Recent Approaches of Temporal Social Network Prediction: A Survey

Fred Mubang
Department of Computer Science & Engineering
University of South Florida
Tampa, FL, USA
fmubang@mail.usf.edu

Lawrence O. Hall
Department of Computer Science & Engineering
University of South Florida
Tampa, FL, USA
lohall@usf.edu

***Abstract*—Social media has become a ubiquitous part of everyday life. As a result, many researchers have taken an interest in predicting future activity on social media networks. This work is a survey of recent temporal prediction approaches for social media networks. We focus on 2 areas in particular: (1) Github, an online collaborative platform, and (2) political discussions on various social media platforms.**

***Index Terms*—temporal predictions, social networks, machine learning, agent-based learning, prediction, simulation**

I. Introduction

Social media is heavily ingrained in today's society. It has many purposes such as communicating with friends and family, sharing opinions, and spreading information. It stands to reason that it would be ideal to create models that could accurately predict activity on various social media platforms. Such models could help with a variety of issues, such as tracking the spread of misinformation, tracking polarizing narratives, and aiding with cybersecurity.

In this work recent developments of social media prediction frameworks will be discussed. Two main domains will be focused on in this work: (1) social media prediction in Github, an online collaborative platform, and (2) social media prediction of various platforms within the realm of political discussion. These 2 domains were chosen due to their widespread growth in recent years. This review can possibly serve as a guide to researchers who are new to the area of social network activity prediction across time.

This paper is organized in the following manner. Section II contains the motivation for this work. Section III contains the definitions for the terms used throughout this work. Section IV contains discussion regarding the challenges of temporal social network prediction. Sections V and VI contain discussion of the different models covered in our literature review for Github and political discussions, respectively. In Section VII model results are discussed. Section VIII discusses future work and lastly IX contains an overall summary and closing thoughts.

II. Motivation

In this section, the motivations behind predicting activity on Github and political activity prediction on social media platforms are discussed.

Identify applicable funding agency here. If none, delete this.

A. Github Background and Motivation

Recently, much interest has arisen on predicting future phenomenon within the Github domain. Github is a website in which software developers can upload programs in *code repositories*, or *code repos*, for short. These repos can be shared. As of the time of this writing, Github is used by over 65 million developers, 3 million organizations, and has over 200 million code repositories. Futhermore, 72% of the Fortune 50 companies utilize Github for their software development [1].

Since Github is so widely used, if one can predict future user and repository activity, one can gain a better understanding of technological advances. This information can be useful for companies trying to learn how to better service their customers with software that is relevant to their needs.

Furthermore, predicting Github activity can aid with cybersecurity maintenance. In 2016, Black Duck's Center for Open Source Research and Innovation (COSRI) analyzed more than 1,000 applications that were audited as part of merger-and-acquisition transactions [2]. The audit analysis found that 96% of these applications contained open-source software, and more than 60% of those applications contained known open-source security vulnerabilities [2]. If an organization could predict that repositories containing vulnerable open-source code will receive a higher-than-normal amount of activity, that could alert an organization to the fact that their systems utilizing software from those repositories could be the target of hackers. The organization could then take preemptive measures to assure any private data they have will be safe. The authors of [3] found that software vulnerabilities are mentioned on Reddit and Twitter, and that this information can be used to predict repository activity on Github. So, it is possible to use data-driven models to avoid or prevent software vulnerability exploitation.

Such foresight may have been useful to Equifax, a credit bureau, who in 2017 was the victim of a data breach in which the social security numbers of 143 million Americans were put at risk [2]. The hackers were able to attack Equifax due to a software vulnerability in Apache Struts, an open-source framework (on Github) for creating web applications in Java. The specific identifier for this vulnerability is *CVE-2017-5638*

[2].

B. Political Discussion Motivation

The world of politics has become strongly intertwined with social media. Recent research has also shown that various bad actors exploit social media in order to carry out disinformation campaigns. For example, the authors of [4]–[6] found evidence of misinformation campaigns against the White Helmets of Syria on various social media platforms.

In [7], the authors analyzed a Twitter network of users discussing the 2020 Election Fraud Claims. Through analyzing only tweets that contained URLs, they found that there is polarization in this network. In other words they were able to detect 2 distinct communities - those who believed in the election fraud narrative and those who did not.

