecutar estrategias de control (modelo de control, número de balizas a incluir, posición de los mismos en la red, etcétera). Vea las opciones avanzadas en la figura 1.

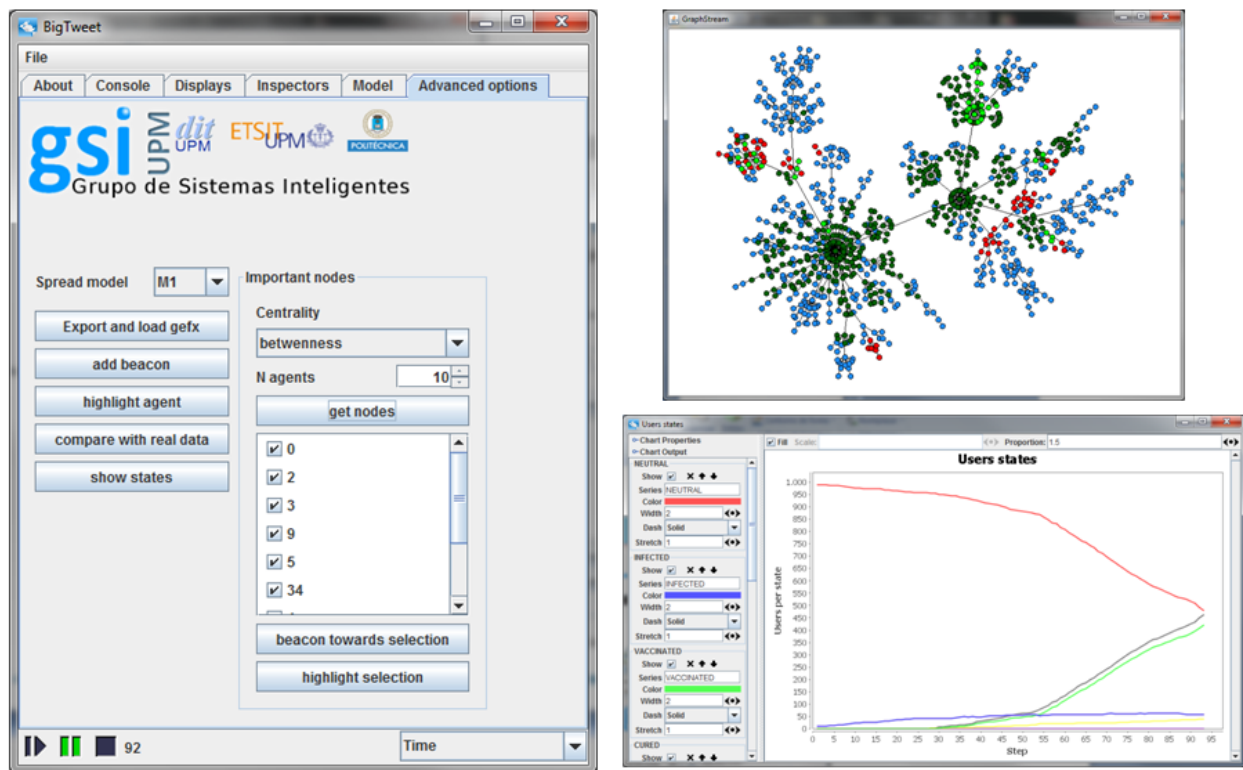


Ilustración 2: Prototipo BigTweet extendido para incluir estrategias de control

Por último, una serie de redes de análisis de redes sociales (SNA) se han conectado a BigTweet [wik, b]: GraphStream para ofrecer una pantalla de red dinámica, Gephi para seguir estudiando la red social de una potente interfaz gráfica de usuario, e iGraph para calcular las funciones de centralidad. Vea la clase GraphTools en la figura 2.

