Computational and Mathematical Organization Theory (2022) 28:112–140
https://doi.org/10.1007/s10588-021-09340-1

MANUSCRIPT

Online discussion threads as conversation pools: predicting the growth of discussion threads on reddit

Sameera Horawalavithana[1] · Nazim Choudhury[1] · John Skvoretz[2] · Adriana Iamnitchi[1]

Accepted: 15 July 2021 / Published online: 27 July 2021

Abstract
This paper proposes a data-driven method that forecasts groups of topic-related, overlapping, online conversation trees. Our method is generative: given a group of original posts, it generates the resulting conversation threads with timing and authorship information. We demonstrate using two large datasets from Reddit that the microscopic properties of such groups of conversations can be accurately predicted when starting from the original posts, without knowledge of the intermediate reactions to such posts. We show that our solution significantly outperforms competitive baselines in terms of predicting the conversation structure and user engagement over time. Potential benefits of this solution include the evaluation of intervention strategies to limit disinformation.

Keywords Reddit · Generative models · Discussion threads

1 Introduction

Discussions on social media have significant impact on society. From recruitment to political movements to disinformation campaigns, social media discussions are the driving mechanism for information diffusion and user engagement.

✉ Sameera Horawalavithana
sameera1@usf.edu

Nazim Choudhury
nachoudhury@usf.edu

John Skvoretz
jskvoretz@usf.edu

Adriana Iamnitchi
anda@cse.usf.edu

[1] Department of Computer Science and Engineering, University of South Florida, Florida, USA

[2] Department of Sociology, University of South Florida, Florida, USA

A particular variation of online discussions is a conversation tree, as seen on Reddit, StackOverflow, or Digg. In Reddit, for example, conversations are grouped on user-defined topics (subreddits). The root of the conversation tree is an original post by a registered user; users respond with comments to the original post or to other users' comments, repeatedly getting involved in the same conversation.

Forecasting how conversations will evolve on such platforms is useful to many applications. For example, while it is difficult to know how many users follow a conversation over time without contributing to it, the number of users who contribute can help estimate the number of users exposed to the conversation. Yet predicting the number of users who will contribute over time to a conversation is challenging, because a user can engage multiple times in the same discussion thread; the same user can participate in multiple related conversation threads, thus affecting the overall audience size; simultaneous related conversation threads might compete for the attention of the same users, thus impeding or accelerating their involvement. At the same time, forecasting user engagement over time with a particular topic across multiple conversation threads can be used to trigger the intervention of a subreddit administrator, for example, if the original posts are predicted to create unwanted engagement such as a coordinated disinformation campaign that is not likely to pass unnoticed. It can also be used to evaluate intervention techniques to encourage engagement (e.g., in the case of health information dissemination) or limit misinformation (e.g., by evaluating how misinformation diffuses if some accounts are prevented from engaging).

Much of previous work has focused on predicting isolated properties of individual social media conversations such as size (Yu et al. 2015), temporal growth (Li et al. 2017), and virality (Cheng et al. 2014). However, these efforts assume to know the initial growth of a conversation to predict the property of interest in the remainder of the conversation. The initial growth of a conversation in the first few hours has been shown to be most useful on predicting the future growth of the conversation (Gao et al. 2019). Moreover, few efforts provide *generative* predictions for overlapping, topic-related conversations or evaluate their predictions of individual conversations as part of a group of overlapping conversations (Krohn and Weninger 2019; Bollenbacher et al. 2021).

This paper proposes a method for forecasting the ensuing conversations with timing and authorship properties when given a set of topic-related original posts in a continuous interval of time on a platform. We show that this method accurately predicts the microscopic properties of a pool of conversations, such as how conversations evolve over time, who its author is likely to be, and the timing associated with each message. The contributions of this work are the following. First, in contrast to most related work, our approach is generative: we predict whether, when and by whom a comment will be made in response to a post or another comment. This contribution is evaluated in terms of conversation structure and user engagement over time. Second, our approach is focused on predicting the microscopic properties of a pool of conversations, and thus it focuses on groups of conversations instead of individual conversations. We show that this focus is beneficial in accurately predicting the collective behavior of users who participate in multiple conversations. And third, our method only assumes the original post/root of individual conversations, without initial reaction information.

2 Related work

Most of the previous work has been focused on modeling and predicting properties of individual cascades (Gao et al. 2019). To predict the size of a cascade, several solutions have been proposed using statistical approaches (Liben-Nowell and Kleinberg 2008; Zhao et al. 2015), while others used machine-learning methods with domain-specific features (Cheng et al. 2014; Yu et al. 2015). More recent work used deep learning models to avoid manual feature engineering tasks on cascade prediction tasks. Embedded-IC (Bourigault et al. 2016) embed cascade nodes in a latent diffusion space to predict the temporal activation of a node. DeepCas (Li et al. 2017) proposed a diffusion-embedding framework to predict the incremental growth of a cascade. Both Embedding-IC and DeepCas exploit the paths in a cascade to improve the accuracy of the prediction task. Topo-LSTM (Wang et al. 2017) utilizes the underlying cascade structure to predict the future node activation in a cascade. They differentiate active/non-active nodes by learning node-embedding vectors for both senders and receivers in the cascade. DeepDiffuse (Islam et al. 2018) and DeepInf (Qiu et al. 2018) utilized the underlying cascade structure to predict the future node activation in a cascade using the recurrent neural network. Specifically, DeepDiffuse predicts the user and the timing of the next infection, but does not predict the evolving cascade structure. They also assume any node can be infected once during a cascade. Chen and Deng (2020) proposed RBMHDRN to predict whether a particular user would retweet a given piece of content or not on Weibo. They extracted a various set of content, user, and network related features to solve this classification task. However, in most of these articles, the prediction tasks either focus on classifying viral cascades (Cheng et al. 2014; Yu et al. 2015) or the future activation of a node over discrete time intervals (Li et al. 2017; Manco et al. 2018). Specific to Reddit, studies focus on predicting the popularity of posts (Fang et al. 2016), detecting influential users (Singer et al. 2014), understanding the user mobility patterns across subreddits (Tan 2018), and predicting many other macro-level properties (Medvedev et al. 2019). Dutta et al. (2020) predict the volume of Reddit discussions leveraging the text from news and initial set of comments using a recurrent neural network architecture. Zayats and Ostendorf (2018) proposed a graph-structured LSTM model to capture the temporal structure of a conversation. They show the effectiveness of the model on predicting the popularity of Reddit comments. In our work, we use a similar data-representation in a generative technique to build the complete Reddit conversations with user and timing information.

While significant work has focused on predicting individual cascades, less attention has been invested in predicting groups of cascades on the same platform defined by time locality. Few studies utilize the information drawn from a group of cascades to predict a single property of interest (e.g., intensity of user activities, user participation in a cascade). For example, several works predict the aggregate volume of user activities on Twitter via Hawkes processes that model the events around a group of cascades (Valera and Gomez-Rodriguez 2015; Zarezade et al. 2017). Myers and Leskovec (2012) predict the future action of a user in a cascade given her previous exposures to multiple other cascades on Twitter. In a similar setting, Weng et al. (2012) developed an agent-based model to predict the probability of a user performing a retweet when exposed to multiple memes on Twitter. They discovered an adversely negative and positive effect on simultaneous cascades that are of unrelated and related content, respectively. Krishnan et al. (2016) extracted several structural features from a set of cascade trees (i.e., a forest of cascades) to distinguish viral cascades from broadcasts. Theoretical models that capture the spread of social-influence when a group of competitive cascades evolve over a network (He et al. 2012; Lu et al. 2015) have also been proposed. Other works have made similar observations when exploring inter-related cascades in multiplex networks (Xiao et al. 2019). These studies stress the importance of focusing on a group of cascades instead of an individual cascade for improving the prediction results of user-level diffusion behavior.

Many recent generative models for conversation trees are statistical approaches that have focused on predicting an individual tree structure (Aragón et al. 2017a; Medvedev et al. 2019; Ling et al. 2020). These data-driven models attempt to capture and interpret some interesting phenomena of a given dataset, by estimating statistical significance of different features related to human behaviour, in contrast to the fully data-driven models. Consequently, the predictive performance of these parsimonious models may deteriorate due to the dependence on the chosen parameters and optimization of the likelihood function. Further, these models lack the capability of mapping the users and exact timing information to the internal nodes (Aragón et al. 2017a). Wang et al. (2012) present a tree generation approach in the dynamic setup. The authors proposed a theoretical model to capture the temporal evolution of conversation trees by employing a Levy process to attach timing information. They used preferential attachment mechanism to build the conversation tree. Aragón et al. (2017b) used reciprocity (i.e., strong exchange of messages between users) as a behavioral feature to predict the temporal evolution of a conversation with respect to the depth of a tree. The proposed statistical approach utilizes the mutual dependency between the authorship and conversation structure. Several works (Medvedev et al. 2018; Krohn and Weninger 2019) model the dynamics of conversation trees using a Hawkes process. Medvedev et al. (2018) did not use any of the conversations in the training data to estimate the parameters in the Hawkes model. They estimated the parameters from the initial comments of a conversation to predict the remainder. Krohn and Weninger (2019) improved the previous solution in the proposed CTPM model as the parameters are estimated from the post information. More recently, Bollenbacher et al. (2021) proposed the Tree Growth Model (TGM) to predict the final size and shape of conversations given the partial conversation tree information. CTPM and TGM are the closest works to our problem setup. However, these models do not assign the author information to comments, and do not account a pool of conversations in the problem setup. We acknowledge the recent challenges highlighted by Bollenbacher et al. (2021) on the difficulty of predicting individual user actions. For example, Krohn and Weninger (2019);

Bollenbacher et al. (2021) argue that predicting microscopic user actions is difficult in the long-lived online conversations with a pure generative approach. With this prior experience taken into account, we present a simulator design that combines both discriminative (machine learning) and generative approaches.