The authors of [8] analyzed a Twitter network related to the Venezuelan political crisis. This was a heavily polarized network, in which one community contained pro-Maduro people and the other contained pro-Gauido people. They found that influential users have an effect on the polarization of the network and can further exacerbate division among the users.

Being able to predict social media activity within various online topic discussions could be very useful. For example, one could use social media predictions to predict the spread of misinformation, or, one could detect whether certain tweets or Youtube videos might be very controversial and polarizing. These controversial posts could be counteracted by preemptively spreading content that seeks to unite people rather than divide them.

III. Temporal Graph Definitions and Measurement Discussion

A. Temporal Graph Prediction

This review focuses on social media prediction in temporal graphs. In the following subsections, the definitions for temporal graphs for Github networks and political discussion networks are given.

B. Github Temporal Graph Prediction

In Github, the temporal graph is represented as the tuple $G = (U, R, E, T)$. U is the set of all users such that $u \in U$ represents a user. R is the set of repos such that $r \in R$ represents a repo. The set, E can be thought of as a set of edge tuples, each of which of the form (u, r, a, t, w). This represents the interaction between user u and repo r. The term, a represents an action type in Github such as a Fork or Push. The term, w represents the number of times user u performed Github action a on repo r at some timestep t such that $1 \leq t \leq T$. The term, T represents the latest timestep of G.

There are various ways Github temporal graph activity can be predicted. Most of the works covered in this review tackle the problem of predicting user-repo links spanning from time $T + 1$ up to $T + S$, given input features up to time T. S represents the number of desired future timesteps that one wishes to predict.

A couple of works have slight variations on this task. The *RA-DCRNN* [9] aims to only predict the number of activities a repo r has performed upon it from $T+1$ to $T+S$. The user-repo interactions are not of concern. In the *DeepFork* paper [10], the authors aim to predict whether or not a link is created among a given *(user, repo, follower)* triplet, or (u, r, f_u), in which f_u is a user who follows u.

C. Political Discussion Temporal Graph

In the context of political discussion prediction, graph G represents any social media network containing political discussions, such as Twitter, for example. In this survey, there are two types of ways the graphs are modeled.

In the first representation, each temporal graph G can be thought of as a tuple of form (U, E, T), in which U is the set of users, and E is the set of edges of form (u, v, t, w). This tuple represents the number of times (w) user u interacted with user v at timestep t such that $1 \leq t \leq T$. The term, T represents the latest timestep of G.

The second type of temporal graph representation is that of the information cascade. Let I represent a temporal sequence of information cascade sets. Furthermore, let $I^{(t)}$ represent the set of information cascades that occurred at time t. Lastly, let $I_j^{(t)}$ represent the j^{th} information cascade in $I^{(t)}$. Each cascade $I^{(t)} = (c_j, \{u_k^{cas}\}_{k=1}^{K})$. The term, c_j represents the text content of $I_j^{(t)}$ (so, for example, the entire collection of tweets and retweet text in a given Twitter cascade). The sequence, u^{cas} represents the sequence of users in the cascade of interest $I_j^{(t)}$ in the order in which they posted in $I_j^{(t)}$.

In the cascade prediction papers, the task is to predict if a user $u_i^{global} \in u^{global}$ will be the next user to engage with cascade $I_j^{(t)}$. The term, u^{global}, is the sequence of all users that have appeared in any cascade up to time t. Note that this is a different sequence than u^{cas}, which only represents the users in some cascade of interest $I_j^{(t)}$.

D. Measurement Granularities

An important point of consideration is how each prediction task is measured. This paper classifies prediction tasks and measurements into 2 main categories, *macroscopic* and *microscopic* levels. Firstly, there is macroscopic measurement. This involves measuring a prediction of a high-level phenomenon. For example, one could measure the accuracy of the predicted time series of tweets in Twitter, or the predicted time series of new user creation in Youtube. One could also measure some high-level aspect of the predicted network, such as the distance between the predicted and ground truth networks' degree distributions.