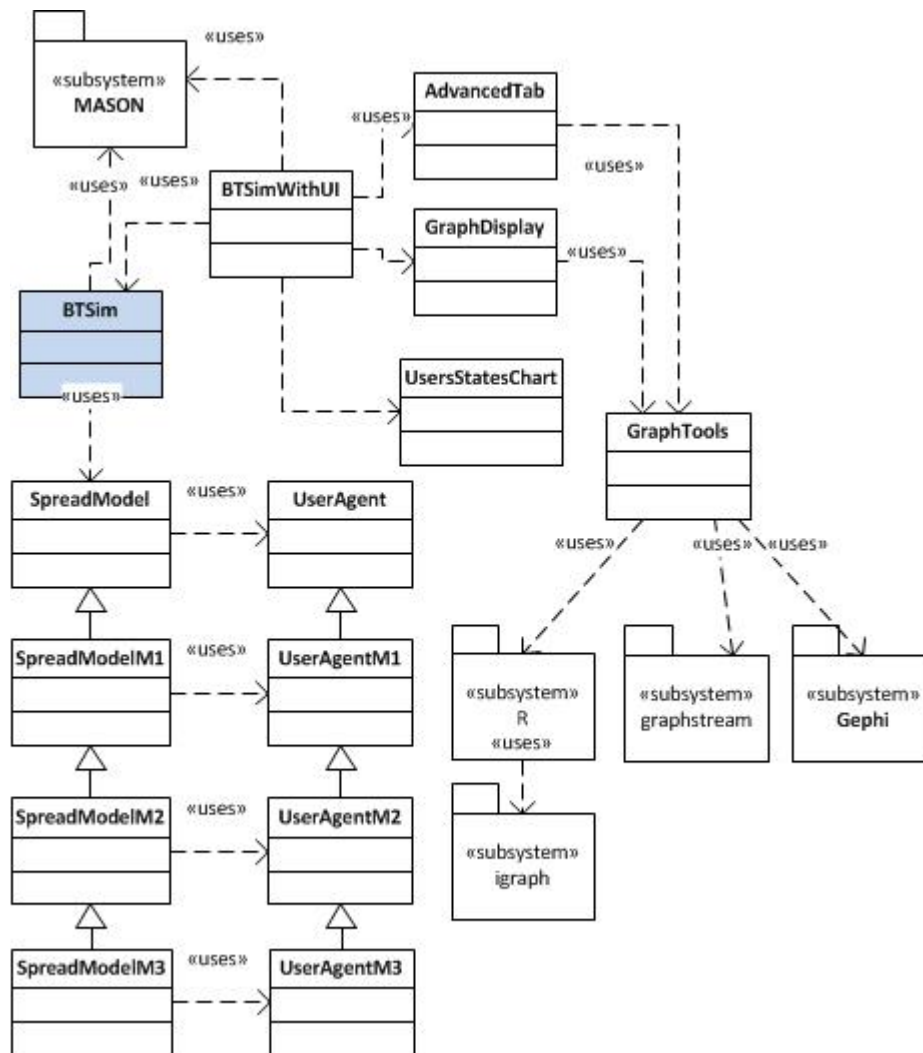


Ilustración 3: Diagrama de clases UML de BigTweet

2.2.2. BigTweet javadoc

BigTweet javadoc se encuentra disponible online¹. Incluye las siguientes clases Java:

- [AdvancedTab](#)
- [BTSim](#)
- [BTSimWithUI](#)
- [Clicks](#)
- [GraphDisplay](#)
- [GraphTools](#)

¹<http://www.gsi.dit.upm.es/~eserrano/BigMarket/P5.3>

- [SpreadModel](#)
- [SpreadModelM1](#)
- [SpreadModelM1.State](#)
- [SpreadModelM2](#)
- [SpreadModelM3](#)
- [UserAgent](#)
- [UserAgentM1](#)
- [UserAgentM2](#)
- [UserAgentM3](#)
- [UsersStatesChart](#)

3. Pruebas

3.1. Instalación

3.1.1. 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 4: 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 5: 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 .