Other recent attempts to model online social behavior are as part of the Computational Simulation of Online Social Behavior (SocialSim) program sponsored by the Defense Advanced Research Projects Agency DARPA (2021). Abdelzaher et al. (2020) proposed SocialCube, an agent-based approach to predict social media activities. This solution decides optimal agent-specific configurations from past social media traces. Garibay et al. (2020) proposed DeepAgent to simulate the social media activity in the population, community, user, and content levels. This framework used a generative rule-driven approach where specific rule sets were built to model agent behavior using both endogenous and exogenous signals. While we have similar objectives, our solution is not composed with specific individual agents' actions or hand-crafted rule sets.

This paper proposes a data-driven model to predict conversation trees with author identities and continuous timing information mapped onto the nodes in the tree. We use machine-learning models to capture the dependence between authorship, timing and structure in a conversation. The approach used is to consider the group of simultaneous conversations that take place during a fixed interval of time on the same platform, as such conversations may share users or, alternatively, preclude users from participating in simultaneous conversations.

3 Predicting pools of conversations

Our objective is to predict the microscopic properties of a set of possibly interrelated, simultaneous conversations over time. The operational scenario we are considering is the following: given the initial postings described by content, author and time on a given social platform (such as the four messages depicted on the horizontal time axis in Fig. 1), generate the emerging discussion threads by specifying which message is in response to which message, and the author and time of each message. Each discussion thread generated will be represented as a conversation tree, where a child node is a message in response to its parent node in the tree; users can engage repeatedly within a conversation; the delay in responding to a previous message is unbounded; and a user may respond to his own message, typically with additions or clarifications. Table 1 presents the terminology used in this paper.

Our solution is as follows. We generate probabilistically pools of independent conversation trees rooted in each input seed. We assign users and timing information to all nodes in every conversation tree. We thus end up with naive groupings of independent conversations, where user and time assignment to messages in a conversation are oblivious to what happens in other conversations in the same pool (Sect. 3.1). We then use a genetic algorithm to reconstruct a realistic pool of conversations from the arbitrarily generated ones (Sect. 3.2).

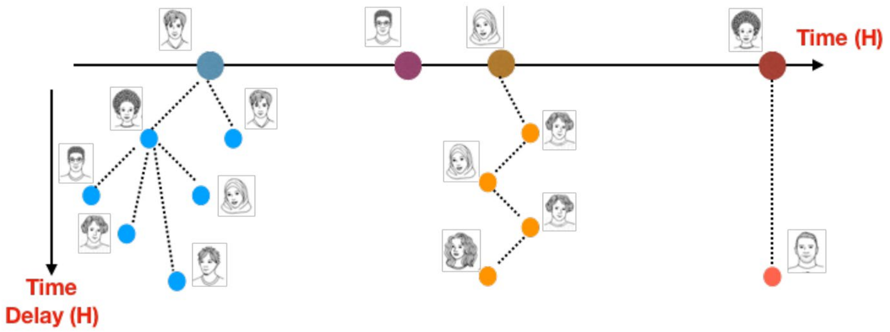


Fig. 1 Sample scenario where, given four original posts, the objective is to generate the corresponding conversation trees given that previously unseen users can engage in conversations; messages may be posted with unbounded delay; some original posts will remain unanswered; the conversation trees will have highly different structures; and users may engage repeatedly with the same or different conversations

Table 1 Terminology used in this paper

Term	Description
Node	Message in a discussion thread described by content, author, and posting time
Conversation tree	A conversation thread represented as a tree of messages, as shown in Fig. 2a
Conversation pool	A collection of conversation trees within a finite period
Conversation size	The number of messages in a conversation
Conversation pool size	The total number of messages in all the conversations in a pool
Conversation depth	The number of levels in a conversation tree
Conversation breadth	The number of messages in a given level of a conversation tree
Node degree	The number of immediate messages in response to the parent message
Seed	A message at the root of the conversation tree
Propagation delay	The time difference between the posting of a message and that of its parent
Structural virality	Wiener Index of a conversation tree (Goel et al. 2015)
Collectivity	Group behavior of users engaged in multiple conversations (Lu et al. 2018)

3.1 Generating pools of conversations

We employ the branching model (Cheng et al. 2016) to construct pools of conversations. We are building on research (Cheng et al. 2018) that shows that branching models based on node degree distributions can be used to accurately generate subtrees of conversations. In this work, we extend this technique to generate temporal conversation structures of any depth while attaching user information.

We build each conversation tree recursively, as presented in the function generate_conversation in Algorithm 1. From the training dataset that contains a large number of conversation trees, we build degree distributions per level. Thus, for each level, we will have a degree distribution for the nodes located at that level across all conversation trees. The node degree is defined as the number of children of that node in the conversation tree. Given an initial seed that functions as the root of the conversation tree to be generated, we recursively build tree structures by selecting node degrees from the degree distribution of the corresponding level. For a set of *n* input seeds, we thus generate *n* independent conversation trees that we consider a pool.

Algorithm 1 Probabilistic Generation of a Conversation Pool

```
PREREQUISITES: degree distributions per level of a conversation
INPUT: parent node
OUTPUT: a conversation tree

function GENERATE_CONVERSATION($parent, conversation$)
    if $parent$ is $NULL$ then
        $root\_level \leftarrow 0$
        $root\_degree \leftarrow$ SAMPLE_DEGREE($root\_level$)
        $parent \leftarrow Node(root\_level, root\_degree)$
        $conversation.set\_root(parent)$
    $level \leftarrow parent.get\_level()$
    $N_{children} \leftarrow parent.get\_degree()$
    for $j \leftarrow 1$ to $N_{children}$ do
        $child\_level \leftarrow level + 1$
        $child\_degree \leftarrow$ SAMPLE_DEGREE($level$)
        $child \leftarrow Node(child\_level, child\_degree, parent)$
        $conversation.set\_child(child)$
        return GENERATE_CONVERSATION($child, conversation$)

INPUT: $seeds_1 \ldots seeds_N$
OUTPUT: a pool of conversations

function GENERATE_CONVERSATION_POOL($seeds[]$)
    $seed\_size \leftarrow length(seeds)$
    $conversation\_pool \leftarrow []$
    for $k \leftarrow 1$ to $seed\_size$ do
        $conversation \leftarrow Tree()$
        $conversation.set\_root(seeds_k)$
        GENERATE_CONVERSATION($seeds_k, conversation$)
        $conversation\_pool[k] \leftarrow conversation$
    return $conversation\_pool$
```

In order to assign authors to nodes in a conversation tree we exploit the social network topology of previous user interactions. Specifically, from the training dataset, we extract the interaction network in which vertices are users and directed edges represent previous interactions. We also extract edge weights that represent the number of previous interactions. Note that a user can be part of her own neighborhood if she replied to her own post in the past. This is reflected by a weighted self-loop in the network. We use this directed, weighted interaction network to bias the assignment of users to messages as follows. We start with a conversation tree, as generated above, whose root has a user assigned (from the input data). Recursively, for every node with a user u assigned, we probabilistically select d users from u's neighbors $N(u)$ in the interaction network and assign them as authors to the node's d children. If $d > N(u)$, we add $(d - N(u))$ new users who are previously not seen in the training data to the chain of responses. We bias the probabilistic selection using the weights in the interaction network. Note that this approach allows for the same user to participate multiple times in the conversation tree.

In order to assign time to nodes in the conversation tree, we use a propagation delay distribution conditioned by the size of the conversation. We consider the propagation delay as the difference between the time of each comment and the time of parent comment/ post in the training dataset. For each conversation, we extract the size of the conversation and the sequence of propagation delays. In the generated conversation, we use the size of the conversation as resulted from the generation process (Algorithm 1) to randomly select a sequence of propagation delays from a previously seen conversation of that size. We sort the nodes of the generated conversation by level, assign the propagation delay to nodes, and compute the message time using the time of the seed message and the assigned propagation delay.

After this procedure, we end up with conversation trees rooted in the original message from the input data, in which each message node has a user and a time assigned. This simple probabilistic approach generates pools of independent conversations that ignore multiple aspects of real-world behavior, such as users participating in multiple conversations within the same period of time or, alternatively, being unable to participate simultaneously in many conversations. During empirical evaluations based on a variety of performance metrics that will be described later, we observed that all pools perform comparably and poorly compared with testing data.

3.2 Reconstructing a realistic pool of conversations

Ideally, given a set of possible pools of n conversations each corresponding to the n input seeds, we would construct a new pool consisting of the "best" conversation for each seed. However, there are two challenges. First, it is impossible to know which conversation is the best before the testing of the entire pool. This is mainly due to the huge variation of possible conversations that can be generated randomly.