Secondly, there is the more fine-grained *microscopic* measurement, that involves measuring the low-level phenomenon of a network prediction, specifically "who does what when", or "who engages with whom when". In the former case, we are just concerned with what an individual does at time $T+S$. In the latter case, we are concerned with (1) who did what, (2) when, and (3) with whom (link prediction).

Note that in this work, all approaches covered involve predicting user or user-to-user activity at some future timestep, however, not all approaches utilize microscopic measurements. This is because for some datasets, it is not feasible to predict microscopic measurements with reasonable accuracy.

More details regarding how each approach measures accuracy will be given later in the paper.

IV. Challenges

There are various challenges involved with predicting activity on social media. One of the most obvious difficulties is that of the inherent issue of predicting the future. This issue can further be exacerbated by the issue of noisy historical data collection. If the historical data needed as initial conditions for temporal prediction is noisy or incomplete, this can be make it difficult or even impossible to predict future events with even some small degree of accuracy.

In [11], the authors explored the effect of data filtering on the predictiability of two different cyber attack time series data sets. They showed that (1) the auto-correlation decreases at low sampling rates, (2) permutation entropy increases, and (3) the error of model-based techniques increases. Figure 1 illustrates this.

Another challenge in social media prediction is that of large data. There are can be potentially millions of users on various social media sites, so preprocessing steps or model training times can be greatly impacted by the overwhelming amount of data.

Lastly, too little data can be detrimental to model training as well. In [9], the authors noted the difficulty of predicting the time series of retweet activity for infrequently active users. Figure 2 shows a failed attempt to predict the time series of retweet activity from an infrequently active user. The authors note that the model simply predicted all 0's because the user was not frequently active.

In this work, we will discuss how various previous social media prediction works deal with these various challenges.

V. Application 1: Predicting Behavior in an Online Social Collaborative Platform (Github)

A. Github Prediction Types

In this section we discuss the different approaches used for Github prediction. While reviewing the various literature, a hierarchy for the frameworks was observed. Figure 3 contains a diagram of this hierarchy.

Firstly, we have the *Model Driven* vs. *Model Agnostic* frameworks. The *Model Agnostic* category contains 2 frameworks.

The *Model Driven* category contains more subcategories. It can be further divided into the *Non-Decompositional* and *Decompositional* approaches. The next set of subcategories are *Clustering-Based*, *Volume-to-User*, *Follower-Followee Driven*, and *Non-Follower-Followee Driven approaches*. The leaves of the hierarchy tree contain the different frameworks used in each category. In the following sections, these categories and frameworks will be discussed in more detail.

Note, we define framework as the simulation pipeline that is comprised of pipeline inputs, data pre-processing, model simulation, and pipeline output.

B. Model Driven vs. Model-Agnostic Frameworks

Figure 3 shows the *Model-Driven* and *Model-Agnostic* subcategories. A *Model-Driven* approach is defined as prediction pipeline comprised of (1) data as input, and (2) a model that performs predictions using this input data.

However, a *Model-Agnostic* approach is a prediction pipeline comprised of (1) data **and** a model as input, as well as (2) a program that performs predictions using both this model and data.

In this survey, 7 of the 9 known Github prediction approaches are *Model-Driven*, and 2 of them are *Model-Agnostic*. These two are the *FARM* [12]–[14] and *Matrix* [15] frameworks. These 2 approaches are Distributed-Computing Based. This means that they are programs made for a distributed computing system in which there is a controller process that communicates with multiple compute nodes. These Distributed-Computing frameworks emphasize usability and scalability.

The FARM [12]–[14] framework divides the simulation task among multiple computing nodes and uses a controller process in order to aggregate the final results and track progress. The Github graph is split up into multiple partitions using a graph partitioning algorithm and these partitions are placed into separate nodes for efficiency. The authors of [13] and [14] used 3 sampling-based agent models, a link prediction, and a Bayesian model with FARM and noted that by using FARM, they cut the runtime of these algorithms by 67% on average. It took FARM 20 minutes to simulate a Github network with 3 million users, 6 million repos, and 30 million events.