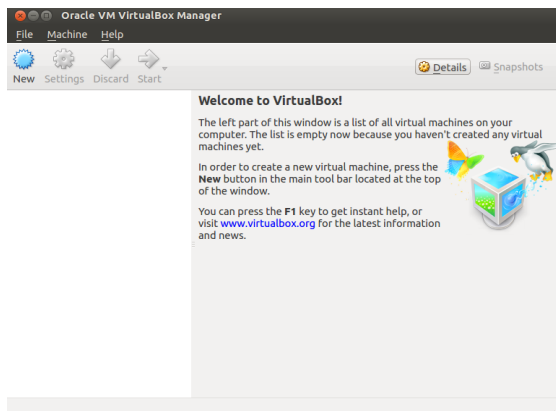


Ilustración 6: Asistente de VirtualBox

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

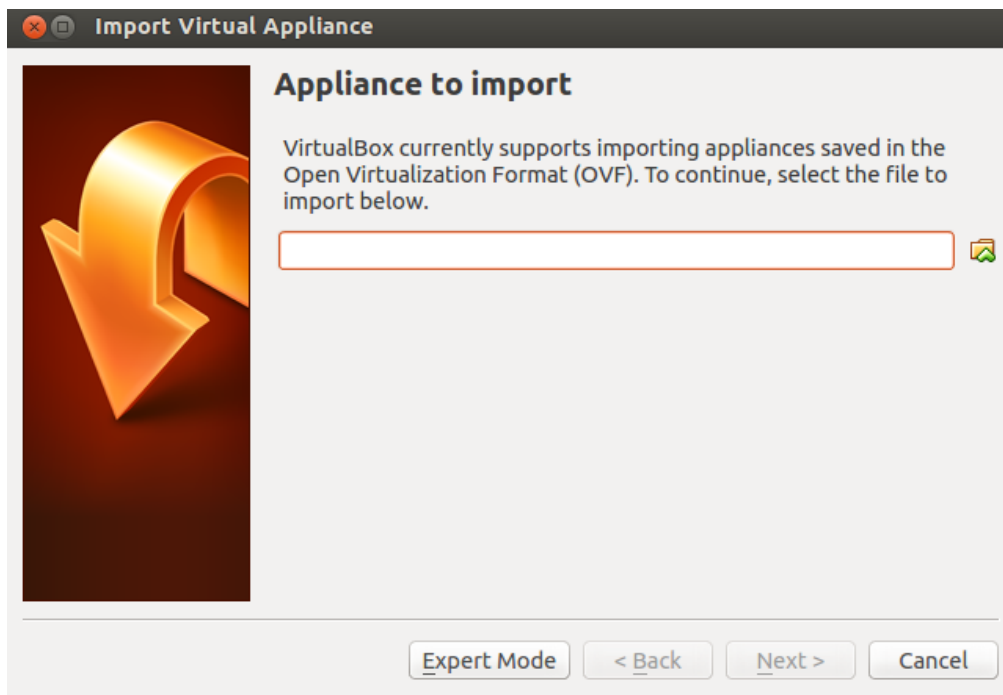


Ilustración 7: Asistente de VirtualBox para realizar la importación de OVF

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.