Second, using a single performance metric that evaluates the "goodness" of individual conversations, selecting a pool of the best such individual conversations does not lead to a pool good enough in other metrics. For example, a pool constructed from the best individual conversations according to structural properties metrics might evaluate poorly in user-level metrics.

To address these challenges, we treated the pool reconstruction problem as an optimization problem that we solved using a genetic algorithm. As the fitness function in the genetic algorithm, we used the output of two trained machine learning models to evaluate the goodness of a conversation.

3.2.1 Modeling the problem for a genetic algorithm

Genetic algorithmic representations provide powerful search heuristics for complex search spaces (De Jong 1990). To proceed with the standard steps of genetic algorithms, we map our problem into the genetic algorithm context as follows: We consider a *gene* an individual conversation, represented by the message tree with assigned user and timing information to nodes. An *individual* from evolutionary computation is thus a pool of conversations in our context. The *population* is the set of conversation pools we generated with the probabilistic approaches described earlier. The objective is to create a pool of conversations which outperforms any existing pool of conversations.

We use the standard framework of a genetic algorithm and repeat the process until there is no improvement in the best solution. We start with the initial set of conversation pools as described earlier. We measure the fitness of a conversation pool using two trained machine-learning models as described next. For mate selection, we rank the conversation pools according to the fitness function and consider only the top 80%. Given a pair of conversation pools selected from a top-ranked pool and a least-ranked pool, we randomly draw conversations to form a new pool for the next generation. The new generation entirely consists of mated conversation pools from the top 80% of the conversation pools in the previous generation. Accordingly, we re-construct a number of new pools across generations.

We do not use mutations in this approach for the following reason. Mutations require to modify the initial conversation structures (with user and timing information) generated by the probabilistic model. The mapping of users to the internal conversation nodes is done via a recursive chain of user assignments using the interaction network. When we modify the structure, this method of mapping users becomes obsolete, and lead to inaccurate view of user responses.

We summarize all algorithmic steps in Algorithm 2.

Algorithm 2 Selection of the Best Conversation Pool with a Genetic Algorithm

```
INPUT: a set of conversation pools, $\gamma$ the probability of mate selection
OUTPUT: a set of re-constructed conversation pools

function nextGeneration(P,$\gamma$)
    $P \leftarrow$ rank_pools($P$)
    $P_{mates} \leftarrow$ select_best_pools($P, \gamma$)
    $P_{gen} \leftarrow$ reconstruct_pools($P_{mates}$)
    return $P_{gen}$

INPUT: initial set of conversation pools
OUTPUT: best conversation pool

function generate(P,$\gamma$, $N_{Gens}$)
    for $N_1 \leftarrow 1$ to $N_{Gens}$ do
        $P \leftarrow$ nextGeneration($P, \gamma$)
    $P \leftarrow$ rank_pools($P$)
    $pool \leftarrow$ select_best_pool($P$)
    return $pool$
```

3.2.2 Ranking pools of conversations with machine learning

In order to rank the pools of conversations, we assign a goodness score to each conversation in the pool and consider the sum of all such scores as the goodness score of the pool. The goodness score of each conversation has two components: a score relative to the structural properties (i.e., shape of the conversation tree), and a score relative to the timing of the nodes in the conversation. Specifically, we feed each conversation into two trained machine-learning models to assess the goodness of the branching factor and propagation delay with respect to the attached user information and semantic structure.

Training: We use two individual-level properties—branching factor and propagation delay—of conversation nodes as the target units for the prediction tasks. Any information regarding branches is important for the accurate creation of the conversation structure as they evolve in the form of sub-trees under the same original post or another comment (Medvedev et al. 2019). Therefore, we first classify the messages as leaf or branch nodes in the tree. Note that these node positions determine the shape of the conversation.

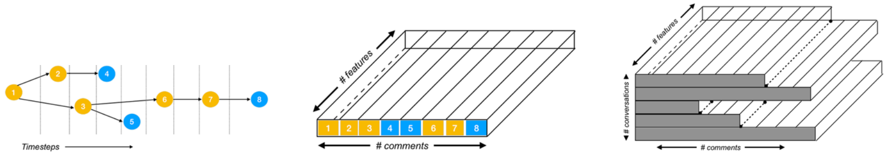
We classify the messages by the delay with which they are posted in response to their parents to distinguish fast-paced conversations from slow-paced conversations. We consider the median propagation delay within a conversation as the borderline between the two classes: messages with a propagation delay larger than this median are called late adopters, while the others are early adopters. We used this binary classification approach to seek the hourly time granularity predictions. We discovered empirically that the median propagation delay is close to 1.5 h and a binary classification satisfies the hourly granularity. For finer time granularity, we might need classifying propagation delays by quartiles, or predicting exact propagation delay value in seconds. This would remain as a future work to improve the time predictions (Tables 2, 3).

Each message is described by features from three main categories: (i) spatio-temporal features, that capture the position of an individual message in a conversation; (ii) user features; and (iii) content features. These features are detailed in Table 4.

We use the LSTM model to capture the chronological order of messages in a conversation. The input to the LSTM algorithm is a conversation as shown in Fig. 2b. We use the memory-cell design of a standard LSTM in our work (Hochreiter and Schmidhuber 1997) which is implemented in Keras (Chollet et al. 2015). Our LSTM setup includes two blocks of memory-cells with 32 and 8 hidden units, and we use Adam algorithm for the optimization with a learning rate of 0.001 based on hyper-parameter optimization. Conversation representations are different in shape mainly by the number of messages in the online conversation, and thus we input them one by one for training.

Testing: During testing, we extract the features described in Table 4 from the generated conversations. The activity-level features of the users in a particular conversation are constructed considering their activities in other conversations. To account for the interaction among multiple conversation trees, we dynamically update the user features. Specifically, when a user j is assigned to a message in a new conversation tree at time t, her activity features such as the number of past activities $A_j^{t'}$ at time $t' < t$ is updated to $A_j^t = A_j^{t'} + 1$. Since we do not predict the content of the messages in a conversation, we assign content-level features to messages in the testing period randomly based on previously seen conversation nodes in the same level.

Once we construct the data structure shown in Fig. 2b with all necessary features, we infer one binary vector that represents branch/leaf using the branch discriminator model, and another binary vector that represents the early/late adopters using the delay discriminator model. We consider these two binary vectors as the inferred ground truth to assess the generated conversation. The assessment is done by comparing the inferred ground truth with the same binary vectors extracted from the generated conversations using area under the curve (AUC). Each conversation receives a goodness score as the mean of two AUC scores



(a) Conversation Tree (b) Individual Conversation (c) A Pool of Conversations

Fig. 2 Representation of conversation trees. **a** Nodes (messages) are ordered chronologically. Yellow nodes represent internal nodes and blue nodes are leaf nodes. **b** Each node is represented by a spatio-temporal feature vector. Feature vectors are ordered chronologically and grouped by conversation. **c** Multiple conversations of arbitrary sizes are stacked together for training/testing

Table 2 Subreddits used for data collection

Domain	List of subreddits
Crypto-currency	/r/Bitcoin, /r/Ethereum, /r/Monero, /r/Paycon, /r/DopeCoin, /r/Lisk, /r/Donationcoin, /r/Pivx, /r/Orocoin
Cyber-security	/r/netsec, /r/netseclounge, /r/technology, /r/techsupport, /r/pcmasterrace, /r/linux, /r/hacking, /r/Piracy, /r/sysadmin, /r/HowToHack, /r/privacy, /r/Windows10, /r/programming, /r/networking, /r/softwaregore, /r/compsci,/r/talesfromtechsupport, /r/msp, /r/security, /r/SocialEngineering, /r/Malware, /r/AskNetsec, /r/blackhat, /r/ReverseEngineering, /r/crypto, /r/pwned, /r/netsecstudents, /r/securityCTF, /r/hacktivism, /r/browsers,/r/linuxadmin, /r/websec, /r/antivirus, /r/Ransomware, /r/Pentesting, /r/OpenHacker, /r/blackhatting, /r/Android

We collected 0.2M conversations from nine subreddits related to crypto-currency and 1.76M conversations from 38 subreddits related to cyber-security

from the two models. The goodness of a pool of conversations is the sum of the goodness scores of the conversations in the pool. We use this pool goodness score to rank the pools of conversations in the genetic algorithm.

