Social Network Analysis and Mining (2024) 14:127
https://doi.org/10.1007/s13278-024-01284-z

ORIGINAL ARTICLE

Evaluating collective action theory-based model to simulate mobs

Samer Al-khateeb[1] · Jack Burright[1] · Nitin Agarwal[2]

Received: 12 November 2023 / Revised: 16 June 2024 / Accepted: 17 June 2024

Abstract
A mob is an event that is organized via social media, email, SMS, or other forms of digital communication technologies in which a group of people (who might have an agenda) get together online or offline to collectively conduct an act and then disperse (quickly or over a long period). In recent years, these events are increasingly happening worldwide due to the anonymity of the internet, affordability of social media, boredom, etc. Studying such a phenomenon is difficult due to a lack of data, theoretical underpinning, and resources. In this research, we use the Agent-Based Modeling (ABM) technique to model the mobbers and the Monte Carlo method to assign random values to the factors extracted from the theory of Collective Action and conduct many simulations. We also leverage our previous research on Deviant Cyber Flash Mobs to implement various scenarios the mobber could face when they decide to act in a mob or not. This resulted in a model that can simulate mobs, estimate the mob success rate, and the needed powerful actors (e.g., mob organizers) for a mob to succeed. We finally evaluate our model using real-world mob data collected from the Meetup social media platform. This research is one step toward fully understanding mob formation and the motivations of its participants and organizers.

Keywords Collective action · Mob · Flash mob · Meetup · Computer simulation · Agent-based modeling · Monte Carlo method · Pearson correlation coefficient · Spearman correlation coefficient

1 Introduction

A "mob" is an event that is organized via social media, email, SMS, or other forms of digital communication technologies in which a group of people (who might have an agenda) get together online or offline to collectively conduct an act and then disperse (quickly or over a long period) (Al-khateeb and Agarwal 2021; Al-khateeb et al. 2021). To an outsider, such an event may seem arbitrary. However, a sophisticated amount of coordination is involved. In recent years, mobs *"have taken a darker twist as criminals exploit the anonymity of crowds, using social networking to coordinate everything from robberies to fights to general chaos"* (Tucker and Watkins 2011; Steinblatt 2011). The term *"mob"* has been increasingly used to remark an electronically orchestrated violence such as the January 6, 2021 attack on the United States Capitol in Washington, D.C that led to property damages, government disruption, and injuries or death for some of the protesters (Staff 2021; Barry et al. 2021). In the same month and year, an army of small investors from all over the world used Reddit to coordinate *"flashmob investing"* (Pratley 2021) to create a stock market frenzy causing GameStop's stock value to rise from $20 to $483 in less than a month (Brignall 2021). More recently, a series of brazen flash mob-style robberies of various stores, such as the Nordstrom store in Walnut Creek, CA (Barnard 2021); the 7-Eleven gas station in Los Angeles, CA (Arreola and Lloyd 2022); and the Nordstrom store in Los Angeles, CA (Wehner 2023), have caused significant financial losses. These events show that our systems (security, financial, etc.) are not equipped to handle such highly coordinated and sometimes flash actions, underscoring the importance of systematically studying such phenomena. However, mobs are not always deviant (i.e., illegal and involve violence),

✉ Samer Al-khateeb
SamerAl-khateeb1@creighton.edu

Jack Burright
JackBurright@creighton.edu

Nitin Agarwal
nxagarwal@ualr.edu

1 Department of Computer Science, Design and Journalism, Creighton University, 2500 California Plaza, Omaha, NE 68178, USA

2 COSMOS Research Center, UA-Little Rock, 2801 S University Ave, Little Rock, AR 72204, USA

Published online: 02 July 2024

for example, the "Unexpected Flash Mob Audition Shocks Simon Cowell" on February 23, 2022 (see https://blog.dancevision.com/the-best-flash-mobs-ever, for a list of videos). A study by Al-khateeb and Agarwal (2021) analyzed more than 33 research articles that used the term "flash mob" in various contexts such as cultural studies, marketing, biomedical research, etc. They found that this term was used to mean various things depending on the discipline used. So, they created a typology for the various types of flash mobs based on distinctive features such as agenda, synchronicity, and the existence of violent acts. They also proposed using the term "Mob" as the root term and defined it based on the common properties that all these types of events share such as time, environment, motive, and type of participants.

One way to study mobs is to collect data on various mobs and document various mobbers' behaviors, shared orientations, etc. Subsequently, machine learning algorithms can be employed to predict the outcome or occurrence of a mob. However, as straightforward as this task may seem, it is not. Identifying a set of mobs and gathering data about it is a complex task due to the possibility of a mob being coordinated on various platforms (e.g., multiple social media sites), data collection restrictions, lack of critical data, privacy issues (e.g., can not access emails and private messages), etc. Additionally, this method may work for a specific type of mob coordinated on that particular platform.

Fortunately, "Event-Based Social Network" (EBSN) such as Meetup.com hosts a wide range of events, which makes it crucial and well-suited for studying various events in general, and mobs in particular. Meetup.com focus on bringing like-minded people together (Grundke et al. 2023). Meetup is different from many other social media sites such as Facebook in how members develop their connections (Weinberg and Williams 2006). On Facebook, users take their offline connections and then connect with them online. Meetup is the opposite; users can join groups and connect with people online, and then meet them face-to-face. Another example of a platform that has this same interaction would be the dating site Match.com, where users match online and then can go on dates in person (Weinberg and Williams 2006). However, training a machine learning algorithm on Meetup data will only be good for predicting the results of mobs coordinated using Meetup.com. So, another way to study mobs that is more efficient, cost-effective, and still requires theoretical underpinnings, is to use computer simulation. Simulation can provide a generic way to infer the result of a mob using social science theories. Furthermore, obtaining critical data about the mob is not an easy task. So, in our simulation model, we aim to minimize the amount of data input to the model. This is because we assume that when there is a mob, there is much unknown information about it. Thus, we expect users to provide minimal information and rely on the model to conduct simulations and report results. However, since we have information about mobs collected from Meetup.com, we can use this data for model validation. Specifically, we can test the performance of the simulation model in predicting the outcomes of known mobs on Meetup.

Simulation solved many real-world problems by modeling real-world processes to provide otherwise unobtainable information. Computer simulation has been used to *"predict the weather, design aircraft, create special effects for movies"* (Zelle 2004) among others. Many simulations require events to occur with a certain likelihood. These types of simulations are called Monte Carlo simulations because the results depend on *"chance"* probabilities. The Monte Carlo method (or simulation) is named after the Monaco resort town known for its gambling and casinos. It was invented by Stanislaw Ulam, a Polish-American scientist, in the late 1940s when he was working on the Manhattan Project (Kenton 2020). The Monte Carlo method has myriad applications in various fields such as business and finance, supply chain, oil and gas, science, insurance, and engineering (Palisade 2021). This method is usually used to estimate the likelihood of a certain outcome or predict the future value of a variable, risk analysis, etc.

Individuals in the mob phenomenon are purposive—those who "adjust their actions to counter variable circumstances that prevent their perceptions from matching their objective" (McPhail et al. 1992)—and the mob phenomenon is an example of organized collective action (McPhail et al. 1992). To an outsider, such an event may seem arbitrary, however, a sophisticated amount of coordination is involved. Hence, in this research, we decided to use the ABM method to understand the mob phenomenon and the mobber's behaviors as ABM simulates the actions and interactions of the agents (mobbers) to analyze their effects on the system as a whole. Each agent follows predefined rules and can interact with other agents and the environment. It is also one of the recommended methods for modeling human behavior (Duffy 2021), alongside system dynamics modeling, game theory, Monte Carlo simulation, network analysis, mathematical modeling, and machine learning. It allows us to test various parameters and compare the model data to add or rule out parameters that impact the emerging behavior (Mollona 2008) and to test various questions (Jackson et al. 2017). We also utilized the Monte Carlo Simulation because it involves using random sampling to model the behavior of agents with uncertainty.

In this research, we use computer simulation guided by constructs extracted from the theory of collective action to *implement* a theoretical model previously published[1] in

[1] The present paper is an extended version of our paper that was presented at the International Conference on Modelling and Simulation of Social-Behavioural Phenomena in Creative Societies (MSBC 2022) (Al-khateeb and Agarwal 2023). This paper adds more research

(Al-khateeb and Agarwal 2014, 2014b, 2015a, b) and analyze its performance using data collected from Meetup.com to have a better understanding of the mob phenomenon defined above. More specifically, this research tries to answer the following research questions:

RQ1: Given the following parameters (number of invited people, threshold value of the mob success, the number of simulations (mobs), the number of powerful actors), what is the chance a mob will succeed?

RQ2: Given the following parameters (number of invited people, threshold value of mob success, and the number of simulations (mobs)), how many powerful actors are needed to have a successful mob?

RQ3: Given the following parameters (number of invited people, threshold value of the mob success, the number of simulations (mobs), the number of powerful actors), how does the decision-making time of individuals invited to join a mob affect the mob participation rate?