The authors of [15] introduced the *Matrix* Agent-Based Simulation Framework. Similar to FARM, this approach also utilizes a control process node connected to multiple computing nodes. In [15] a template is provided that a potential Github model must follow in order for it to be properly integrated. The Github network is described as a discrete dynamical system $S(G, K, F, W)$. $G(V, E)$ is the Github graph with a set of nodes V and edges E. $K_t \in K$ is the array of *vertex* states at time t. Each state x_i maps to a node $v_i \in V$ The authors do not specify what is meant by *state*. Perhaps this is because Matrix is supposed to be "model agnostic". Therefore, *state* will be different depending on what you want to predict. For example, the state could be a feature vector describing a node v at time t, or in the case of a binary classification, the state could be a simple 1 or 0 to indicate whether or not a user was active at time t. F is an array of local functions for each node in V. Each function is used to predict the state of node v_i at time $t+1$. Each node is mapped to its own function so that *Matrix* can perform parallel predictions for efficiency. Lastly, W is an update scheme function that sets the ordering for the node functions [15].

Matrix was able to simulate 3 million users, 13 million repos, and 239 million events in about 52 minutes [15].

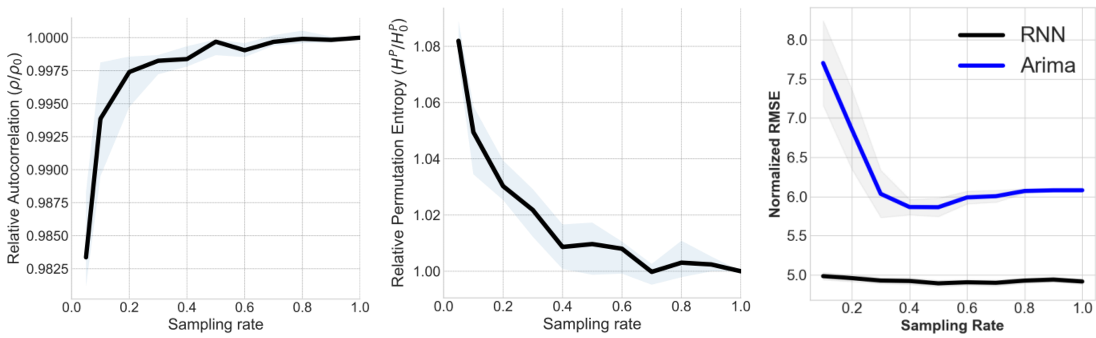

Fig. 1: These plots show the decay of predictability of randomly sampled ISI data. In (a) the auto-correlation decreases as the sampling rate decreases (from right to left). In (b) the permutation entropy increases as sampling rate decreases (from right to left), and in (c) the Normalized RMSE of the RNN model increases slightly, but the Normalized RMSE of the ARIMA model increases dramatically as sampling rate decreases (from right to left). This image is taken from [11] (page 6).

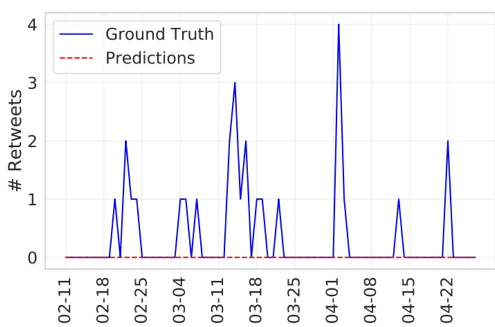

Fig. 2: This is a plot from [9] (page 5). It is a failed attempt by a model to predict the time series of retweet activity for an infrequently active user.

C. Decompositional Approaches

Next, there are *Decompositional* prediction approaches. These include the approaches that break the main prediction task into smaller subtasks. There are 2 ways that the prediction tasks were decomposed in the literature. They are the *Clustering-Based* and *Volume-to-User* based approaches.

D. Clustering-Based Decompositional Approaches

In the *Decompositional Clustering-Based* approach, user-repo pairs are predicted by first clustering the users into different categories based on 1 or more attributes, and then using these clusters to make more informed predictions of which user-repo pairs were active at time $T + 1$.

For example, the Archetype-ABM [16] is a framework that aims to predict user-repository activity through clustering and user archetypes. In this approach, K Means clustering was used to divide users into 16 different groups based on their average monthly activity for 14 different Github events [16]. These clusters were known as "archetypes". These archetypes were then used to predict the actual users, repositories, and events at some future timestep $T + S$.