Table 3 Properties of reddit conversations in our datasets

Measurement	Crypto	Cyber
Number of conversations	209,721	1,762,977
Number of messages	3,580,162	35,381,971
Number of distinct users	144,457	1,647,789
Max conversation lifetime (days)	311	910
Max conversation size	7868	74,032
Max conversation depth	160	971
Max conversation breadth by level	7578	72,955

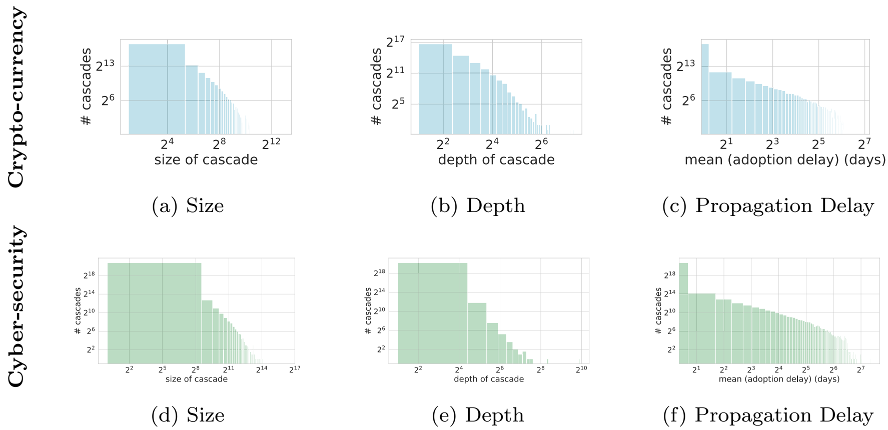
4 Datasets

For empirical evaluations, we focus on Reddit conversations. We selected two active topics, crypto-currency and cyber-security, as our two topic-driven separate datasets. We extracted all conversations between January 2015 and August 2017 posted under the topic-related subreddits and listed in Table 2. Both datasets were provided privately as part of DARPA SocialSim program.

We represented each conversation thread as a conversation tree. A node in the conversation tree consists of the textual content of a Reddit message (post or comment) and its author. A pair of nodes (source to target) are connected by a directed edge where the direction suggests that the target node reacts to (content posted by) the source node. Table 3 presents the structural properties of the conversations in the two datasets. The cyber-security dataset is nearly 10 times the size of the crypto-currency dataset in the total number of messages posted. The properties of the conversation trees are also highly different in scale: the largest conversation in cyber-security contains 74K messages, while in crypto-currency is 7.8K. The depths of the conversation trees are also almost an order of magnitude apart: 971 vs 160. Irrespective of the size and depth disparities, we observe that Reddit conversations are viral and broad. They include both slow (cyber-security) and fast-paced (crypto-currency) conversations which can be active for short and long periods. (as seen in Fig. 3). Moreover, the discussions originated from crypto-currency subreddits exhibit diverse characteristics related to the scale and speed of discussion spread (Glenski et al. 2019). While we believe the Reddit discussions originated from cyber-security and crypto-currency topics might show unique characteristics compared to the Reddit discussions originated outside these topics, they represent a focused group of users engage on a set of operationally relevant topics. In future work, we plan to evaluate our approach on a broader community (i.e., political, sports, entertainment etc.).

From these datasets we extract three groups of features (detailed in Table 4): (i) structural features, (ii) user-level features, and (iii) content-level features.

Structural features: We represent the topology around an individual node in the conversation using two spatio-temporal properties: degree and the birth order of the predecessors. As an example, we use the degree and birth order of the parent (level $i-1$) and the grand-parent (level $i-2$) nodes to represent a node in level i.



(a) Size

(b) Depth

(c) Propagation Delay

(d) Size

(e) Depth

(f) Propagation Delay

Fig. 3 Basic characteristics of reddit conversations. The distribution of cascades is presented by (**a**, **d**) size, (**b**, **e**) depth, (**c**, **f**) the mean delay between the time of a comment and the time of the original post as observed in the conversations

Table 4 Features used to represent a message (node) in a reddit conversation

Feature domain	Feature description
Structural features	Number of comments for comment/post
	Adoption delay from the parent comment/post
	Adoption delay from the root post/root
	Level of the conversation tree
	Birth order of comment
	Number of comments for the parent comment/post
	Birth order of the parent comment
	Number of comments to the grandparent comment/post
	Birth order of the grandparent comment
User features	Total number comments received by the comment author in the past
	Total netscore (upotes−downvotes) of the comment author in the past
	Total number comments made by comment author in the past
Content features	Netscore of the comment
	Subjectivity score of the comment
	Controversiality score of the comment
	Netscore of the parent comment
	Subjectivity score of the parent comment/ post
	Controversiality score of the parent comment
	Netscore of the grand parent comment
	Subjectivity score of the grand parent comment/ post
	Controversiality score of the grand parent comment

User-level features: Actions in a conversation could be in response to the users who authored the previous message rather than simply to the content with which the users interact. We thus represent a user via a set of features describing her status on the platform, measured by the amount of activity she has done prior to the particular reaction. Such activities reflect user's interest on other conversation threads. We also extract the popularity of the user in terms of *upvotes* and *downvotes* received to her posts or comments in the past. These endorsements summarize the influence of a user in a community.

Content-level features: We extract the sentiment scores of Reddit comments that quantify the subjective and controversial content (a Python library of a natural language toolkit is used to calculate these two features (Hutto and Gilbert 2014)). We also capture the semantic structure of the comments at predecessor nodes. Another useful feature is the popularity of posts or comments that is captured by *net-score*, the difference between up-votes and down-votes received for a particular post or comment from all users.

5 Experimental results

The primary objective of the generative model proposed in this study is to predict the complete conversation structure with authors and timing information. For a comprehensive evaluation, we compare the following outcomes against the ground truth conversations: (i) the structural characteristics in terms of size and virality of the predicted conversations; (ii) the volume as measured in the number of comments generated to the seed posts and audience size as measured in the number of distinct users who participate in the conversations over time, and (iii) the collective behavior of users who engage in multiple conversations. We select three baseline models as described in the sections below to compare against the performance of our model.

5.1 Evaluation methodology

For testing the generated pools of conversations, we used a subset of the testing data as follows. We used as seeds the posts made between August 1 and August 3, 2017 and the resulting conversations as seen by the end of August 2017. There were 3740 and, respectively, 3463 such conversations in the crypto-currency and cyber-security domains. Because seeds are chosen from a continuous time interval, the ensuing conversations can overlap in time.

We compare the quality of our model with respect to three baseline models. First, we use a state-of-the-art generative model (i.e., Lumbreras Model (Lumbreras 2016; Aragón et al. 2017a)) that predicts the entire structure of the conversation instead of aggregate metrics such as size or virality. The *Lumbreras model* proposed an improved solution compared with a family of generative approaches (Kumar et al. 2010; Gómez et al. 2013) that use the branching process in the generation of conversation structures. A more recent work (Aragón et al. 2017b) which adds reciprocity as a model parameter acknowledges increased computational costs relative to previous work due to various optimization functions. Due to the size of our datasets, we chose to compare with the less computationally intensive Lumbreras model. This model uses the parameters related to popularity, novelty (preference to reply to a newer post), root-bias (preference to reply to a post rather then to a reply itself), and user roles to predict the growth of discussions. We construct ten pools of conversations from this solution to account the bias in parameter selection criteria. However, this model does not assign user information, and maps only discrete timestamps to the generated comments. We do not use Lumbreras model in the temporal measurements due to the mismatch between our continuous time and its discrete time approaches.

Next, we use two baseline models that draw the events from the training data repeatedly into the testing time period. *Baseline* (*recent-replay*) draws the most recent *n* conversations from the training data. *Baseline* (*random*) draws *n* conversations from the training data at random (where *n* is number of seeds in the testing period). We construct ten pools of conversations in the Baseline (random) solution to minimize the bias of random selection. In the baseline solutions, we keep all other event information (e.g., author, conversation structure, etc.) of the conversations except the event timestamps, which are shifted by the time interval between the seed post and their corresponding root message. Because these baseline models repeat events from the recent past, they proved to be very challenging to outperform in simulating user activities in multiple social platforms (Abdelzaher et al. 2020; Bollenbacher et al. 2021), including Reddit (Krohn and Weninger 2019).

In order to evaluate the accuracy of our conversation reconstruction solution, we use several measurements. First, we evaluate the goodness of our fitness score used in the conversation reconstruction algorithm (Sect. 5.2). Second, we present the structure of conversations in the reconstructed pool with respect to size and virality (Sect. 5.3). Third, we evaluate the volume of messages generated from the original posts with respect to the community of users who authored them and timing information (Sect. 5.4). Finally, we quantify the engagement of users in multiple conversations (Sect. 5.5). These metrics are reported in comparison with ground-truth data and the baseline models mentioned above.

5.2 Evaluation of the goodness score of a conversation

We measure the two components of the goodness score: predicting the position of a message as a branch or leaf node in the conversation tree, and the timing of the message as early or late compared to the median propagation delay relative to