RQ4: How can we infer the result of real-world mobs using our simulation model? Furthermore, how does the model perform when simulating real-world mobs with values for the following parameters: the number of invited people, the threshold value of the mob success (which is the mob participation rate), the number of simulations, the number of powerful actors (i.e., mob organizers), and the outcome of the real-world mob (success or failure)?

Next, we provide a brief Literature Review of the topics related to this paper. Then, we discuss the Methodology we followed to implement our model. In Sect. 4, we explain our analysis and report our findings. Sect. 5 addresses the model limitations. In Sect. 6, we conclude with possible future research directions.

2 Literature review

In this section, we provide a concise thematic literature review Quamina-Aiyejina (2022) covering the topics addressed in this paper. Given that this research focuses on the mob phenomenon; we begin by examining relevant literature on mobs. We also utilized computer simulation (employing the Monte Carlo method and Agent-Based Modeling technique), the Euler's method, and validated our model performance using data collected from Meetup.com. Hence, we also provide a brief review of these topics. The review provided in this section is by no means exhaustive; undoubtedly, there are many other important research articles that we may have overlooked or were unable to include, due to paper length constraints. However, we believe that this brief thematic review provides the reader with background knowledge on the topics and techniques employed in this article to facilitate understanding the main goal of this paper. Also, this review elucidates our rationale for selecting these topics as we highlight the similarities and differences between previous literature and our current work. In most of the literature surveyed, we found a lack of systematic and computational models that aim to study mob formation and predict mob occurrence or result (i.e., success or failure). This research is one step in this direction.

The mob phenomenon has been studied in various disciplines such as communication studies (Nicholson 2005), marketing (Barnes 2006), cultural studies (Do Vale 2010), and other disciplines (see (Al-khateeb and Agarwal 2021), for an overview).

For example, Kaulingfreks and Warren (2010) studied one specific example of flash mobbing behavior named “mobile clubbing" where individuals get together quickly in a public space and start dancing to the music they are listening to on their MP3 players or iPod's. They focused in their study on the organizational structure of these individuals and how the cities' structure can facilitate such gatherings. They also discussed the potential risks of such an organizational structure.

In her dissertation, Rebecca Walker (Walker 2011), in a non-computational way examined Bill Wasik's original eight flash mobs through multiple lenses to find out “how the flash mob might serve as a locus of community, creativity, and politics in an age overrun by spectacle and surveillance". Through a rigorous analysis of the mob's various components, she discovered the “objects of those fears, desires, and tensions: surveillance, community, space, and power".

Houston et al. (2013) studied the urban youth's perspectives on flash mobs to understand their motivation to participate, the causes and consequences they think they might face, and the possible solutions to avoid violent flash mobs. They surveyed a focus group that consisted of 50 participants from Kansas City, Missouri because it was the place of a recent violent flash mob where youth were the primary participants (or the participant's peers). They found that youth violent flash mobs are usually caused by: youth boredom and youth wanting to: gain attention, see each other, watch fights, and be visible or recognized (some assertion of identity). They also found that providing youths with more

Footnote 1 (continued)

questions, literature reviews, experiments, model validation, and new findings.

safe and fun activities, improving the various city and community services, more police presence in existing events, and trying to fix the social disorder in the community, in general, could help in mitigating youth violent flash mobs (Houston et al. 2013).

Finally, Galen et al. (2021) created an agent-based model using the Artificial Society Analytics Platform (ASAP) to study the effects of religious belief and affiliation on *Prosociality* (i.e., the values and behaviors that benefit others). They used data collected from the World Values Survey to validate their model performance. The ABM agent has a variable with the value supernatural (i.e., belief in God) or naturalist (i.e., no belief in God) worldview. Our model is similar to this one as it also uses ABM, but our agent has two main variables namely, *interest* and *control*, and is implemented using Python. Additionally, we used data collected from Meetup.com to validate the simulated mob results (simulated using the ABM) by comparing them to the mob results in the collected data.

The first method we used in this research is the Monte Carlo method. We used it to randomly assign *interest* and *control* to each agent (mobber) in the event (mob) and also to conduct many simulations. The Monte Carlo method/simulation *"is a computerized mathematical technique"* (Palisade 2021) used in decision-making, testing statistical procedure robustness (Muralidhar 2003), and quantitative analysis (Palisade 2021; Muralidhar 2003). It is used to simulate all possible outcomes of a decision to make a better decision (e.g., the one that involves the least risk) (Palisade 2021). Since its inception during World War II, it has been used in a variety of fields such as project management, research and development, manufacturing, engineering, and transportation (Palisade 2021; Kenton 2020). The Monte Carlo method uses probability distributions (e.g., normal, lognormal, uniform, triangular, pert, or discrete) as input samples. Then it outputs all possible outcomes with the likelihood of their happening (Palisade 2021). It works by assigning multiple values to uncertain variables, and then the average of the multiple results is used as an estimate of the final outcome (Kenton 2020). It is an example of stochastic modeling (Kenton 2020) which is much better and more realistic than deterministic modeling because it allows an analyst to see what input(s) has more impact on a specific outcome (Palisade 2021).

A study close to this study was conducted by Sung-Ha Hwang (Hwang 2009). He used Lanchester's equations,[2] the Monte Carlo method, and adopted the collective punishment hypothesis to develop a model of public good with punishment (cost of acting). The goal of his research was to study the *group-size* effects on collective action. In his research, the actor can be either a punisher (act), defector (act against), or cooperator (do not act). He found that *"an increase in group size always favors punishers and cooperators"* (Hwang 2009). This means that the bigger the group size (number of mobbers) with a higher number of punishers and cooperators (those who act and those who do not act), the higher the chance of achieving collective action (and thus a higher chance of mob success).

The second technique we used was Agent-Based Modeling (ABM). We used it to model the mobbers (i.e., the agents) in the simulation who will have *interest* and *control* assigned randomly using the Monte Carlo method. ABM is a bottom-up approach for studying emerging patterns from simple interactions among agents (Duffy 2021). It is applied in a plethora of disciplines such as sociology, economics, political science (Jackson et al. 2017), and applications that range from cargo routing to Artificial Intelligence (Duffy 2021). These models can be a great choice for researchers because they allow for a high level of control on the experiment; can be run an infinite number of times (on a large scale) with all reasonable values for parameters that can allow researchers to generate all possible outputs (Mollona 2008); able to model nonlinear dynamics over time; and allow researchers to test questions that otherwise would not be possible because of ethical concerns (Jackson et al. 2017). Being able to test all parameters allows researchers to compare the model data and add or rule out variables that impact the emerging behavior (Mollona 2008).

In this research, we also used the Euler method to model the mobber decision-making time, i.e., the time it takes the mobber to decide whether to act, withdraw, act against, or perform a power exchange. The Euler method is an effective and flexible general numerical technique for solving Ordinary Differential Equations (ODEs), which can model many problems in mathematics, science, and engineering. Euler's equation has applications in areas such as cell growth, fluid dynamics, quantum, optics, mechanics, and electrical engineering. There are several numerical techniques in existing literature for solving ODEs with initial value problems using Euler's method, for instance, numerical analysis of the linearly implicit Euler method with truncated Wiener process for the stochastic SIR model (Yang et al. 2023b), nonlinear infection-age SIR models (Yang et al. 2023a), deep Euler method to solve ordinary differential equations (Shen et al. 2020), a performant and feature-full agent-based modeling software of minimal code complexity (Datseris et al. 2022), Agent-based simulation from system dynamics model using the forward Euler method (Macal 2010), Extension of a Mathematical model using agent-based simulation to Zombie attack (da Costa Junior et al. 2018).

[2] These are differential equations that are used to calculate the strength of the military and were invented during WWII by Frederick W. Lanchester, an English engineer (Lanchester 1916).

Finally, to validate our simulation model, we collected data from Meetup.com. We then simulated these mobs by feeding our model with the following inputs: the number of invited people, the threshold value of the mob success (the mob participation rate), the number of simulations, and the number of powerful actors (i.e., mob organizers). Subsequently, we compared the outcomes (success or failure) of the simulated mobs with the outcomes of the real-world mobs that we collected from Meetup.com. Meetup is an EBSN that aims to allow organizers to plan events about any topic ranging from very formal business meetings to casual events like a movie night (Ricken et al. 2017). Users can create and join groups based on their interests, in addition to their ability to organize events (Huang et al. 2020).

In their study, Schneider et al. (2015) aimed to assess the potential of Meetup.com as a platform for promoting physical activity and fostering a sense of community. They divided eight neighborhoods into two groups, with one joining dog walking Meetup groups and the other participating in the American Heart Association (AHA) group. The six-month study resulted in an average increase of 919 daily steps for participants in the Meetup groups and a 427-step increase for the AHA group. Although the sense of community in the Meetup groups did not significantly change, members reported positive behavioral outcomes.

In another study, Huang et al. (2020) created a Deep Learning model for predicting the growth and success of Meetup *groups*. They utilized a Kaggle.com database containing data from 1,087,923 users, 16,330 groups, and 5,801 events. The Kaggle.com data lacked user-event linkage, so the authors collected additional data using Meetup's API, focusing on users attending events and their group memberships. The combined data set revealed that, on average, each event attracted around 15 participants, and each group hosted approximately 11 events. They also classified groups as either "successful" or "unsuccessful" based on their data analysis (Huang et al. 2020). Note that this study focused on the success and failure of the *group*, while our study focused on the success and failure of the *event* (i.e., mob).