Another clusering-based approach is the Proposed Community Features Model (PCFM) [17], which is an agent-based approach that utilizes community clustering to predict user-to-repo activity. Each community is defined using a topic based approach using the profiles of the GitHub repositories in order to generate a fixed set of communities. Some examples of topics used include programming languages, operating systems, and profile keywords [17].

E. Volume-to-User Decompositional Approaches

The *Volume-to-User Decompositional* approaches in this work aim to predict user-repo interactions in a two step approach. First, the overall activity time series is predicted for a particular Github event. Then, these macroscopic activity counts, along with user-repo pair features are sent through a 2nd module to perform the more microscopic task of predicting user-to-repo activity over time. The 3 frameworks that use this approach are the *CVE-Action-to-Pair (CVE-ATP) Model* [3], the *Cyber-Action-to-Pair (Cyber-ATP) Model* [18] and the SocialCube model as mentioned in [19] and [20].

The *CVE-ATP* and *Cyber-ATP* frameworks are 2 variations of the Action-to-Pair framework (ATP) [3], [18]. *ATP* is an LSTM neural network approach to Github user prediction. It decomposes the prediction problem into two tasks. Firstly, *ATP* predicts the daily-level volume of Github activities in the prediction time period of interest. This is the *Daily Level Prediction Task* [18]. It uses external features from Reddit and Twitter in order to make more accurate predictions. Then, *ATP*'s predicted daily counts, along with user-repo time series features are used to predict the number of events a user u performs on a repository r for each hour within each day of the prediction time frame of interest. This is the *Hourly User-Level Prediction Task* [18]. LSTM neural networks are used for both tasks.

The *ATP* framework was used to predict Github activity in two different datasets in two different works - [3] and [18].

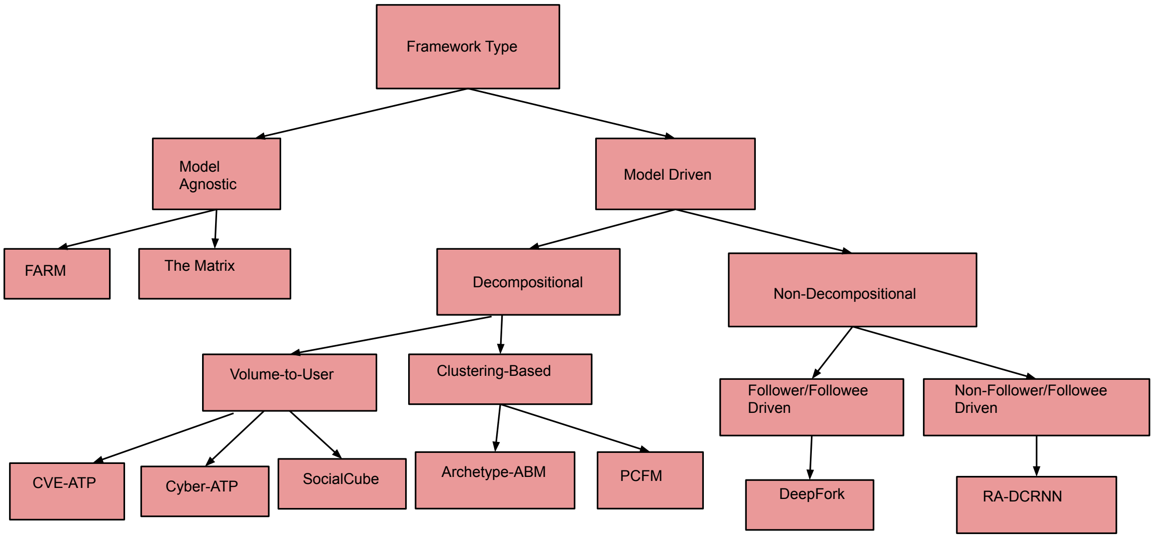

Fig. 3: The observed hierarchy of the Github frameworks discussed in this review

With this in mind, we will refer to this framework as two different frameworks. The *ATP* implementation in [3] will be referred to as *CVE-ATP* because it was used to predict activity related to repositories in the Common Vulnerabilities and Exposures database (CVE). The data company Leidos used this database to create the CVE repo dataset.