Table 5 Reddit conversations grouped by post time

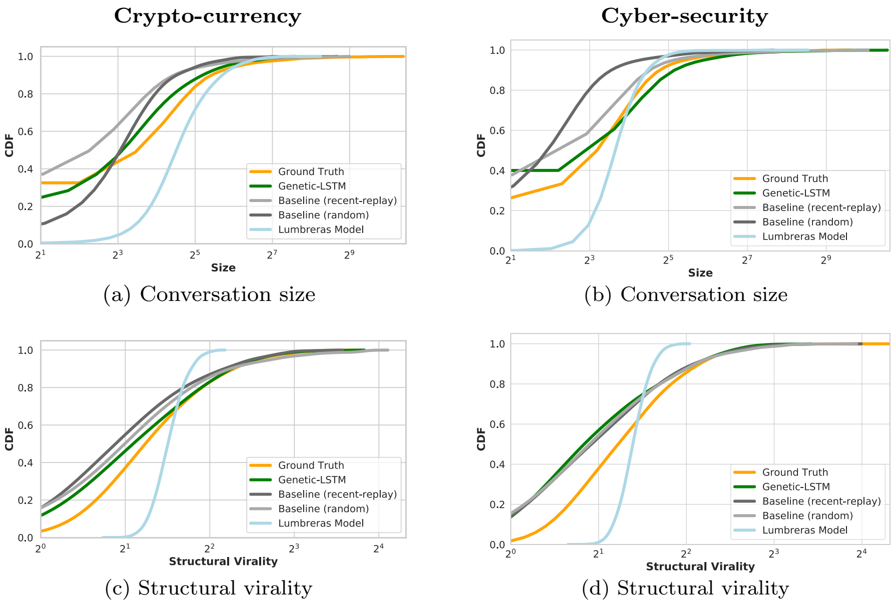
Domain	Training (Jan '15–Jul '17)		Testing (Aug '17)	
	# Conversations	# Messages	# Conversations	# Messages
Crypto	0.19 M	3.3 M	0.02 M	0.25 M
Cyber	1.7 M	34 M	0.06 M	0.9 M

that conversation. We train four LSTM models in total for two training datasets as described in Table 5. The outputs of these LSTM models are used to assess the likelihood of a conversation in the conversation reconstruction algorithm. LSTM-degree models achieve 73–75% F1 score in discriminating leaves vs. branching nodes in respective domains. A majority vote would achieve 65% accuracy on predicting branches as the two classes are balanced in the ratio of 65%:35% across both datasets. The F1 score of our LSTM-delay models in distinguishing between early and late adopters is 83–89% while a random draw should achieve 50% given the perfectly balanced classes.

5.3 Evaluation of the structure of conversations in the pool

To measure the size and structural virality of the generated conversations irrespective of the temporal aspects, we compare the re-constructed conversation pool with the baseline generative approaches.

We show the CDFs of individual conversation sizes and structural virality scores for conversations resulted from our model, the baseline approaches, and the ground truth in Fig. 4. For fairness in evaluating the baseline approach, for the Lumbreras model and Baseline (random) we generated ten solutions for each seed and reported the average. We calculate the absolute percentage error (APE) of the mean size and the mean structural virality between the generated conversations and the ground truth conversations. We also report the JS divergence between the distributions of



(a) Conversation size

(b) Conversation size

(c) Structural virality

(d) Structural virality

Fig. 4 The distribution of conversations reconstructed by the genetic algorithm, compared with ground-truth and Lumbreras-generated conversations

Table 6 Performance of the size and structural virality of the conversations generated by different models

Domain	Model	Size		Virality	
		JSD	APE	JSD	APE
Crypto	Baseline (recent-replay)	0.40	51.7	0.043	17.6
	Baseline (random)	0.14	43.5	0.074	23.7
	Lumbreras model	0.49	37.4	0.046	11.8
	Genetic-LSTM (our solution)	**0.15**	**25.4**	**0.012**	**7.5**
Cyber	Baseline (recent-replay)	0.39	28.9	0.035	14
	Baseline (random)	0.41	57.6	0.036	62.7
	Lumbreras model	0.34	12	0.062	**0.3**
	Genetic-LSTM (our solution)	**0.23**	**8.6**	**0.029**	15.7

We compare the distribution of size and virality of the generated conversations with the ground truth using JS Divergence (JSD). We also report the APE for the mean size and structural virality of the conversations after compared with the respective values in the ground truth. We highlight the lowest JSD and APE values in bold

the structural metrics reported of the generative models and of the ground truth (as shown in Table 6). A lower JS divergence value denotes that the distribution of the sizes/structural virality of the generated conversations is closer to that of the ground truth. We have three observations from these measurements.

First, our solution achieves the lowest JS divergence value after comparing the distributions of sizes and virality scores between the predicted and the ground truth conversations (as shown in Table 6). We also record the mean conversation size closer to the ground truth value across both datasets as shown by the lowest APE values for sizes in Table 6.

Second, we noticed that the mean structural virality scores of the conversations generated by our solution are closer to the ground truth in crypto-currency related discussions (lowest APE values for virality in Table 6) more than the cyber-security related conversations. We believe this is due to the slight over-prediction (12–18%) of the number of smaller conversations (i.e., conversations with size smaller than the median size) compared to what exists in the ground truth. The majority of smaller conversations only have immediate comments to the original post, thus the virality scores are very low.

And finally, we also notice the difficulty of accurately predicting the properties of the largest and most viral conversations. Note that the most viral conversation may not be the largest conversation (Goel et al. 2015). For example, the size of the most viral (virality = 12) conversation is 136, and the virality of the largest conversation (size = 1301) is 5 in crypto-currency discussions. We do not accurately predict the size and virality of such conversations compared to other baseline models (as shown in Table 7). However, we noticed that baseline models are not consistent on achieving the best results across crypto and cyber discussions. These conversations are very rare to observe and are likely to grow under external events (Myers et al. 2012; Goel et al. 2015). These external events may be in the form of crypto-currency prices, cyber-security attacks, or news events as reported by journalists.

Table 7 Performance of the largest and the most viral conversation generated by different models

Domain	Model	Largest Size (APE)	Most viral Virality (APE)
Crypto	Baseline (recent-replay)	62	17
	Baseline (random)	**10**	83
	Lumbreras model	113	**0**
	Genetic-LSTM (our solution)	69	8
Cyber	Baseline (recent-replay)	**34**	**21**
	Baseline (random)	121	147
	Lumbreras model	358	53
	Genetic-LSTM (our solution)	87	47

We highlight the lowest APE values in bold

Our probabilistic model does not account these external events on generating the conversation structure, thus it is unable to reproduce the properties of the most viral conversation. We plan to incorporate external events on modeling conversations in future work.

In conclusion, while our solution more accurately traces both the distribution of conversation sizes and that of conversation viralities than any of the baselines, it struggles with the end points of the spectrum: very small and very large conversation properties. However, we can conclude that we generate a pool of conversations that are closer to forecasted activity than simply representing the past through random sampling, because in all metrics we consistently outperform the random and recent-replay baselines. The challenges posed by the two baseline models extracted from training data are evident also in comparison with the performance of the Lumbreras model: only once does the Lumbreras model outperforms both baselines in terms of JS distances (Table 6). In terms of APE values (as presented in Table 6), it competes closely with the baselines.

5.4 Evaluation of temporal conversations

We compare the reconstructed pool of conversations with the ground truth data in different temporal measurements. We compare (i) the size of the conversation pool as measured in the overall number of comments generated to the seed posts, and (ii) the number of distinct users who participate in the conversation pool over time. We report Dynamic Time Warping (dtw) and Root Mean Square Error (rmse) on these measurements between the conversations in the reconstructed pool and the conversations in the ground truth. We use daily granularity to bin the timeseries for comparison, and group these timeseries into five time intervals of 1–5 days, 5–7 days, 7–14 days, 14–21 days, and 21–28 days for a deeper evaluation.

Table 8 shows the APE values for the number of messages and the number of distinct users after comparing different models with the ground truth. Our simulations result in better estimations of the total number of messages than any of the baselines, with 25.3 and 8.5 absolute percentage error (APE) in the two datasets,

Table 8 Performance of the total number of messages and unique users in the conversation pools generated by different models

	Model	# Messages (APE)	# Users (APE)
Crypto	Baseline (recent-replay)	52	29
	Baseline (random)	50	**22**
	Lumbreras model	37	–
	Genetic-LSTM (our solution)	**25**	36
Cyber	Baseline (recent-replay)	29	**2**
	Baseline (random)	58	27
	Lumbreras model	11	–
	Genetic-LSTM (our solution)	**8**	67

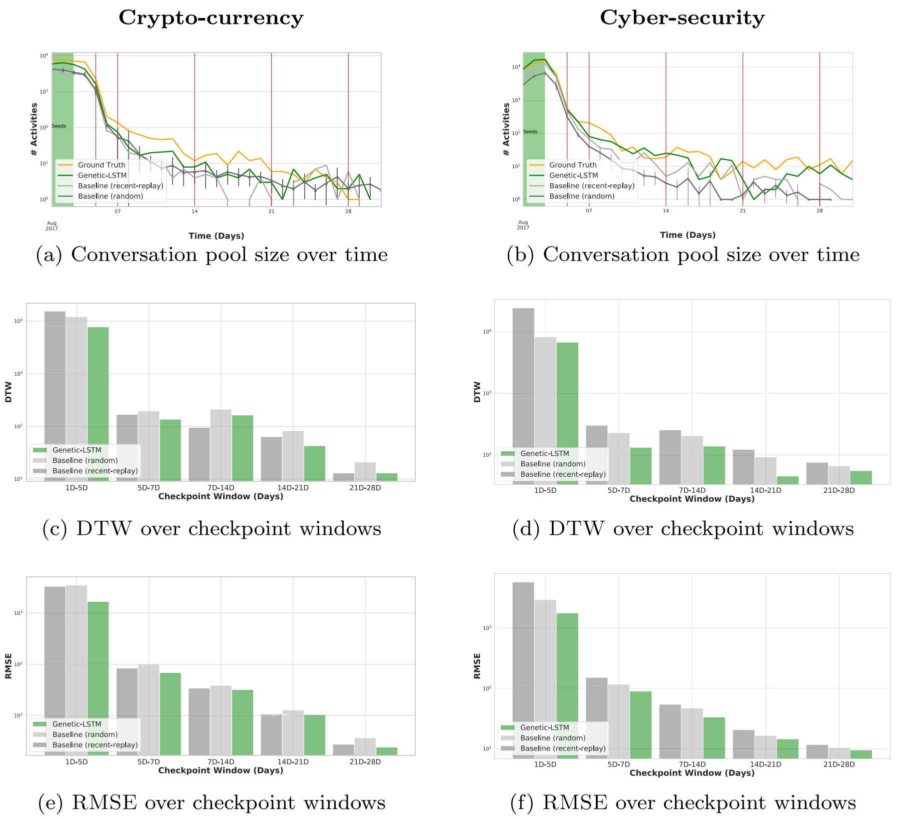
We do not report the number of distinct users for the Lumbreras Model as it does not predict user assignments. We highlight the lowest APE values in bold

which leads to 35–50% performance gain over the best-performing baseline. However, our solution does not achieve the lowest APE on the total number of distinct users as we over-predict the number of users who participate in these conversations.