Grundke et al. (2023) developed a web crawler to gather data from Meetup and utilized this data to conduct experiments aimed at enhancing the events recommendation page, known as the "cold-start" page. This page is presented to users when they initially join Meetup and have not yet attended any events. The authors employed five features as filters to help users discover interesting events. These features included the number of event attendees, the group's trend (indicating whether the group attendance was increasing, decreasing, or stable), expected member loyalty, the formality of the event description, and the compactness of the event description (measured in terms of relevant words relative to the description's length). Three of these features (number of attendees, trend, and expected member loyalty) relied on RSVPs, which were infrequently filled out. In contrast, the other two features (formality and compactness) were based on event descriptions that are always available and this allows for the features to be estimated. The researchers observed that their website effectively increased the number of events that users expressed interest in attending. This led them to conclude that their event-finding filter page might be more effective than the "cold-start" recommendations page (Grundke et al. 2023). We wrote a Python code that used the Meetup API to collect Meetup events data, which we used to validate our simulation model.

Finally, Howard Dean, in his 2004 Democratic Party presidential nomination campaign, used Meetup.com to arrange electronic events that would transition into in-person gatherings, referred to as electronic to face-to-face (E2F). So, Weinberg and Williams (2006) conducted a study to pursue two research questions during that period. Their first research question aimed to determine whether a significant relationship existed between Meetup events and crucial campaign indicators, such as campaign contributions, volunteering, and candidate support. Secondly, they sought to understand if Meetup attendance influenced the level of activism among attendees. The findings showed that attendees who participated in more Meetup events tended to donate more money, perceived Meetup as a valuable tool, felt more engaged in the campaign, and were more likely to vote for the candidate and encourage others to join the campaign. Consequently, the researchers concluded that Meetup and other electronic-to-face (E2F) platforms have the potential to be highly valuable for raising awareness or promoting engagement on various topics (Weinberg and Williams 2006). Our research aims to simulate both E2F events and cyber events, and it utilizes Meetup data for model validation.

3 Methodology

In this section, we provide some details about the Meetup mobs data that we collected. Then, we explain the logical/theoretical framework we previously constructed based on the sociological theory of collective action (Al-khateeb and Agarwal 2014, 2014b, 2015a, b). We then proceed to describe the various scenarios a mobber can face when it comes to their decision to act in a mob or not. Finally, we explain how the simulation model was implemented using the Python programming language.

3.1 Meetup data collection

Using Meetup.com's GraphQL API, we gathered data from 27 distinct groups, all featuring the topic "Flash Mob". For each group, we collected data from every event (or mob) they had organized. This data has 85 attributes and encompasses details such as the number of attendees, RSVP times, all comments and replies associated with the mob, images shared, URLs in comments, the event description, location of the event, time of the event, etc. This resulted in 3,536 mobs with over 18,000 RSVP's , all dated from May 3, 2009, to September 10, 2023. We stored this data in a MySQL database. Later we filtered the data by removing all events from private groups since we couldn't collect their comments. Moreover, events for which we couldn't calculate the number of individuals invited were also excluded. As a result of these filtration steps, 459 mobs (24 cyber mobs and 435 physical mobs) remained for our analytical examination. For the cyber mobs, we have 22 successful mobs and 2 failed mobs, while for the physical mobs, we have 247 successful mobs and 188 failed mobs. The 459 mobs we analyzed were organized by 16 different groups, with sizes ranging from 33 to 9,203 members. The members of these groups (i.e., mobbers) are from different parts of the world, and the majority are located in big cities such as New York, Sydney, and London. These mobs were tagged with 21 different topics such as "Singing Lessons", "Choir", "Film Festivals", "Directors", "Outdoors", "Dancing", "Gorilla Filmmaking", "Flash Mob", etc. It's important to note that a mob can be assigned more than one topic. We examined these topics and found that all analyzed mobs were benign (no deviant mobs were included), which is expected considering that the groups that organized these mobs are public. This data was primarily used to validate the simulation model-predicted class. Therefore, we only needed to utilize two attributes: *MobID* (numerical sequence) and *TrueClassForMeetupMobs* (1 or 0 indicating successful or failed mob, respectively).

3.2 The theoretical model

Collective action occurs when a group of people work together to achieve a common goal. There are various forms of collective action, such as deviant mobs (i.e., violent mobs), cyber mobs, entertaining flash mobs, protests (peaceful or violent), grassroots campaigns, social movements, etc. These actions could be organized by groups or organizations that aim to achieve a common goal (e.g., social, political, or religious), such as worker unions, interest groups, deviant groups, etc. Therefore, we have chosen the sociological theory of collective action to model mobs in our previous studies and this study. However, in this paper, we implement the previous theoretical model into a simulation model (which uses the Monte Carlo method and the Agent-based approach) that leverages the logic of collective action and utilizes the factors we extracted from the theory to understand the mob phenomenon and try to infer its result (success or failure) whether these mobs were carried out in the cyberspace, physical space, or both, i.e., the cybernetic space. We simulated many real-world mobs that were organized on Meetup.com and compared the result of the simulation model to the actual result of the mobs to test the performance of the simulation model (i.e., to answer, how well can our model infer the result of these mobs?).

Collective action can be defined as all activity of common or shared interest among two or more individuals (Olson 1965; Oliver 1993). From the logic of collective action by Olson (1965), we found that one factor that encourages mobbers to participate in a mob is the amount of utility (benefits) they will gain by participating. This is supported by Coleman's argument in his book *"there is a single action principle which governs the actions of the actors in the system: Each actor chooses those actions which maximize his utility given environmental context created by the events..."* (Coleman 2017). Also, the *utility difference* which is defined as the amount of utility (benefit) gained by an actor (a mobber) from the pair of possible outcomes (success or failure) of the same event (e.g., the mob) will determine his/her *interest* in participating in that event (mob). As the amount of *gained utility* increases, the *interest* in participating in the mob will also increase and vice versa. Another factor that Olson mentioned in his book that can affect mobbers' decision to participate is *control*, i.e., to what extent an individual could affect the outcome of the event (Olson 1965). If the mobber has an *interest* in participating in a mob and has *control* over the outcome of the mob, the mobber is considered powerful (e.g., a mob organizer). By summing the power of all mobbers, we can determine the *importance* of a mob. If the importance of a mob exceeds a certain (predetermined) *threshold* value, then we can hypothesize the mob will more likely succeed. Otherwise, the mob is more likely to fail. Here, we measure the sum of the power of all mobbers using what we called the *participation rate*. The threshold value can be estimated using various methods, e.g., based on empirical observations of known mobs or shared knowledge from law enforcement agencies, etc.

3.3 Scenarios a mobber could face

Based on the factors mentioned above (i.e., interest, control, and power) a mobber[3] could face four possible scenarios when it comes to deciding what to do when s(he) sees a mob in physical or cyberspace (or gets invited to participate in a mob). These scenarios will determine the decision of the mobber when it comes to acting in a mob or not. The assumption here is their decision will be based only on the

[3] In this paper, we use the words "mobber" and "participant" interchangeably, referring to the initial number of agents in the simulation model. In the real world, these would be the people who either see an invitation to join a mob on social media or receive an invitation to join a mob by any means or in any form.

aforementioned factors (i.e., interest, control, and power) and a mob will succeed when the mob *participation rate* exceeds a *threshold value*. The four scenarios are:

1. If an individual has interest and control, then the likelihood of the individual participation is the highest, i.e., the individual will act.
2. If an individual has interest but does not have control, then the individual may act (i.e., has a 50/50 chance of acting or withdrawing). So, the likelihood of individual participation is relatively lower than in the previous scenario.
3. If an individual does not have interest but has control, then the individual has two choices—either will withdraw, i.e., will not act, or execute power exchange (i.e., relinquish power to possibly gain control over other events (mobs) or to simply gain social capital[4]). These individuals can perfectly exchange power with mobbers of the second scenario above.
4. If an individual has no interest and no control, then the individual will have two choices—either withdraw or act against the group.

We assume powerful actors (e.g., mob organizers) will always have *interest* and *control*, so they will always act in the mob. These scenarios have been coded using Python programming language and the details of the scripts are explained in the next subsection. These scripts are used to answer the research questions mentioned in the Introduction section.