The *ATP* implementation in [18] will be referred to as *Cyber-ATP* because it was used to predict activity related to cybersecurity repositories. This data was also gathered by Leidos. They mined the text of issue comments in Github and added a repo to the cybersecurity list if the repo's associated issue comments contained keywords related to cybersecurity such as "security" or "bot", etc.

The Socialcube [19], [20] model predicts user-to-repo Github activity in two steps. First, an ARIMA model is used to predict the overall activity time series for 10 different Github events. Then, an ARIMA model is used to predict the activity time series of each user-repo pair.

F. Non-Decompositional Approaches

The *Non-Decompositional* approaches do not break the overall prediction task into subtasks. Instead, they directly predict the node or edge pair activity at time $T+1$. There are 2 types, *Follower-Followee Driven* and *Non-Follower-Followee Driven* approaches.

G. Follower-Followee Driven, Non-Decompositional Approaches

In the *Follower-Followee* driven approach, the framework predicts user and repo activity at time $T+1$ by explicitly leveraging the relationships between a user, u, a follower of user u, f_u, and a repo, r.

The approach in this category is the DeepFork Framework [10]. It is a neural network based approach that treats the user-repo prediction task as a binary classification. DeepFork attempts to predict "information diffusion" among a user, follower, and repo triplet at time $T+1$. In other words, instead of predicting whether (u, r) forms a link at $T+1$, it predicts whether a link is formed among (u, r, f_u) at $T+1$. In this case, f_u is a follower of u. This link represents the act of user u performing an action on repo, r; follower, f_u seeing this action, and then f_u also performing an action on repo r. In Github, it is possible for f_u to know u's actions because users can follow one another on Github.

H. Non-Follower-Followee-Driven, Non-Decompositional Approaches

The term *Non-Follower-Followee-Driven Approach*, refers to any type of framework that does not explicitly model the dynamics between (u, r, f_u) triplets. Instead, a framework in this category directly predicts the node or edge activity at time $T+1$ using some other means.

The work in the *Non-Follower-Followee* category is the *Repo Activity Diffusion Convolution Recurrent Neural Network* model, or *RA-DCRNN* of [9]. This model was used to perform temporal link predictions on a Github CVE (Common Vulnerabilities and Exposures) dataset. The dataset contains information pertaining to CVE exploits posted in cyber-security-related Github repositories.

In this work, the authors modelled the Github network as a homogenous, undirected network, in which all nodes were repos. Note this is different from the other works mentioned in which nodes were modelled as both users and repos. In this particular dataset, an edge exists between two CVEs in a given timestep t if both repos contain Github events pertaining to the same CVE identifier [9].

The *RA-DCRNN* is a Diffusion Convolutional Recurrent Neural Network. Each timestep represents the daily granularity, and the prediction task was as follows: Given the 7-day history time series for a repo, r, predict the number of Push Events performed on this repo over the next 7 days.

The Diffusion Convolutional Recurrent Neural Network is a recurrent neural network that utilizes two popular neural network layers, Gated Recurrent Units (GRU) [21] and Graph Convolutional Networks (GCNs) [22]. These types of networks are useful for performing sequence to sequence predictions with graph data.

I. Github Data Summary

Table I contains test set information for each of the different Github frameworks. The *What is Predicted* column shows the prediction task. The *Users*, *Repos*, and *Events* columns show the number of those elements in the test set. The *#Timesteps in Testing Period* column shows the number of timesteps in each testing period. The *Timestep Granularity* column shows the granularity of each time step. The *Prediction Period Dates* column shows the timespan of prediction. Lastly, the *Prediction Runtime* column shows how long it took the framework to perform the prediction. For example, the *FARM* framework was tested on the testing period spanning from 2/1/2018 to 2/28/2018. There were 28 days in this period, with a daily granularity. Lastly, note that any value that a given paper did not report was marked as "N.M.", which stands for "Not Mentioned".

VI. Application 2: Predicting Political Discussions in Social Media

Similar to the Github section, we discuss the different approaches used within the political discussion prediction literature. There are 5 different frameworks that will be discussed in this section. Each framework falls into one of two categories.