We are interested, however, in evaluating our predictions over the simulated time. This is particularly relevant for application scenarios such as designing intervention techniques, when one would like to investigate "what if" scenarios and their consequences at particular times. Figures 5 and 6 report the volume of comments and the number of distinct users who participate in these conversations. There are multiple observations to be made from these plots. First, the trend of number of messages and distinct users over time holds for our simulations and for the baselines. This is because all models capture the intuitive phenomenon of high activity and user involvement when a post is freshly made, and the decay in interest as time passes.

Second, our solution fares better than the baselines not only in the aggregate number of messages at the end of the simulation period, but also over time: the green lines in Fig. 5a and b are generally the closest to the ground truth plots in yellow. As shown in Fig. 5e, our solution records a rmse value of 1685 compared to the rmse values of 3697 and 3329 for the two baseline models on predicting the conversation pool size during the first five days (1D-5D). During the next time intervals, our solution records 2–39% performance benefit in rmse values over both datasets compared to the best performed baseline solution (as shown in Fig. 5e and f).

Third, our performance advantage over the baselines is higher in the cyber-security conversations, where our solution is always better than both baselines in both rmse and dtw measurements for all interval periods shown in Fig. 5d, and f. This is probably due to the significantly larger dataset in cyber security which is 10× larger than crypto-currency dataset. A larger dataset generally helps our machine learning models to train and make better predictions. In general, our improved performance over baselines is likely due to incorporating original post information in generating the conversations, and optimizing branching factor and propagation delay in the



(a) Conversation pool size over time

(b) Conversation pool size over time

(c) DTW over checkpoint windows

(d) DTW over checkpoint windows

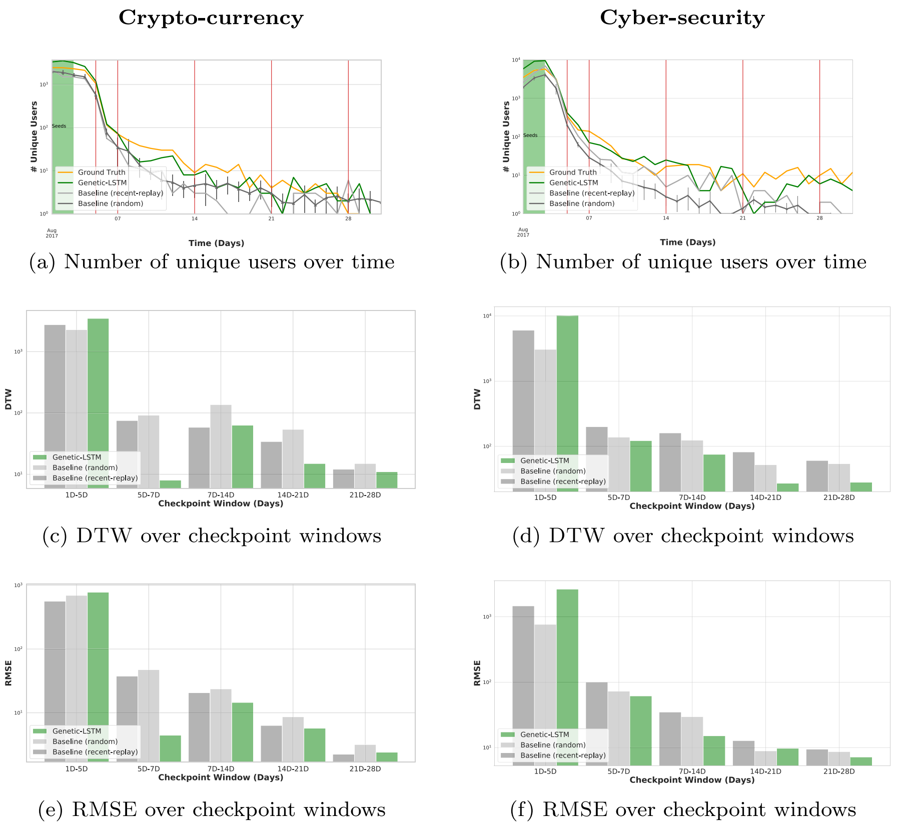
(e) RMSE over checkpoint windows

(f) RMSE over checkpoint windows

Fig. 5 The size of the conversation pool by the number of comments received over time for crypto-currency (**a**) and cyber-security (**b**) discussions. Genetic-LSTM (our solution) is compared with two competing baseline models, Baseline (recent-replay) and Baseline (random). Baseline (random) predictions are normalized over 10 different runs, and the error bars are reported for the standard deviation. The performance is reported over two quantitative metrics, (**c**, **d**) dynamic time warping (DTW) (lower is better), (**e**, **f**) RMSE values (lower is better) after comparing each model predictions with the ground truth over different time intervals

predicted pool of conversations. The baseline models do not account for such attributes but only replay the past events.

And finally, our model performs better than the baselines also in the number of users engaged over time in these conversations. For Reddit-like conversations this is a challenge since discussions may lead to provocative, offensive or menacing comments that end up repeatedly involving a sub-group of users (Medvedev et al. 2019). For example, there are only 6818 users who participate in 32,533 comments in crypto-currency discussions. In the largest conversation, the ratio between the number of comments and the number of users is 2.35 in the ground truth, and 2.06 in our solution. Our model tends to over-predict the number of users engaged short time after the seed messages are posted (as shown in Fig. 6e and f for the interval 1D–5D), and consistently performs well for the more distant future. As shown in



(a) Number of unique users over time

(b) Number of unique users over time

(c) DTW over checkpoint windows

(d) DTW over checkpoint windows

(e) RMSE over checkpoint windows

(f) RMSE over checkpoint windows

Fig. 6 The number of unique users participate in the conversation pool over time for crypto-currency (**a**) and cyber-security (**b**) discussions. The performance is reported over two quantitative metrics, (**c**, **d**) dynamic time warping (DTW) (lower is better), (**e**, **f**) RMSE values (lower is better) after comparing each model predictions with the ground truth over different time intervals

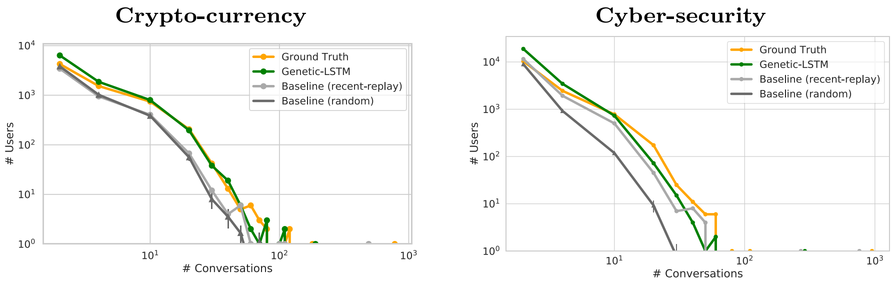
Fig. 6e–f, our solution achieves the lowest DTW and RMSE values for the interval 5D–7D across two datasets, respectively. This is particularly relevant, because it shows our model's predictive power for longer-term simulations: from the 6th to the 28th day of the simulation period, our model consistently predicts better the number of users and the timing of their comments.

5.5 Evaluation of collective behavior

Another important characteristic related to user engagement is the co-engagement with various topics. Specifically, empirical studies (DiResta et al. 2018) have shown coordinated campaigns run as troll farms or cyborgs, where groups of users engage in multiple related conversations to shift the opinion of the general audience. Accurately predicting the group of highly engaged users is important for developing intervention techniques to control information or manipulation spread and to accurately gauge the community opinion.

We report two measurements to capture the collective behavior of users who participate in these conversations. First, we present the number of users engaged in multiple conversations (as shown in Fig. 7a and b). Specifically, we record the number of conversations that a user engaged with, and count the users who engaged with X number of conversations. We noticed a heavy-tailed distribution, where few users engage in many conversations. We calculate the JS divergence between each models' distribution and the ground truth distribution. Lower JS divergence values reflect predictions closer to the number of actively engaged users observed in ground truth. Our solution achieves the lowest JSD value of 0.05 (crypto) and 0.07 (cyber) after compared with the respective baseline models. We also predict the number of highly active users closer to the ground truth value than any other baseline solution. In the crypto-currency discussions, we predict 1916 users who engage with more than two conversations, while there are 2438 such users in the ground truth and 1310 such users in the best-performing baseline solution. Our relative success is due to implicitly accounting for simultaneous conversations with possibly common users in our modeling of the problem as a pool of conversations. Specifically, our LSTM-based model that helps selecting the best pool of conversations accounts for user participation in multiple conversations, thus is able to predict better the number of highly engaged users than a model that simply repeats the past.