3.4 Model implementation

In this subsection, we provide more details about the model implementation by describing the model entities and variables, in addition to their possible domain values. We also explain the method we used for assigning these values. The four Python scripts explained next use the following main entities and variables:

- *Agent* is an entity that represents an individual (a mobber). It is implemented in Python as a *class* with two instance variables, namely, *interest* and *control*.
- *interest* is an instance variable of the *Agent* class. Its domain is 0, 1. This variable represents the interest of a mobber (Agent) in acting in a mob. The value is randomly assigned using Python's *randrange*(0, 2) function.
- *control* is an instance variable of the *Agent* class. Its domain is $\{0, 1\}$. This variable represents the control of a mobber (Agent) on the mob outcome. The value is randomly assigned using Python's *randrange*(0, 2) function.
- *Mob_Practitioners* is a variable. Its domain is $\{1 - \infty\}$. This variable represents the initial set of agents or "mobbers" or "participants". In the real world, these would be the people who either see an invitation to join a mob on social media or receive an invitation to join a mob by any means or in any form. This value is entered manually by the user/person who is running the model.
- *Num_Powerful_Actors* is a variable. Its domain is $\{0 - Mob\_Practitioners\}$. This number represents the number of powerful actors, i.e., those with $interest = 1$ and $control = 1$ (e.g., mob organizers). This value is entered manually by the user/person who is running the model. It will be subtracted from the *Mob_Practitioners* at the beginning of the simulation since these participants are assumed to be *acting* (so they do not need to figure out their decision) and toward the end of the simulation, these will be added to the *Act_Counter*.
- *Act_Counter* is a variable used to count the number of mobbers who decided to act based on their *interest* and *control*. Its domain is $\{0 - Mob\_Practitioners\}$. This value is determined using the scenarios explained in Subsect. 3.3. Also, the manually entered value for the *Num_Powerful_Actors* will be added to it.
- *Withdraw_Counter* is a variable used to count the number of mobbers who decided to withdraw from acting based on their *interest* and *control*. Its domain is $\{0 - Mob\_Practitioners\}$. This value is determined using the scenarios explained in Subsect. 3.3.
- *S_Withdraw_Counter* is a variable used to count the number of mobbers who decide to withdraw in the second scenario mentioned in Subsect. 3.3. Its domain is $\{0 - Mob\_Practitioners\}$. This is a special withdrawal case because if the mobber gains *control*, they will act.
- *Power_Exchange_Counter* is a variable used to count the number of mobbers willing to engage in power exchange. Its domain is $\{0 - Mob\_Practitioners\}$. These individuals are willing to power exchange with the special withdrawal case above. This value is determined using the scenarios explained in Subsect. 3.3.
- *Act_Against_Counter* is a variable used to count the number of mobbers who decided to act against the mob. Its domain is $\{0 - Mob\_Practitioners\}$. This value is determined using the scenarios explained in Subsect. 3.3.
- *Success_Threshold* is a variable used to set the threshold of participation required to consider a mob successful. Its domain is $\{0 - 1\}$. This value is entered manually by the user/person who is running the model.
- *Participation_Rate* is a variable that represents the percentage of mobbers who decided to act in a mob. It's

[4] Social capital, as stated by Pierre Bourdieu, is *"the value that one gain from personal connections such as membership in a family, an ethnic association, elite clubs, or other solidarity groups"* (Biggart 2008).

Table 1 Table showing the time it took to run each of our experiments

Experiment for question#	Condition	Execution time in seconds
RQ1	Using Eq. 1	20.91042184829712
RQ1	Using Eq. 2	20.852996826171875
RQ2	Using Eqs. 1 and 3	1172.9467859268188
RQ2	Using Eqs. 2 and 4	1168.5381560325623
RQ3	Using Eqs. 1 and 5	2.3534278869628906
RQ3	Using Eqs. 2 and 5	2.4167799949645996
RQ4	Using Eqs. 1 and 5 for Cyber mobs	39.100542068481445
RQ4	Using Eqs. 2 and 5 for Cyber mob	39.77706289291382
RQ4	Using Eqs. 1 and 5 for Physical mobs	2133.105674982071
RQ4	Using Eqs. 2 and 5 for Physical mobs	2116.772106409073

domain is $\{0 - 1\}$. This value is calculated using Eqs. 1 or 2.

- *Num_Of_Simulation* is a variable that is used to set the number of mobs to simulate (i.e., the number of simulations). It's domain is $\{1 - \infty\}$. This value is entered manually by the user/person who is running the model.
- Ni_1 is a variable used in $Mob\_Simulator - Q3.py$ to set the number of participants (mobbers or agents) picked at each "wave" to decide what to do. The numerical value of this variable will be determined using Eq. 5.

All the scripts created or used in this paper are available at https://github.com/SamerAl-khateeb/MobsSimulator-Python. To answer the four research questions written in the Introduction section we used four Python scripts to run our experiments and an additional three scripts to analyze the data (*fig2and3corr.py*, *fig4corr.py*, and *modelperformance.py*). Next, we will focus on explaining the four scripts we wrote to implement our model and run the experiments needed to answer the research questions. In all four scripts, the `randrange()` method from the `random` library was used to implement the four scenarios mentioned above. The `randrange()` method was used to give each mobber a random *interest* and *control*; then a 50/50 chance of either *acting* or *withdrawing* (as in the second scenario); to *withdraw* or *power exchange* (as in the third scenario); and to *withdraw* or *act against* (as in the fourth scenario). All simulations were carried out on a MacBook Pro with a 2.9 GHz 6-Core Intel Core i9 CPU, 16 GB 2400 MHz DDR4 RAM, and Radeon Pro 560X 4 GB + Intel UHD Graphics 630 1536 MB GPUs. We list the running/execution times of our experiments in Table 1.

By running the first script, i.e., `Mob_Simulator-Q1.py` the user will be asked to enter *the total number of invited people*, the *threshold value of the mob success*, the *number of simulations (mobs)*, and *the number of powerful actors*. Then the code will report the result of each mob simulated (success or failure), the number of mobbers who: acted (participated), withdrew, did power exchange, acted against the mob, and the overall mob participation rate. We calculate the *participation rate* using two formulas shown in Eqs. 1 and 2 below. Equation 1 takes into consideration the effect of people who *act against* the mob, while Eq. 2 does not take into consideration the effect of people who act against the mob.

$$Participation\ Rate = \frac{(Num\,Act + Num\,Powerful\,Actors - Num\,Act\,Against)}{Num\,Of\,Invited\,Mobbers} \tag{1}$$

$$Participation\ Rate = \frac{(Num\,Act + Num\,Powerful\,Actors)}{Num\,Of\,Invited\,Mobbers} \tag{2}$$

The reason we used these two equations is to examine the effect of the people who oppose the mob goal (and act against it) on the mob outcome. In the second equation, we ignore the effect of those people. So, the first case is analogous to viewing what happens during a mob as two competing events where some people act while others act against them. However, in Eq. 2, the mob can be viewed as one event represented by only those who act. In both cases, the number of individuals who act against the mob will be present. However, in Eq. 1, we account for it, while in Eq. 2, we do not account for it. Also, in both cases, the *participation rate* will be used to determine if the mob succeeded or not. The assumption is that if the *participation rate* exceeds the provided threshold value, it means the event was important enough to attract a sufficient number of people to participate. Consequently, it will be marked as a successful mob; otherwise, it will be marked as an unsuccessful (failed) mob. The script also reports aggregate results, i.e., the overall success and failure rate (out of the simulated mobs, how many succeeded? and how many failed?). Finally, the script reports *the average participation rate* of all the simulated mobs.

Running the second script file, i.e., `Mob_Simulator-Q2.py`, will prompt the user to enter *the total number of* *invited people*, the *threshold value of the mob success*, and the *number of simulations (mobs)*. The script will report the result of each mob simulated (success or failure), how many powerful mobbers were required to transform a failed mob into a successful one, the number of mobbers who acted (participated), withdrew, did power exchange, acted against the mob, and the participation rate. For the *participation rate*, we also used Eqs. 1 and 2, as shown above, for the same reasons mentioned. However, we do not account for the number of powerful actors as in the case of `Mob_Simulator-Q1.py`. This is because we are trying to estimate how many powerful actors are needed when the mob fails in both cases—that is, in competing events and as one event. To do this, we used Eqs. 3 and 4, shown below, to calculate those needed powerful actors.

$$NPA = \frac{(STho \cdot MP)}{100} - AC + AAC \tag{3}$$

$$NPA = \frac{(STho \cdot MP)}{100} - AC \tag{4}$$

Where: *NPA* is the number of required powerful actors (e.g., mob organizers), *STho* is the mob success threshold value, *MP* is the number of invited individuals, *AAC* is the number of individuals who act against the mob, and *AC* is the number of individuals who act (participate) in the mob. This script will also provide aggregate results, including the overall success and failure rates (i.e., out of the simulated mobs, how many mobs succeeded and how many failed). It will also report *the average number of powerful actors needed to make most of the failed mobs succeed*.

Using the third script, i.e., `Mob_Simulator-Q3.py`, the user will be asked to enter the same input values as in the script `Mob_Simulator-Q1.py` (explained above). However, this script will *not* assign the interests and control of all the invited people randomly *all at once*; instead, it will assign them in "waves" (analogous to real-world scenarios, i.e., when individuals receive an invitation to join a mob, they do not all decide to join or not join the mob at the same time, they take time to think then decide). We assume these "waves" correspond to periods of time during which people decide what to do (i.e., act, withdraw, exchange power, or act against the mob). The size of the "wave" is determined using the Euler method (see Eq. 5 below).

$$wave\ size = \frac{Num\ Of\ Invited\ People}{((\Delta t \cdot e) + 1)} \tag{5}$$

Where t is the time, Δt is 1, and e is the Euler number (i.e., 2.71828). The script reports the same variables as reported by `Mob_Simulator-Q1.py`. However, the purpose of creating this script is to test the effects of applying the Euler method as a model for the mobbers' decision time. This script will lead to answering the third research question mentioned in the Introduction section.