Firstly, there are 3 *Cascade Prediction* approaches, that aim to predict whether a particular user will engage with a cascade in the future. Secondly, there are the 2 *Non-Cascade* approaches, which are the approaches that predict future user activity without modelling the social network of interest as a cascade. Figure 4 shows an overview of the different types of political discussion prediction frameworks covered in this section.

A. Cascade Prediction Approaches

In [23], the authors utilize LinkLDA [24] and an extension of it called the CommentLDA algorithm [23] in order to predict whether a user would comment on a post on a social media site. The data used were blog posts and comments from 40 blog cites focusing on America politics during November 2007 to October 2008, during a presidential election. The discussions in these blogs related to Democratic and Republican candidates, speculation about the results, and international/domestic politics [23].

Both the LinkLDA and CommentLDA algorithms are extensions of the Latent Dirichlet Allocation algorithm (LDA), which is a generative, unsupervised machine learning algorithm for dividing documents into different subgroups by topic [25].

The LinkLDA is a variation of LDA that also predicts the most likely authors of a given document (in this context, a post), and the CommentLDA predicts both the most likely authors, and most likely words that these authors would comment on a post.

Another type of cascade-prediction approach can be found in [26]. The authors sought to better understand why various users on Twitter would retweet or reply to politicial tweets. To that end, they created various machine learning models, called *Political Tweet Engagement Models* to predict whether a user would respond to a post in a tweet cascade. They created a pool of "potential users" which was comprised of users who were more active in political discussions on Twitter. The authors obtained various insights from their experiments. For example, they found that the popularity of a tweet's author has little effect on predicting whether or not a tweet will receive a reply. Also, the authors found that users who have engaged in numerous comments in a thread are less likely to re-engage [26].

The authors of [27] introduced the *Dynamic Diffusion Variational Autoencoder*, or *DyDiff-VAE* model. This is a model that utilizes a Variational Autoencoder Neural Network that also contains a graph convolutional layer and a Gated Recurrent Unit (GRU). In this work, the authors sought to predict the probability that a user, $u_i^{global} \in u^{global}$ would engage with information cascade, $I_j^{(t)}$. Here, u^{global} is the sequence of all users who existed before time t.

The inputs to the model are as follows. Firstly, there is the vector $\mathbf{x_i}$, which is a Doc2Vec text embedding of u_i^{global}'s post history. Secondly, there's h^t, which the authors call the *social influence vector*. It represents the entire post history of the temporal network, G up to time t. Lastly, there's c_j, which is the Doc2Vec embedding of the text content of cascade $I_j^{(t)}$. The output of *DyDiff-VAE* is the probability that user u_i^{global} would engage with information cascade, $I_j^{(t)}$.

The authors trained and tested *DyDiff-VAE* on 3 Twitter datasets and 1 Youtube dataset. The 3 Twitter sets were related to the Venezuelan elections, general Venezuelan discussions, and White Helmets of Syria. The 1 Youtube dataset is related to general Venezuelan discussions.

B. Non-Cascade Prediction Approaches

In addition to the aforementioned *RA-DCRNN* model, the authors of [9] also created a User-Activity DCRNN model (*UA-DCRNN*). This was a Diffusion Convolutional Recurrent Neural Network Model trained on a Twitter dataset containing tweets related to the White Helmets of Syria. Each node in the temporal graph represented a user, and a user has an outgoing edge to a neighbor if that neighbor retweeted the user within the time frame of the dataset.

The authors of [28] created the *Node-Aware Attention Model*. This is a neural network-based model that utilizes graph convolutional layers, LSTM layers, and attention layers.