Second, we evaluate whether users participate in these conversations as a group according to a metric (*collectivity*) proposed by Lu et al. (2018). We record user participation in conversations in a vector $[c_1, c_2, ..., c_n]$, where c_i indicates a binary value to reflect the user involvement in the ith conversation. For this metric, we only consider the most active users who participate in at least three conversations (on average, a user participates in two conversations in the ground truth dataset). The original paper (Lu et al. 2018) used the Pearson correlation coefficient to compare all pairs of binary vectors. The higher the correlation coefficient values, two users participate in the same set of conversations. They also used Jaccard coefficient to compare the overlap of conversations between two users. According to their experiments, the Pearson correlation coefficient and Jaccard coefficient values



(a) User Engagement (b) User Engagement

Fig. 7 The number of users who engaged with X number of conversations is shown Figures **a** and **b** compared with the baseline models. The values in the *y*-axis are binned by the intervals of 10 in the *x*-axis

are correlated. While we do not experiment with any other similarity metric (e.g., Hamming distance), we believe they would result in distributions similar to what observed using Pearson correlation coefficient or Jaccard coefficient. In this work, we use the Pearson correlation coefficient to quantify the collective behavior of user involvements.

We calculate the JS-divergence and RMSE between the coefficient distributions of the simulation and the ground truth data (as shown in the Table 9). Lower JS-divergence values reflect collective behavior closer to that measured from ground truth. We achieve the lowest 0.07 and 0.12 JS-divergence values, and lowest 1815 and 976 rmse values for the respective domains after compared with the respective baseline models.

In summary, our experimental results show that, in addition to accurately predicting the structural properties of individual conversations, predicting pools of conversations also leads to more accurate predictions of user involvement over time.

6 Summary and discussions

This paper introduces a generative technique for predicting a group of simultaneous conversations in social media. Our solution uses a probabilistic generative model with the support of a genetic algorithm and LSTM neural networks. We tested our technique on two topic-based collections of Reddit conversation trees. Given a set of posts in a continuous time interval, our solution generates the full set of reactions to each message, including reactions to reactions, without having access to, for example, intermediate states of the conversation tree. In addition to generating the structure of conversation trees, our solution also assigns authorship and timing information to each message. The code for this framework is available publicly (Horawalavithana 2021).

Table 9 A comparison of the collectivity scores of users who participate in multiple conversations

	Model	Collectivity	
		JSD	RMSE
Crypto	Baseline (recent-replay)	0.09	8036
	Baseline (random)	0.14	8210
	Lumbreras model	–	–
	Genetic-LSTM (our solution)	**0.07**	**1815**
Cyber	Baseline (recent-replay)	**0.12**	1779
	Baseline (random)	0.23	3049
	Lumbreras Model	–	–
	Genetic-LSTM (our solution)	**0.12**	**976**

We show JS-divergence (JSD) and RMSE values after comparing each models' distributions of collectivity scores with the ground truth values. We do not report the number of these measurements for the Lumbreras Model as it does not predict user assignments. We highlight the lowest JSD and RMSE values in bold

Our solution captures the relationship between different microscopic conversation properties including the structure, propagation speed (timing), and the users who participate in a set of simultaneous conversations. We trained two LSTM models *on pools of conversations* to capture this relationship. In the first model, we predict whether a node in the conversation is branching (thus, generating more reactions) or is a leaf in the conversation tree. The second model classifies messages by the delay which they are posted in response to their parent. Both models use structural, user and content features in the temporal space. While structural and content level features represent the characteristics of individual conversations, the user-level features capture the characteristics of users who participate in simultaneous conversations. In the genetic algorithm, we assess the likelihood of a user action in a conversation based on the output of these two machine-learning models. Experimental results show that this technique can generate accurate conversation topological structures over time, and can accurately predict the volume of messages and the engagement of users over time.

Our solution can be applied to study "what if" scenarios in an operational setup. For example, what response would be generated if a particular post is made by a particular user account? That is, how large of a reaction would that generate in terms of messages and user engagement over time? What if that same message is posted by a different user? (say, a government organization vs. a bot account?). What if the same message is posted when there are related conversations going on, or when the conversations at the time include disjoint user groups? Answers to such questions can inform intervention strategies such as for messaging in health-related campaigns (Zarocostas 2020) or for injecting factual information in an attempt to limit disinformation (Jahanbakhsh et al. 2021). For example, Jahanbakhsh et al. (2021) suggest that predicting the engagement that users receive at posting time can help reduce the likelihood of sharing false information. In another example, a group of users can loosely coordinate to promote misleading content (Starbird et al. 2019; Aliapoulios et al. 2021). Aliapoulios et al. (2021) show that the high volume of discussions related to QAnon conspiracy theory was mainly due to a coordination between a small number of users on Reddit. Accurately simulating the growth of conversations would reveal the potential damage these users could cause in the propagation of such information.

We show the effectiveness of our approach on two groups of highly related communities: nine subreddits focused on crypto-currencies and 38 subreddits focused on cyber-security topics. The prediction of user involvement over different simultaneous conversations can also be used by community organizers to control the focused discussions, or to promote positive community norms.

Our solution has a number of limitations. One is that in evaluating the generated conversation trees, our model arbitrarily maps the content-level features from a distribution built from training data. In an ideal scenario, we should predict the attributes of the comments (e.g., polarity, subjectivity) to draw these features accurately. Moreover, a rich set of content-level features to capture humour, adversity, emotions, etc. could be developed to improve the machine-learning models. Another limitation is that our approach tends to repeat in prediction the user interactions seen in the training data. A better approach would use information about the users who have been exposed to a message and thus may be candidates for responding. However, this true diffusion structure is hidden and inferring it is difficult (Gomez-Rodriguez et al. 2016).

Our data-driven solution could be applied to other online platforms. This framework can further be extended in different ways. For example, we plan to incorporate external events such as artificial inflation of cryptocurrency prices through deception as a way to better predict unusual conversation structures. Further, the genealogical inception of subcommunities (e.g., how new subreddit communities emerge from older ones (Tan 2018)) can also be considered in generative models.

Acknowledgements This work is supported by the DARPA SocialSim Program and the Air Force Research Laboratory under contract FA8650-18-C-7825. The authors would like to thank Leidos for providing data.

Funding This work is supported by the DARPA SocialSim Program and the Air Force Research Laboratory under contract FA8650-18-C-7825.

References

Abdelzaher T, Han J, Hao Y, Jing A, Liu D, Liu S, Nguyen HH, Nicol DM, Shao H, Wang T et al (2020) Multiscale online media simulation with socialcube. Comput Math Organ Theory 26:145–174 (2020). https://doi.org/10.1007/s10588-019-09303-7

Aliapoulios M, Papasavva A, Ballard C, De Cristofaro E, Stringhini G, Zannettou S, Blackburn J (2021) The gospel according to q: understanding the qanon conspiracy from the perspective of canonical information. https://arXiv.org/210108750

Aragón P, Gómez V, García D, Kaltenbrunner A (2017a) Generative models of online discussion threads: state of the art and research challenges. J Internet Serv Appl 8(1):15

Aragón P, Gómez V, Kaltenbrunner A (2017b) To thread or not to thread: the impact of conversation threading on online discussion. In: Proceedings of the International AAAI Conference on Web and Social Media, vol 11, no 1

Bollenbacher J, Pacheco D, Hui PM, Ahn YY, Flammini A, Menczer F (2021) On the challenges of predicting microscopic dynamics of online conversations. Appl Netw Sci 6(1):1–21

Bourigault S, Lamprier S, Gallinari P (2016) Representation learning for information diffusion through social networks: an embedded cascade model. In: Proceedings of the 9th ACM International Conference on Web Search and Data Mining, ACM, pp 573–582

Chen L, Deng H (2020) Predicting user retweeting behavior in social networks with a novel ensemble learning approach. IEEE Access 8:148250–148263

Cheng J, Adamic L, Dow PA, Kleinberg JM, Leskovec J (2014) Can cascades be predicted? In: Proceedings of the 23rd international conference on World wide web, ACM, pp 925–936

Cheng J, Adamic LA, Kleinberg JM, Leskovec J (2016) Do cascades recur? In: Proceedings of the 25th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, pp 671–681

Cheng J, Kleinberg J, Leskovec J, Liben-Nowell D, State B, Subbian K, Adamic L (2018) Do diffusion protocols govern cascade growth? In: Proceedings of the International AAAI Conference on Web and Social Media, vol 12, no 1

Chollet F et al (2015) Keras. https://keras.io

DARPA DARPA (2021) Computational simulation of online social behavior (socialsim). https://www.darpa.mil/program/computational-simulation-of-online-social-behavior

De Jong K (1990) Genetic-algorithm-based learning. In: Machine learning, pp 611–638. Morgan Kaufmann

DiResta R, Shaffer K, Ruppel B, Sullivan D, Matney R, Fox R, Albright J, Johnson B (2018) The tactics & tropes of the internet research agency. New Knowledge