Running the fourth script file, i.e., `Mob_Simulator-Q4.py`, will read an `input.csv` file that contains the ground truth data pulled from Meetup.com. The `input.csv` file should have the following attributes: *mob ID, number of invited people, number of mob organizers, the participation rate of the real-world mob* calculated using Eqs. 6 and 7 (shown below), and *the real-world mob outcome, which will be either "Success" or "Failure"*.

$$\begin{aligned} &Num\ Of\ Invited\ People = Num\ Of\ Event\ Organizers + \\ &(Num\ Of\ Max\ Allowed\ Tickets \\ &\quad + (Num\ Of\ Max\ Allowed\ Tickets \cdot Num\ Of\ Allowed\ Guests)) \end{aligned} \tag{6}$$

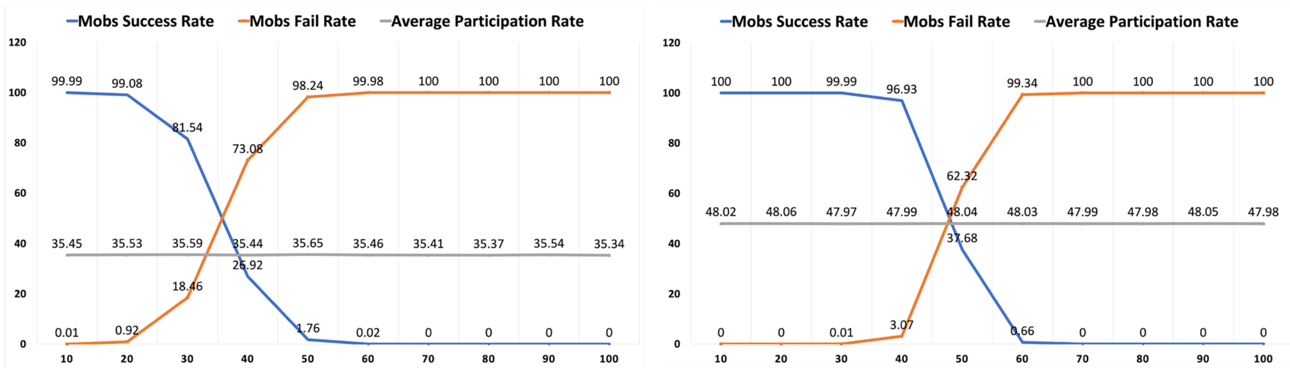
$$Participation\ Rate = \frac{Num\ Of\ Invited\ People\ Responded\ With\ Yes}{Num\ Of\ Invited\ People} \tag{7}$$

Executing this script will generate two .csv files. The first .csv file, i.e., `Q4-IndividualMobStats.csv`, contains information about each of the simulated mobs with the following features (columns): *threshold of the real-world mob, the simulated mob result, the simulated number of powerful actors, the simulated number of acting mobbers, the simulated number of mobbers who withdrew, the simulated number of mobbers who performed power exchange, the simulated number of mobbers who acted against the mob, and the participation rate for each of the simulated mobs*. The second .csv file, i.e., `Q4-OverallSimulationStats.csv`, contains overall mob statistics with the following columns: *threshold of the real-world mob, the number of mobs simulated, average simulated mob success rate, average simulated mob failure rate, average simulated mob participation rate, and the overall simulated mob result, which will be either "Success" or "Failure"*. The data that resulted from running this script was compared to the ground truth data to answer the fourth research question.

4 Analysis and results

In this section, we focus on answering the research questions listed in the Introduction section using the scripts described in the Methodology section.

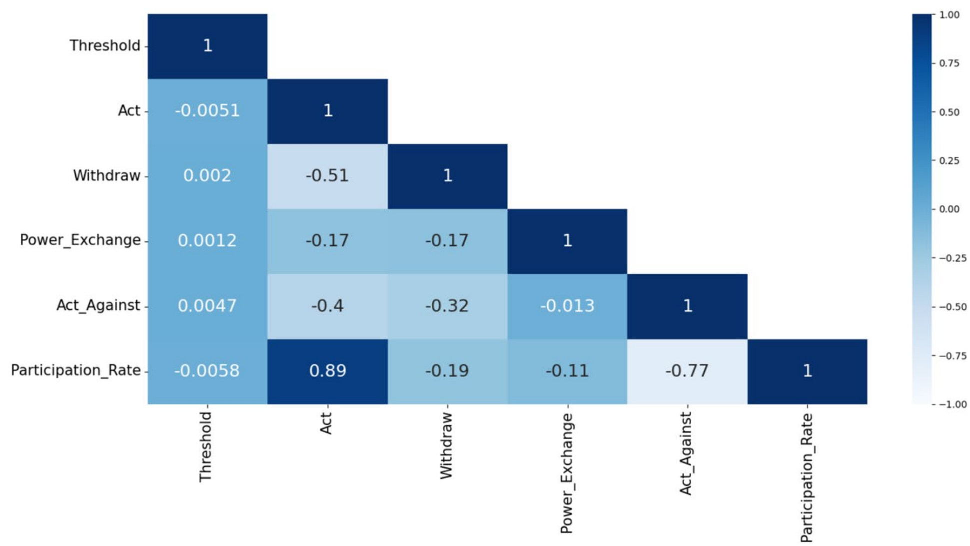
We conducted a set of experiments using `Mob_Simulator-Q1.py` to answer the first research question, i.e., RQ1: Given the following parameters (number of invited people, threshold value of the mob success, the number of simulations (mobs), the number of powerful actors), what is the chance a mob will succeed? we set the *number of invited people* to 100 in all the experiments. We also set the *number of powerful actors* to 0 to simulate the case when we do not



(a) (b)

Fig. 1 Change in the average participation rate, mob success rate, and mob failure rate as the threshold changes from 10 to 100 (the x-axis). As the threshold increased, the mob success rate decreased while the mob fail rate increased. **a** shows the average participation rate using the first equation, which is 35.5%, while **b** shows the average participation rate using the second equation, which is 48%

Fig. 2 Correlation between the various mobber types when we used Eq. 1 to calculate the participation rate



know the number of powerful actors. Finally, we set the *number of simulations* for each case to 10,000 simulations (mobs). The *goal* here is to estimate the *average participation rate* that can be resulted from running the model without knowing the number of powerful actors. Knowing this rate will help in estimating the threshold value that we can use to determine the success or failure of a mob.

We found that the *mob success rate* depends inversely on the *threshold* provided (i.e., if the *threshold* increases the success rate decrease) with *Pearson Correlation Coefficient* (PCC) of − 0.86 when we used Eq. 1 to calculate the participation rate and − 0.90 when we used Eq. 2 to calculate the *participation rate*.

We also found that the *average participation rate* regardless of the provided threshold is around 35.5% when we used Eq. 1 and 48% when we used Eq. 2. This means that, under the current model, if a mob has less than a 35.5% (or 48%) threshold, it will most likely succeed. However, if the threshold value is more than 35.5% (or 48%), the mob will mostly fail (see Fig. 1).

Finally, we found a positive correlation between the *participation rate* and the number of mobbers who *act* ($PCC = 0.89$ using Eq. 1 OR $PCC = 1.0$ using Eq. 2, $p < 0.001$). Also, we found a negative correlation between the *participation rate* and the number of mobbers who: act against ($PCC = -0.77$ OR $PCC = -0.4$, $p < 0.001$); *withdraw* ($PCC = -0.19$ OR $PCC = -0.51$, $p < 0.001$); and *power exchange* ($PCC = -0.11$ OR $PCC = -0.18$, $p < 0.001$). Overall, the mobbers who act against the mob seem to have more negative effects on mob success than mobbers who withdrew or did power exchange (see Fig. 2).