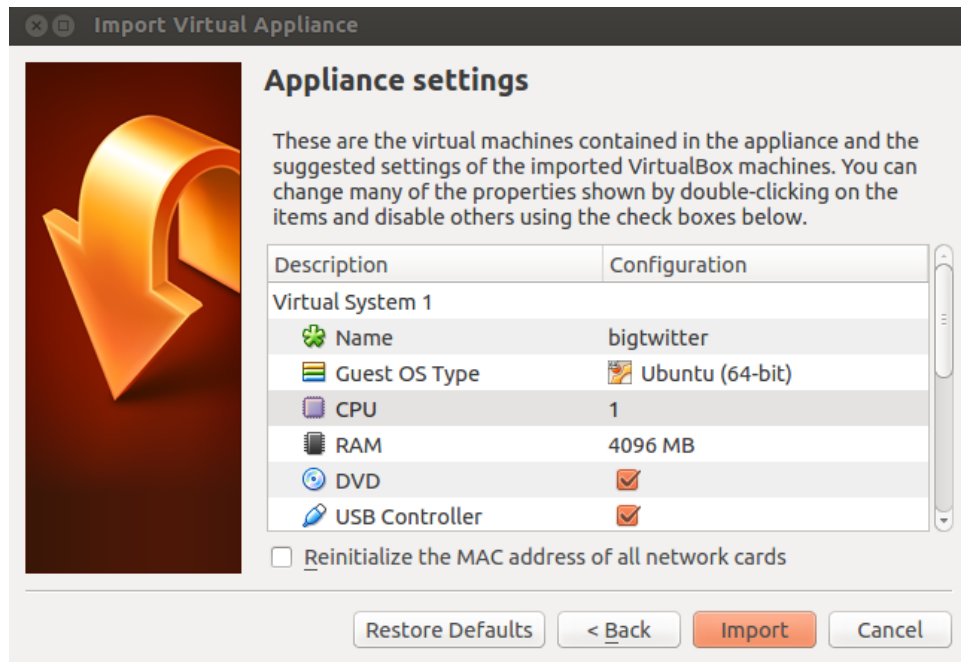


Ilustración 8: Configuración de la máquina virtual

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

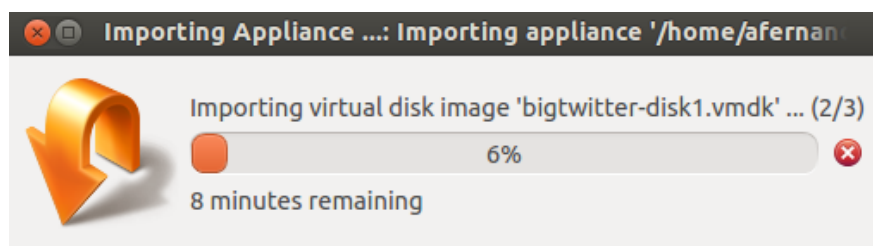


Ilustración 9: Proceso de importación de la máquina virtual

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

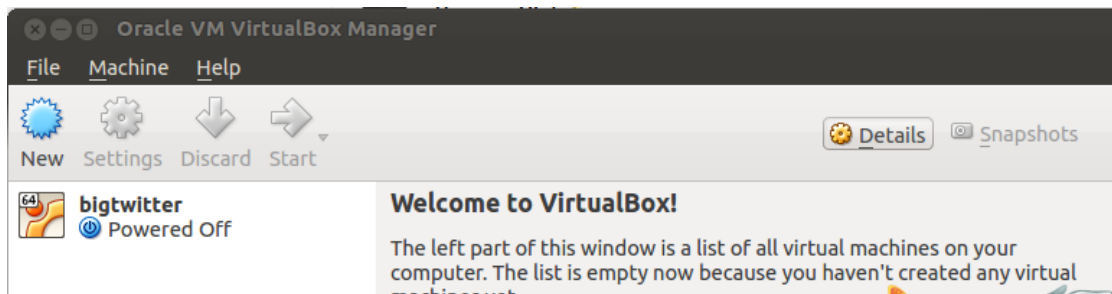


Ilustración 10: Resultado esperado tras importar la máquina virtual

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**

3.1.2. 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 bigtwtter.jar
```

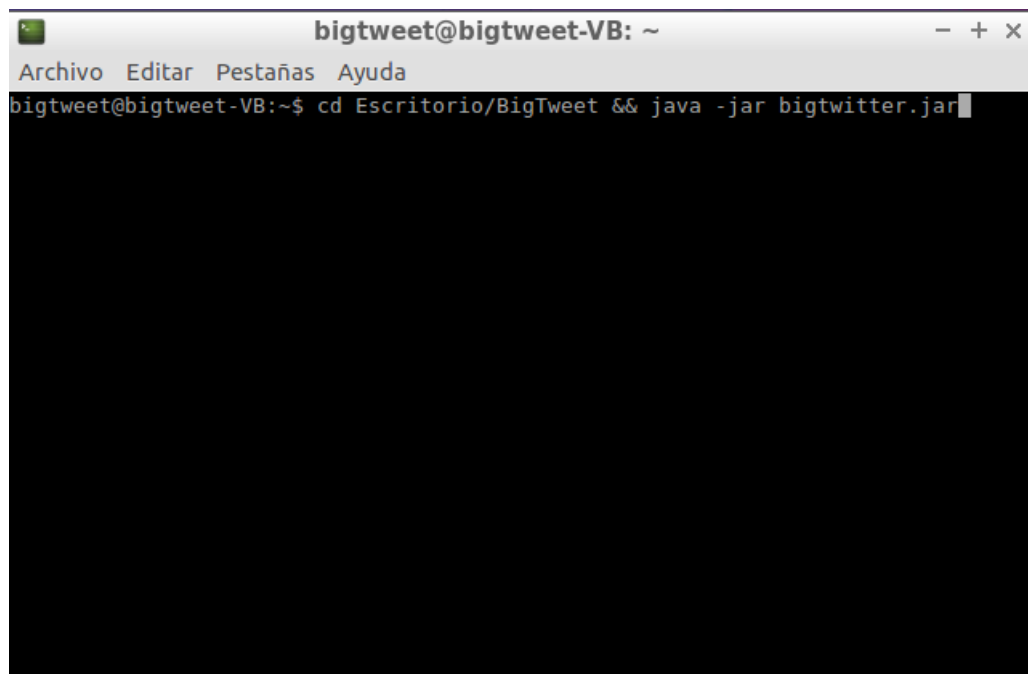


Ilustración 11: consola para lanzar la aplicación

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

3.1.2.1. 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.

3.1.2.1. 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 (betwenness, 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.

3.2. Simulación del escenario de recomendación en social media

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 escenario considerado está definido de acuerdo con el

modelo de simulación social para recomendación en social media. En él, se ha establecido que los usuarios pueden publicar opiniones tanto positivas como negativas sobre la campaña de marketing de una empresa determinada; y además se incluye la figura del usuario especial llamado beacon, que representa una autoridad que detecta la propagación de información errónea y la combate.

El primer paso será la instalación del software BigTweet, descrita en la subsección anterior. 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 12: Pantalla principal del Prototipo

3.2.1. 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 0.016. Además, el número de enlaces con los beacons es de 5 y la centralidad es “betwenness”.

El primer parámetro a configurar será la selección del modelo M3 desde la pestaña de Advanced Options. La prueba inicial se realiza con los siguientes parámetros:

```
"users": 1000,  
"maxLinkPerNode": 1,  
"initiallyInfected":2,  
"probInfect": 0.2,  
"probAcceptDeny": 0.01,  
"probMakeDenier": 0.001,  
"beaconLinksNumber": 5,  
"beaconLinksCentrality": "b",
```

Los resultados obtenidos se pueden observar en las siguientes capturas de pantalla.

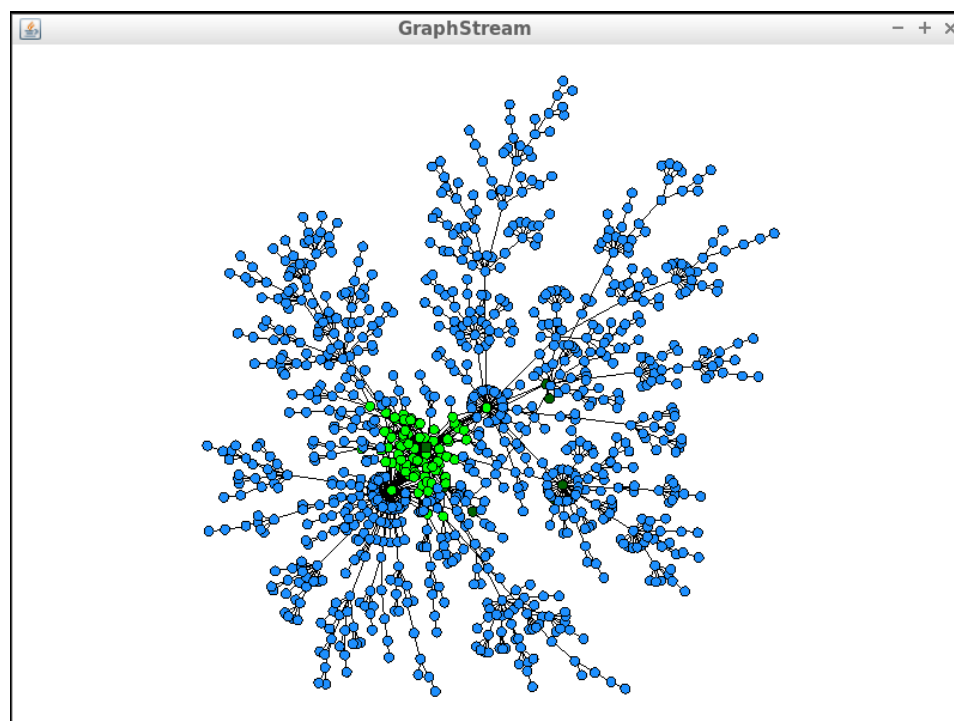


Ilustración 13: Resultado de la prueba

En la captura anterior se puede observar el grafo de la red. Podemos observar cómo los usuarios alrededor de un beacon son curados, mientras que el resto permanecen neutrales. A continuación mostramos la evolución de los estados de los usuarios.

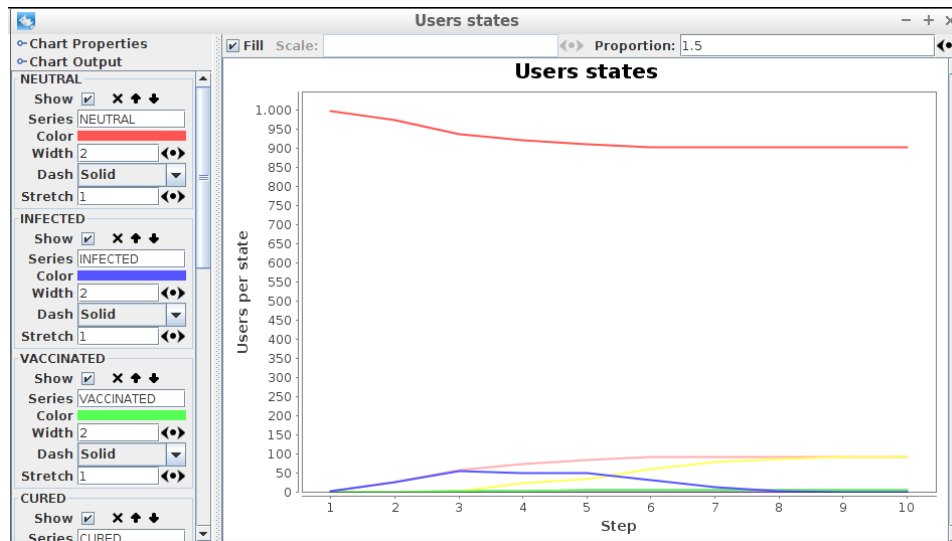


Ilustración 14: Evolución de la prueba

Además, en este caso es posible obtener los nodos de la red mediante el botón “get nodes”, y desactivar los que se crean convenientes para observar los cambios. También es posible cambiar la centralidad.

3.3.2. Prueba aumentando el número de usuarios inicialmente infectados

En este caso se ha aumentado el número de usuarios con opinión negativa inicial de 2 a 200. En este caso el número de usuarios curados finalmente debería ser mucho mayor, pero deben llegar a estar curados prácticamente todos debido a la influencia del beacon. Esto se confirma observando los resultados, mostrados a continuación.

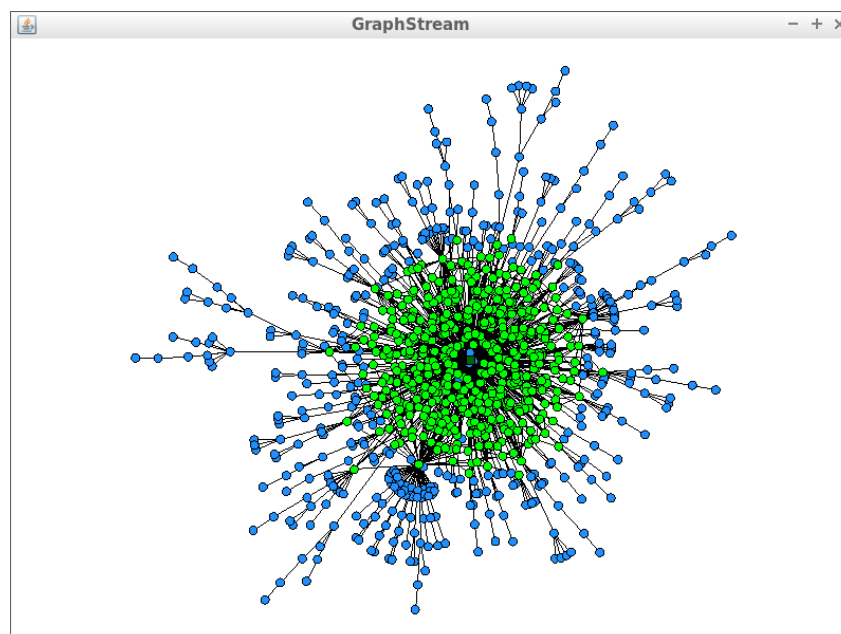


Ilustración 15: Grafo de la prueba

El gráfico de evolución del estado nos muestra cómo los usuarios infectados suben durante unos momentos al comienzo de la simulación, pero después la influencia del beacon comienza a verse reflejada, aumentando el número de usuarios curados y vacunados.

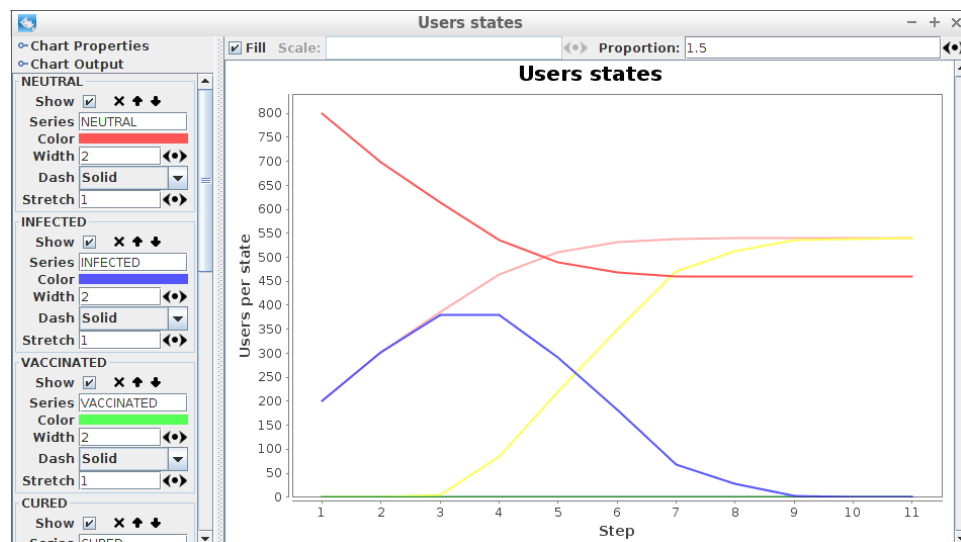


Ilustración 16: Evolución de los estados de los usuarios en la prueba