Dutta S, Masud S, Chakrabarti S, Chakraborty T (2020) Deep exogenous and endogenous influence combination for social chatter intensity prediction. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 1999–2008

Fang H, Cheng H, Ostendorf M (2016) Learning latent local conversation modes for predicting comment endorsement in online discussions. In: Proceedings of The 4th International Workshop on Natural Language Processing for Social Media, Association for Computational Linguistics, Austin, TX, USA, pp 55–64. https://doi.org/10.18653/v1/W16-6209

Gao X, Cao Z, Li S, Yao B, Chen G, Tang S (2019) Taxonomy and evaluation for microblog popularity prediction. ACM Trans Knowl Discov Data (TKDD) 13(2):1–40

Garibay I, Oghaz TA, Yousefi N, Mutlu EC, Schiappa M, Scheinert S, Anagnostopoulos GC, Bouwens C, Fiore SM, Mantzaris A et al (2020) Deep agent: studying the dynamics of information spread and evolution in social networks. https://arXiv.org/200311611

Glenski M, Saldanha E, Volkova S (2019) Characterizing speed and scale of cryptocurrency discussion spread on reddit. In: The World Wide Web Conference, pp 560–570

Goel S, Anderson A, Hofman J, Watts DJ (2015) The structural virality of online diffusion. Manag Sci 62(1):180–196

Gomez-Rodriguez M, Song L, Daneshmand H, Schölkopf B (2016) Estimating diffusion networks: recovery conditions, sample complexity & soft-thresholding algorithm. J Mach Learn Res 17(1):3092–3120

Gómez V, Kappen HJ, Litvak N, Kaltenbrunner A (2013) A likelihood-based framework for the analysis of discussion threads. World Wide Web 16(5–6):645–675

He X, Song G, Chen W, Jiang Q (2012) Influence blocking maximization in social networks under the competitive linear threshold model. In: Proceedings of the 2012 SIAM International Conference on Data Mining, SIAM, pp 463–474

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural comput 9(8):1735–1780

Horawalavithana S (2021) Mcas. https://github.com/SamTube405/MCAS

Hutto CJ, Gilbert E (2014) Vader: a parsimonious rule-based model for sentiment analysis of social media text. In: 8th international AAAI conference on weblogs and social media

Islam MR, Muthiah S, Adhikari B, Prakash BA, Ramakrishnan N (2018) Deepdiffuse: predicting the’who’and’when’in cascades. In: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, pp 1055–1060

Jahanbakhsh F, Zhang AX, Berinsky AJ, Pennycook G, Rand DG, Karger DR (2021) Exploring lightweight interventions at posting time to reduce the sharing of misinformation on social media. In: Proceedings of the ACM on Human-Computer Interaction 5, no. CSCW: 1–42

Krishnan S, Butler P, Tandon R, Leskovec J, Ramakrishnan N (2016) Seeing the forest for the trees: new approaches to forecasting cascades. In: Proceedings of the 8th ACM conference on web science, pp 249–258

Krohn R, Weninger T (2019) Modelling online comment threads from their start. In: IEEE international conference on big data (Big Data), pp 820–829

Kumar R, Mahdian M, McGlohon M (2010) Dynamics of conversations. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining, ACM, pp 553–562

Li C, Ma J, Guo X, Mei Q (2017) Deepcas: an end-to-end predictor of information cascades. In: Proceedings of the 26th international conference on World Wide Web, International World Wide Web Conferences Steering Committee, pp 577–586

Liben-Nowell D, Kleinberg J (2008) Tracing information flow on a global scale using internet chain-letter data. Proc Natl Acad Sci 105(12):4633–4638

Ling C, Tong G, Chen M (2020) Nestpp: modeling thread dynamics in online discussion forums. In: Proceedings of the 31st ACM conference on hypertext and social media, pp 251–260

Lu W, Chen W, Lakshmanan LV (2015) From competition to complementarity: comparative influence diffusion and maximization. Proc VLDB Endowment 9(2):60–71

Lu Y, Yu L, Zhang T, Zang C, Cui P, Song C, Zhu W (2018) Collective human behavior in cascading system: discovery, modeling and applications. In: IEEE international conference on data mining (ICDM), IEEE, pp 297–306

Lumbreras A (2016) Automatic role detection in online forums. PhD thesis Université de Lyon

Manco G, Pirrò G, Ritacco E (2018) Predicting temporal activation patterns via recurrent neural networks. In: International symposium on methodologies for intelligent systems, Springer, pp 347–356

Medvedev AN, Delvenne JC, Lambiotte R (2018) Modelling structure and predicting dynamics of discussion threads in online boards. J Complex Netw 7(1):67–82

Medvedev AN, Lambiotte R, Delvenne JC (2019) The anatomy of reddit: an overview of academic research. In: Ghanbarnejad F, Saha Roy R, Karimi F, Delvenne JC, Mitra B (eds) Dynamics on and of complex networks III. Springer International Publishing, Cham, pp 183–204

Myers SA, Leskovec J (2012) Clash of the contagions: cooperation and competition in information diffusion. In: Data mining (ICDM), IEEE 12th International Conference on, IEEE, pp 539–548

Myers SA, Zhu C, Leskovec J (2012) Information diffusion and external influence in networks. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 33–41

Qiu J, Tang J, Ma H, Dong Y, Wang K, Tang J (2018) Deepinf: social influence prediction with deep learning. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, ACM, pp 2110–2119

Singer P, Flöck F, Meinhart C, Zeitfogel E, Strohmaier M (2014) Evolution of reddit: from the front page of the internet to a self-referential community? In: Proceedings of the 23rd international conference on world wide web, ACM, pp 517–522

Starbird K, Arif A, Wilson T (2019) Disinformation as collaborative work: surfacing the participatory nature of strategic information operations. In: Proceedings of the ACM on Human-Computer Interaction, vol 3 (CSCW), pp 1–26. https://doi.org/10.1145/3359229

Tan C (2018) Tracing community genealogy: how new communities emerge from the old. In: 12th international AAAI conference on web and social media

Valera I, Gomez-Rodriguez M (2015) Modeling adoption and usage of competing products. In: Proceedings of the IEEE international conference on data mining (ICDM), IEEE Computer Society, Washington, DC, USA, ICDM '15, pp 409–418. https://doi.org/10.1109/ICDM.2015.40

Wang C, Ye M, Huberman BA (2012) From user comments to on-line conversations. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 244–252

Wang J, Zheng VW, Liu Z, Chang KCC (2017, November) Topological recurrent neural network for diffusion prediction. In: 2017 IEEE International Conference on Data Mining (ICDM). IEEE, pp 475–484

Weng L, Flammini A, Vespignani A, Menczer F (2012) Competition among memes in a world with limited attention. Sci Rep 2:335

Xiao Y, Zhang L, Li Q, Liu L (2019) Mm-sis: model for multiple information spreading in multiplex network. Phys A: Statist Mech Appl 513:135–146

Yu L, Cui P, Wang F, Song C, Yang S (2015, November) From micro to macro: uncovering and predicting information cascading process with behavioral dynamics. In: 2015 IEEE International Conference on Data Mining. IEEE, pp 559–568

Zarezade A, Khodadadi A, Farajtabar M, Rabiee HR, Zha H (2017) Correlated cascades: compete or cooperate. In: Proceedings of the 31st AAAI conference on artificial intelligence, San Francisco, California, USA, pp 238–244. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14360

Zarocostas J (2020) How to fight an infodemic. Lancet 395(10225):676

Zayats V, Ostendorf M (2018) Conversation modeling on reddit using a graph-structured lstm. Trans Assoc Comput Linguist 6:121–132

Zhao Q, Erdogdu MA, He HY, Rajaraman A, Leskovec J (2015) Seismic: a self-exciting point process model for predicting tweet popularity. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, ACM, pp 1513–1522

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Sameera Horawalavithana received his Bachelor's degree in Computer Science from University of Colombo, School of Computing, Sri Lanka. He is currently a doctoral student in the Department of Computer Science and Engineering, University of South Florida. He was also a visiting scholar at the Umea University, Sweden in 2013. His research interests include computational sociology, social dynamics and distributed systems.

Nazim Choudhury received his M.S. degree from the University of Technology, Sydney (UTS), and his Ph.D. in computer science from The University of Sydney in 2018. Before joining the department of computer science and engineering at the University of South Florida as a postdoctoral fellow, he used to work as an adjunct academic at Central Queensland University and Federation University, Australia. He also worked as a software developer and technical solution specialist at the Australian telecommunication carrier, Telstra.

John Skvoretz is a Distinguished University Professor of Sociology from the University of South Florida and Emeritus Carolina Distinguished Professor of Sociology from the University of South Carolina, is a Fellow of the American Association for the Advancement of Science and recipient of the 2012 James S. Coleman Distinguished Career Award from the Mathematical Sociology Section of the American Sociological Association. His research interests include the development and application of formal theoretical methods, such as simulation, stochastic processes, and statistical modeling and analysis, to a variety of sociological questions.

Adriana Iamnitchi is a Professor in the Department of Computer Science and Engineering at University of South Florida. She is the recipient of the National Science Foundation CAREER Award, published over 60 articles in distributed systems and computational sociology, and has served in numerous conference program committees. She has been associate editor of IEEE Transactions of Parallel and Distributed Systems and Springer Online Social Networks and Media