To answer the second research question, i.e., RQ2: Given the following parameters (number of invited people, threshold value of mob success, and the number of

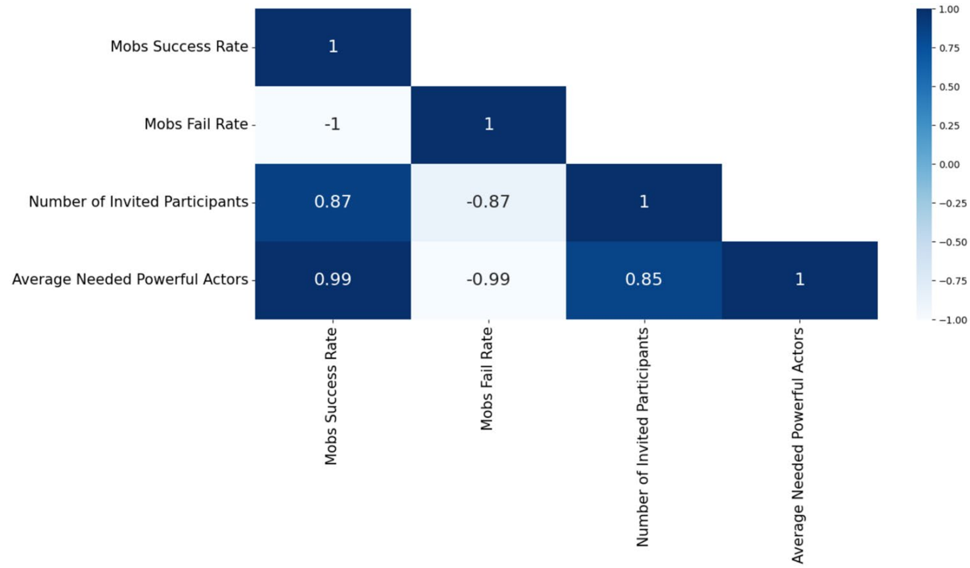


Fig. 3 Correlation between the mobs' success rate, mobs fail rate, number of invited mobbers, and the average needed powerful actors, when we used Eq. 1 to calculate the participation rate

simulations (mobs)), how many powerful actors are needed to have a successful mob? we used `Mob_Simulator-Q2.py` to conduct another set of experiments. The average *threshold* value the model was able to produce without knowing the *number of powerful actors* was around 35.5% (or 48% using Eq. 2), so we used these two values for the next set of experiments (simulated 10,000 mobs for each case) to answer the second research question. We varied the *number of invited people* from 10 to 100, then from 100 to 1000, and finally from 1000 to 10,000 to study the effect of *crowd size* on the success (or failure) rate of the mob. This should also help to find the relationship between the *number of invited people* and *the needed powerful actors*, e.g., for a given mob with a specific number of invited people, how many organizers do we need to make the mob succeed?

We found that the average number of *needed powerful actors* is positively correlated ($PCC = 0.85$ using Eq. 1 OR $PCC = 0.21$ using Eq. 2, $p < 0.001$) with the *number of invited people* which means the bigger the *crowd size* the *more* powerful actors we need to make a mob succeed. Also, when we ignore the people who might act against the mob (i.e., using Eq.2), the correlation decreases, which means we do not need that many powerful actors (or organizers) to make the mob succeed. In other words, less powerful actors (organizers) are needed when no people act against the mob.

Finally, we found that in both cases (counting or ignoring the participants acting against the mob), there is a positive correlation ($PCC = 0.87$ using Eq .1 OR $PCC = 0.79$ using Eq. 2, $p < 0.001$) between the number of invited people and the mob success rate (see Fig. 3). This means as more people are invited to a mob, the chance of participation increases, which also increases the chance of having a successful mob. This finding aligns with the findings of Hwang (2009) who stated that larger group sizes favor punishers (in our case, those who participate in the mob).

To answer the third research question, i.e., RQ3: Given the following parameters (number of invited people, threshold value of the mob success, the number of simulations (mobs), the number of powerful actors), how does the decision-making time of individuals invited to join a mob affect the mob participation rate? we used `Mob_Simulator-Q3.py` to run an experiment where we set the *number of invited people* to 10,000. We also set the *number of powerful actors* to 0 to simulate the case when we do not know the number of powerful actors. Finally, we set the *number of simulations* for each case (i.e., using Eqs. 1 and 2) to 100 simulations, resulting in 200 mobs. The goal of this experiment is to estimate the average participation rate that can result from running the model without knowing the number of powerful actors and to compare it to the result obtained from running `Mob_Simulator-Q1.py`, which *did not* use the Euler method.

We found that the average participation rate, regardless of the provided threshold value, is around 37.40% (i.e., higher than 35.5% without the Euler method) when we used Eq. 1 to calculate the participation rate and 49.76% (i.e., higher than 48% without the Euler method) when we used Eq. 2. This shows that when people are given time to think about what to do, the overall participation rate increases, i.e., more people will participate.

To validate this claim, we calculated Spearman's correlation coefficient (SCC) to determine the relationship between the 459 Meetup.com mobs participation rate and the average mobbers' time to respond with yes values. We found a strong, positive monotonic correlation (i.e., a high *SCC* value) between the mob participation rate (calculated using Eqs. 6 and 7) and the average mobbers' time to say

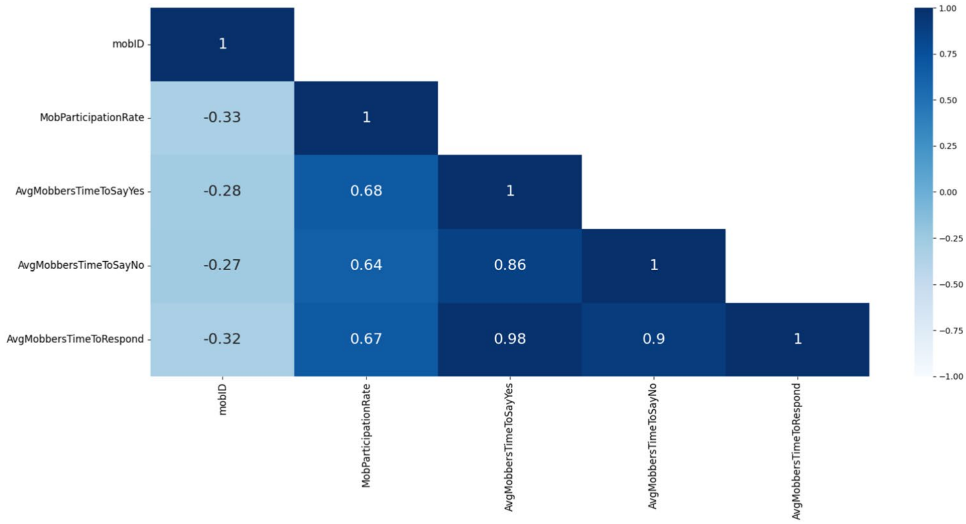


Fig. 4 The Spearman correlation coefficient values (SCC) between the mobs' participation rate and the average mobber's time to respond with a yes, no, or simply a yes or no, measured in minutes

yes ($SCC = 0.68$, $n = 459$, $p < 0.001$), the average mobbers' time to say no ($SCC = 0.64$, $n = 459$, $p < 0.001$), and the average mobbers' time to respond with either yes or no ($SCC = 0.67$, $n = 459$, $p < 0.001$), see Fig. 4. We used the Pearson correlation to measure the correlations between the variables related to RQ1 and RQ2 because there was a linear relationship between these variables. This was confirmed via the coefficient scores and p-values, which were all found to be less than 0.001. We also used the Spearman correlation to measure the correlations between the variables related to RQ3 because there is a monotonic relationship between these variables. This was also confirmed via the coefficient scores and $p - values < 0.001$. Positive monotonic correlation means that as one variable's values increase, the values of the other variable also tend to increase. It doesn't mean that the increase is constant; it only means that higher values of one variable are associated with higher values of the other, even if the relationship is curved or uneven Weir (2023).

To answer the fourth research question, i.e., RQ4: How can we infer the result of real-world mobs using our simulation model? Furthermore, how does the model perform when simulating real-world mobs with values for the following parameters: the number of invited people, the threshold value of the mob success (which is the mob participation rate), the number of simulations, the number of powerful actors (i.e., mob organizers), and the outcome of the real-world mob (success or failure)? we used `Mob_Simulator-Q4.py` to run an experiment where we simulated each of the real-world mobs 100 times, resulting in 45,900 mobs. We simulated both the cyber mobs (24 mobs) and the physical mobs (435 mobs) that we identified on Meetup.com. Additionally, we used Eqs. 1 and 2 to simulate the mobs because we lacked information on whether individuals were opposing these mobs (such data is not available on Meetup.com). We used the result of the simulated mobs in the second output file, i.e., `Q4-OverallSimulationStats.csv`, and the result of the ground truth data to evaluate the model performance. In other words, since we have the result of each mob in the ground-truth data (succeeded or failed), and our model can infer the result of each mob and store the inference result in the Q4-OverallSimulationStats.csv file, evaluating the model's performance in this case, became like evaluating the performance of a binary classifier. Hence, we used Accuracy, Precision, Recall, and F1-Score to answer the fourth research question. We used the accuracy_score, precision_score, recall_score, and f1_score functions from the scikit-learn Python library[5] to calculate the values of these metrics.

Accuracy measures the number of times the model can correctly detect the positive and negative classes. *"It's computed by the sum of True Positives and True Negatives divided by the total population"* (Jankay 2018). Precision focuses on *"the success probability of making a correct positive class classification. It's computed as the number of True Positives divided by the total number of positive calls"* (Jankay 2018) and Recall shows *"how sensitive the model is towards identifying the positive class. It's computed as the number of True Positives divided by the sum of True Positives and False Negatives"* (Jankay 2018). A high *accuracy* means the model is good at distinguishing between a successful mob and a failed mob. High *precision* indicates that the model can capture most of the positive classes (successful mob), i.e., a low false positive, while a high *recall* value means a low false negative.