4. Resumen final y conclusión

Este entregable revisa las estrategias de control para la propagación de rumores en Twitter. Las principales consideraciones para estos modelos han sido introducidas. Se ha revisado e implementado un modelo de control de referencia. Además, se ha propuesto un nuevo modelo de control. Por último, el simulador BigTweet, véase P5.2, se ha ampliado para incluir estos modelos de control.

Referencias

[wik, a] Agent-based modeling software frameworks list.. Accessed February 1, 2015.
http://en.wikipedia.org/wiki/ABM_Software_Comparison

[wik, b] Social network analysis software. Accessed February 1, 2015.
http://en.wikipedia.org/wiki/Social_network_analysis_software.

[twi, a] Twitter REST API console website.. Accessed February 1, 2015.
<https://dev.twitter.com/rest/tools/console>

[twi, b] Twitter REST API documentation website. Accessed February 1, 2015.
<https://dev.twitter.com/overview/documentation>

[twi, c] Twitter terms of use website.
<https://dev.twitter.com/overview/terms/policy>. 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, 2015] Cohen, H. 72 marketing definitions. . Accessed February 1, 2015. <http://heidicohen.com/marketing-definition/>

[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., Flanagan, 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. <https://github.com/gsi-upm/BigTweet>. 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. <http://www.gsi.dit.upm.es/~eserrano/BigMarket/EAAI2015/>
- [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, 2015] Statell, G. Big data and simulations are transforming marketing.. Accessed February 1, 2015. <http://www.businessinsider.com/big-data-in-marketing-2013-5>
- [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.