[5] Available at: https://scikit-learn.org/stable/modules/classes.html#sklearn-metrics-metrics

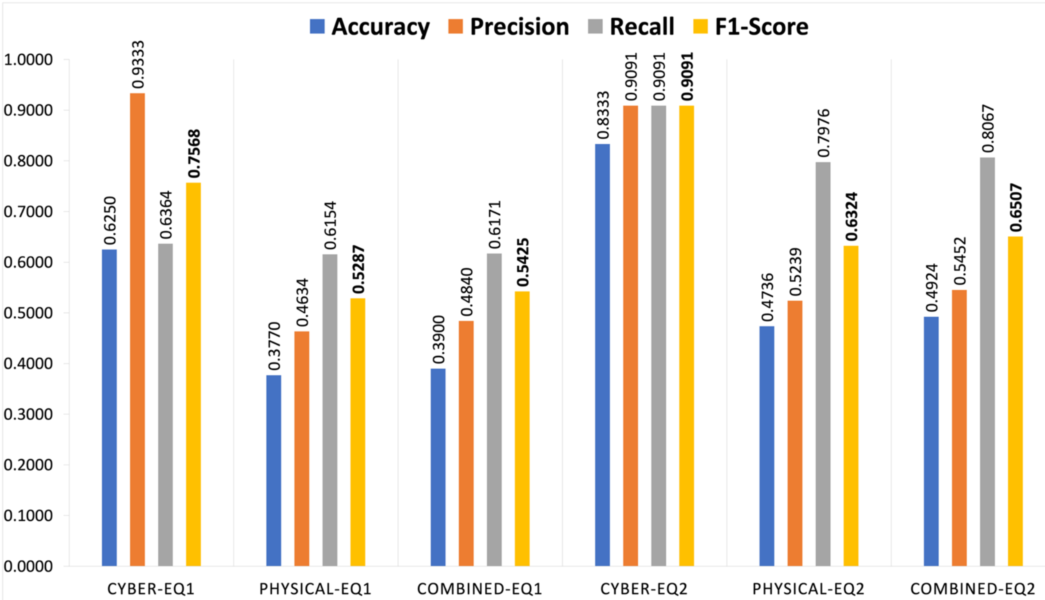


Fig. 5 Model performance is measured in terms of Accuracy, Precision, Recall, and F1-Score

While *accuracy* is simple to measure and understand, it does not provide a good performance measure when we have imbalanced data. So the harmonic mean of precision and recall (i.e., F-1 Score) can provide a better understanding of the model performance as *"it takes into account the type of errors - false positive and false negative - and not just the number of predictions that were incorrect"* (Sharma 2023). Our ground-truth data is imbalanced, i.e., we have more successful mobs (269) than failed mobs (190), so the F1-Score will give the best performance metric for our simulation model.

We found that our model can infer cyber mobs results (62.50% and 83.33%) with higher accuracy than physical mobs results (37.70% and 47.36%), and this is confirmed by the F1-score also (0.7568 and 0.9091 F1-Score) versus (0.5287 and 0.6324 F1-Score). Figure 5 shows the model performance for the cyber mobs, physical mobs, and combined. It also illustrates the performance of the model when Eqs. 1 or 2 is used. We also found that if the model uses Eq. 2, we obtain better inference results than using Eq. 1, suggesting that there was no opposition to the mobs in any of the instances of the ground-truth data, especially since all the Meetup mobs we collected were benign, i.e., not malicious or deviant mobs that involve violence or illegal acts.

5 Limitation

The theoretically supported model we've developed can infer the success (or failure) of real-world mobs and the number of organizers needed for a real-world mob to achieve success, with a good F1-score, especially when Eq. 2 was used (≈ 0.91). Most of the factors used in the theoretical model have been studied in the literature but have never been used computationally to simulate and study mobs.

The model still needs improvements, which can be achieved by considering other factors such as the existence of social ties between the mobbers, the location of the events, and other social science theories such as the Diffusion of Innovation Theory, Lifestyle-Routine Activity Theory, etc. All these factors and theories can significantly affect the mobber's decision to act/participate (or not) in a mob.

However, building a more accurate model requires interdisciplinary knowledge and collaboration between psychologists, sociologists, computer scientists, and others. This model serves as a proof of concept that factors from the theories of social science can be extracted and computationally used to simulate real-world mobs or behaviors to understand their effects on society and be prepared when things do not go as expected.

6 Conclusion and future research direction

In this study, we simulated around 806,100 mobs using the Monte Carlo method and the ABM technique guided by constructs (the agent decision rules) extracted from the theory of Collective Action. We also validated the model by comparing the simulated mobs' results to the ground-truth data we collected from Meetup.com. The goal is to build a theoretical model that can help us better understand the mob phenomenon. We found that the mob success rate, using either Eqs. 1 or 2, has a strong negative correlation with the threshold value. Additionally, we observed that the average participation rate, regardless of the provided threshold, is around 35.5% when using Eq. 1 and 48% when using Eq. 2. This implies that when individuals act against the mob, a lower participation should be anticipated. The participation rate is positively correlated with the number of mobbers who act and negatively correlated with those who act against, withdraw, or engage in power exchange. This suggests that as more people actively participate, the likelihood of the mob succeeding increases. Moreover, the number of required powerful actors shows a positive correlation with the number of invited people using either Eqs. 1 or 2. This implies that a larger crowd size necessitates more powerful actors to ensure mob success. Notably, this correlation is weaker when using Eq. 2, highlighting the impact of individuals who can act against the mob. In other words, if there is minimal opposition to the mob, fewer powerful actors (mob organizers) are needed for success, and vice versa.

We also observed that, in both cases, the higher the number of people invited to the mob, the greater the likelihood of success (i.e., a strong positive correlation exists between the number of invited people and the mob success rate). Additionally, when we employ the Euler method to model the time it takes for an invited person to decide what to do, the average participation rate slightly increases. This observation suggests that witnessing others joining the mob might influence the decision of potential participants, encouraging them to join and resulting in a higher overall participation rate. Lastly, we found that the model is more adept at inferring cyber mobs than physical mobs and mobs without opposition.

As stated in the limitations section, this theoretically supported model offers valuable insights into the mob and the mobbers' behaviors, serving as a proof of concept. However, future refinement of the model is necessary, as additional factors could be incorporated, and more theories could be applied. Currently, the model is implemented using the Python programming language; however, developing a web-based tool for the model would better cater to non-technical users and could be a potential avenue for future work. Finally, McPhail et al. (1992) stated that purposive individuals in the same gathering can generate similar reference signals, leading to various forms of collective action with varying levels of complexity: (1) *Independently*: no communication between individuals when they decide to act or not, (2) *Interdependently*: individuals communicate with other individuals to figure out what to do, and (3) *Voluntarily* or *Obediently*: individuals communicate with their bosses (e.g., powerful actors) and do what they are asked to do, i.e., to act the same way as the powerful actors, i.e., act. Our model is analogous to the first form mentioned above, i.e., *Independently*: after individuals are invited to participate in the mob, they decide what to do based on their interest and control. However, in real-world cases, many people change their mind as the mob progress. For example, some of the invited people to a mob might decide not to act at the beginning, then once they see many others participating, they change their mind and decide to act (i.e., follow the herd) and vice versa. So, one possible future research direction is to investigate the dynamic nature of such a phenomenon, i.e., giving participants the freedom to change their minds, *Interdependently*, *Voluntarily*, or *Obediently*.

Acknowledgements This research is funded in part by the U.S. National Science Foundation (OIA-1946391, OIA-1920920, IIS-1636933, ACI-1429160, and IIS-1110868), U.S. Office of the Under Secretary of Defense for Research and Engineering (FA9550-22-1-0332), U.S. Army Research Office (W911NF-20-1-0262, W911NF-16-1-0189, W911NF-23-1-0011, W911NF-24-1-0078), U.S. Office of Naval Research (N00014-10-1-0091, N00014-14-1-0489, N00014-15-P-1187, N00014-16-1-2016, N00014-16-1-2412, N00014-17-1-2675, N00014-17-1-2605, N68335-19-C-0359, N00014-19-1-2336, N68335-20-C-0540, N00014-21-1-2121, N00014-21-1-2765, N00014-22-1-2318), U.S. Air Force Research Laboratory, U.S. Defense Advanced Research Projects Agency (W31P4Q-17-C-0059), Arkansas Research Alliance, the Jerry L. Maulden/Entergy Endowment at the University of Arkansas at Little Rock, and the Australian Department of Defense Strategic Policy Grants Program (SPGP) (award number: 2020–106-094). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations. The researchers gratefully acknowledge the support. In addition to the generous support by the aforementioned organizations, we appreciate the valuable suggestion to use the Euler method in experiment 3 by Ahmed Al-Taweel at the COSMOS Research Center.

Declarations

Ethical approval In conducting our research on (MEvaluating Collective Action Theory-based Model to Simulate obs), we prioritize the fundamental principle of user privacy. All collected data undergoes a meticulous anonymization process to adhere to the highest standards of data protection laws, ensuring strict compliance with social media platform policies. Our commitment to privacy is unwavering, and we guarantee that none of our results could potentially disclose the identity of any account in the datasets, aligning with both ethical standards and platform guidelines. Our study aims to illuminate mob behavior while concurrently upholding the right to free speech in accordance with social media platforms' content policies. The proposed methodology has been meticulously designed to identify such behavior, providing controlled intervention options for network service providers and policymakers. We are conscientious about minimizing errors to prevent unwarranted censorship and maintaining alignment with social media platforms' policy framework. Rooted in ethical guidelines for scientific research, our investigation is conducted with honesty, transparency, and careful ethical supervision. We believe that our work holds the potential for positive social impact and is dedicated to maintaining the highest ethical standards throughout the research process.

References

Al-khateeb S, Agarwal N (2014) Developing a conceptual framework for modeling deviant cyber flash mob: a socio-computational approach leveraging hypergraph constructs. J Digit Forensics Secur Law 9(2):10

Al-khateeb S, Agarwal N (2014b) Modeling flash mobs in cybernetic space: evaluating threats of emerging socio-technical behaviors to human security. In: 2014 IEEE joint intelligence and security informatics conference. IEEE, pp 328–328

Al-khateeb S, Agarwal N (2015a) Analyzing deviant cyber flash mobs of isil on twitter. In: International conference on social computing, behavioral-cultural modeling, and prediction. Springer, pp 251–257

Al-khateeb S, Agarwal N (2015b) Analyzing flash mobs in cybernetic space and the imminent security threats a collective action based theoretical perspective on emerging sociotechnical behaviors. In: 2015 AAAI Spring symposium series

Al-khateeb S, Agarwal N (2021) Flash mob: a multidisciplinary review. Soc Netw Anal Min 11(1):97

Al-khateeb S, Agarwal N (2023) Studying simulated mobs using monte Carlo method and the theory of collective action. In: Agarwal N, Kleiner GB, Sakalauskas L (eds) Modeling and simulation of social-behavioral phenomena in creative societies. Springer Nature Switzerland, Cham, pp 3–13

Al-khateeb S, Anderson M, Agarwal N (2021) Studying the role of social bots during cyber flash mobs. In: Social, cultural, and behavioral modeling: 14th international conference, SBP-BRiMS 2021, Virtual Event, 6–9 July 2021, Proceedings 14. Springer

Arreola A, Lloyd J (2022) Watch: street takeover 'flash mob' swarms lOS Angeles 7-eleven. https://www.nbclosangeles.com/news/local/watch-unruly-street-takeover-crowd-swarms-los-angeles-7-eleven/2967174/

Barnard C (2021) Brazen flash mob-style robbery of walnut creek nordstrom sparks outrage from city leaders. https://abc7news.com/walnut-creek-nordstrom-looting-robbery-smash-and-grab-wc-broadway-plaza/11260247/, section: crime-safety

Barnes NG (2006) Mob it and sell it: creating marketing opportunity through the replication of flash mobs. Mark Manag J 16(1)

Barry D, McIntire M, Rosenberg M (2021) 'our president wants us here': the mob that stormed the capitol. The New York Times https://www.nytimes.com/2021/01/09/us/capitol-rioters.html

Biggart NW (2008) Readings in economic sociology. John Wiley & Sons, Hoboken

Brignall M (2021) How GameStop traders fired the first shots in millennials' war on wall street. http://www.theguardian.com/business/2021/jan/30/how-gamestop-traders-fired-the-first-shots-in-millennials-war-on-wall-street, section: Business

Coleman JS (2017) Mathematics of collective action. Transaction Publishers, New Jersey

da Costa Junior MJZ, Rodrigues BC, Adamatti DF (2018) Extension of a mathematical model using agent-based simulation to zombie attack. Rev Eletron Argent Bras Tecnol Inf Comun. https://doi.org/10.5281/zenodo.1228489

Datseris G, Vahdati AR, DuBois TC (2022) Agents. jl: a performant and feature-full agent-based modeling software of minimal code complexity. Sage Publication, Sage UK, London, England

Do Vale S (2010) Trash mob: Zombie walks and the positivity of monsters in western popular culture. In: The domination of fear. Brill Rodopi, pp 191–202

Duffy VG (2021) Digital human modeling and applications in health, safety, ergonomics and risk management. In: Human body, motion and behavior, lecture notes in computer science, vol 12777. Springer International Publishing, Cham, https://doi.org/10.1007/978-3-030-77817-0, https://link.springer.com/10.1007/978-3-030-77817-0

Galen L, Gore R, Shults FL (2021) Modeling the effects of religious belief and affiliation on prosociality. Secularism Nonreligion 10

Grundke M, Jung J, Sachse JP, et al (2023) Meetup miner:measuring event interestingness on meetup. https://www.parnswir.de/blobs/smm_meetup.pdf

Houston JB, Seo H, Knight LAT et al (2013) Urban youth's perspectives on flash mobs. J Appl Commun Res 41(3):236–252

Huang T, Liang Y, Sun Z (2020) Predicting group success in meetup with graphs. https://snap.stanford.edu/class/cs224w-2019/project/26422233.pdf

Hwang SH (2009) Larger groups may alleviate collective action problems. http://hdl.handle.net/10419/64195

Jackson JC, Rand D, Lewis K et al (2017) Agent-based modeling: a guide for social psychologists. Soc Psychol Pers Sci 8(4):387–395. https://doi.org/10.1177/1948550617691100

Jankay Y (2018) The 3 pillars of binary classification: accuracy, precision & recall. https://medium.com/@yashwant140393/the-3-pillars-of-binary-classification-accuracy-precision-recall-d2da3d09f664

Kaulingfreks R, Warren S (2010) Swarm: flash mobs, mobile clubbing and the city. Cult Organ 16(3):211–227

Kenton W (2020) Monte Carlo simulation. https://www.investopedia.com/terms/m/montecarlosimulation.asp

Lanchester FW (1916) Aircraft in warfare: the dawn of the fourth arm. Constable limited

Macal CM (2010) To agent-based simulation from system dynamics. In: Proceedings of the 2010 winter simulation conference. IEEE, pp 371–382

McPhail C, Powers WT, Tucker CW (1992) Simulating individual and collective action in temporary gatherings. Soc Sci Comput Rev 10(1):1–28

Mollona E (2008) Computer simulation in social sciences. J Manage Gov 12(2):205–211. https://doi.org/10.1007/s10997-008-9049-6

Muralidhar K (2003) Monte Carlo simulation. https://doi.org/10.1016/B0-12-227240-4/00114-3

Nicholson JA (2005) Flash! mobs in the age of mobile connectivity. 6, Fibreculture Publications

Oliver PE (1993) Formal models of collective action. Ann Rev Sociol 19(1):271–300

Olson M (1965) Logic of collective action: public goods and the theory of groups. Harvard University Press, Cambridge

Palisade (2021) What is monte Carlo simulation? https://www.palisade.com/risk/monte_carlo_simulation.asp

Pratley N (2021) The reddit flash mob won't be able to work the GameStop magic on silver. http://www.theguardian.com/business/nils-pratley-on-finance/2021/feb/01/reddits-flash-mob-wont-be-able-to-work-the-gamestop-magic-on-silver, section: Business

Quamina-Aiyejina L (2022) The literature review: 5. organizing the literature review. https://libguides.uwi.edu/c.php?g=11323 &p=59143

Ricken S, Barkhuus L, Jones Q (2017) Going online to meet offline: organizational practices of social activities through meetup. In: Proceedings of the 8th international conference on communities and technologies, pp 139–148

Schneider KL, Murphy D, Ferrara C et al (2015) An online social network to increase walking in dog owners: a randomized trial. Med Sci Sports Exerc 47(3):631

Sharma N (2023) Understanding and applying f1 score: a deep dive with hands-on coding. https://arize.com/blog-course/f1-score/
Shen X, Cheng X, Liang K (2020) Deep euler method: solving ODEs by approximating the local truncation error of the Euler method. arXiv preprint arXiv:2003.09573
Staff WP (2021) Woman dies after shooting in U.S. capitol; D.C. national guard activated after mob breaches building. Washington Post https://www.washingtonpost.com/dc-md-va/2021/01/06/dc-protests-trump-rally-live-updates/
Steinblatt H (2011) E-incitement: a framework for regulating the incitement of criminal flash mobs. Fordham Intell Prop Media Ent LJ 22:753
Tucker E, Watkins T (2011) More flash mobs gather with criminal intent. https://www.nbcnews.com/id/wbna44077826
Walker R (2011) Eight is not enough: a historical, cultural, and philosophical analysis of the flash mob. Louisiana State University and Agricultural & Mechanical College
Wehner G (2023) California nordstrom ransacked in huge 'flash mob' smash-and-grab, \$60k to \$100k in goods taken: police | fox news. https://www.foxnews.com/us/california-nordstrom-ransacked-flash-mob-smash-grab-goods-taken
Weinberg BD, Williams CB (2006) The 2004 us presidential campaign: impact of hybrid offline and online 'meetup' communities. J Direct Data Digit Mark Pract 8:46–57
Weir I (2023) Spearman's rank correlation- introduction. https://www.statstutor.ac.uk/resources/uploaded/spearmans.pdf
Yang H, Yang Z, Liu S (2023a) Numerical threshold of linearly implicit Euler method for nonlinear infection age sir models. Discrete Contin Dyn Syst Ser B 28(1)
Yang X, Yang Z, Zhang C (2023) Numerical analysis of the linearly implicit Euler method with truncated wiener process for the stochastic SIR model. Mathematics and computers in simulation. Elsevier, Amsterdam
Zelle JM (2004) Python programming: an introduction to computer science, 3rd edn. Franklin, Beedle & Associates Inc, Wilsonville