Github Test Set Data Information								
Framework	**What is Predicted**	**#Users**	**#Repos**	**#Events**	**#Timesteps in Testing Period**	**Timestep Granularity**	**Prediction (Test) Period Dates**	**Prediction Runtime**
FARM [12]–[14]	# of user-repo pair activities	3 mil	6 mil	30 mil	28	daily	2/1/2018 to 2/28/2018	20 min
Matrix [15]	# of user-repo pair activities	3 mil	13 mil	239 mil	14	daily	N.M.	52 min
Archetype-ABM [16]	# of user-repo pair activities	N.M.	N.M.	N.M.	N.M.	N.M.	N.M.	N.M.
PCFM [17]	# of user-repo pair activities	N.M.	N.M.	N.M.	N.M.	weekly	N.M.	N.M.
CVE-ATP [3]	# of user-repo pair activities	N.M.	N.M.	N.M.	744	hourly	8/1/2017 to 8/31/2017	N.M.
Cyber-ATP [18]	# of user-repo pair activities	2 mil	400,000	65 mil	744	hourly	8/1/2017 to 8/31/2017	3 hours
SocialCube [19], [20]	# of user-repo pair activities	N.M.	1000	1.4 mil	31	daily	7/1/2015 to 7/31/2015	6 hours
DeepFork [10]	user-repo-followee triplet	N.M.	N.M.	N.M.	1	monthly	1/1/2017 to 2/1/2017	N.M.
RA-DCRNN [9]	# of repo activities	n/a	22,052	36,584	237 days	daily	8/7/2017 to 3/31/2018	N.M.

TABLE I: Table of test set information for the Github prediction frameworks. Various pieces of information are shown such as the prediction task, number of users in the test set, number of repos in the test set, etc. “N.M.” stands for “Not Mentioned”. It means that the authors of the paper did not state the value in the given cell.

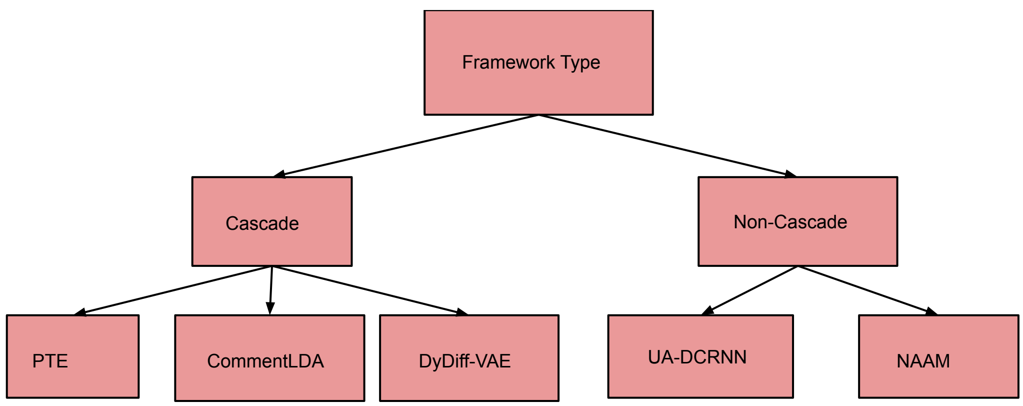

Fig. 4: The observed hierarchy of the political discussion prediction frameworks discussed in this review

The prediction task of this model was to predict how many times a news media Twitter account such as *@NewYorkTimes* would be retweeted or mentioned at time $T+1$. The nodes in the temporal graph represented the news media accounts and an edge existed between media accounts if a user tweeted or mentioned both media accounts within a given timestep t.

The input to the model was a sequence of 7 adjacency matrices that represented the Twitter network across 7 days. This sequence was fed into the graph convolutional layer, which then converted this matrix sequence into what the authors called a sequence of “daily” vectors. Each of the 7 vectors is a latent representation of the Twitter network on a given day. In other words, the graph convolutional network extracted features from the original adjacency matrix sequence. These daily vectors were then fed into the recurrent network layer, which converted these daily vectors into temporal embedding vectors. Next, an attention layer was used to create a score for every node in the network. Finally, the output of the model was 1 graph that included the number of times each news user account was (1) tweeted and (2) mentioned. So, each news node had 2 output values associated with it. Note that this model performed a “next day” prediction.

C. Political Discussion Data Summary

Tables II and III contain the test dataset information for the *Cascade Prediction* and *Non-Cascade Prediction* approaches, respectively. Some of the frameworks listed used multiple datasets, but in each table, we only report the information of the largest dataset used for the sake of space.

In Table II, the *What is Predicted* column shows the prediction task. Furthermore, there is information describing the number of users, cascades, and average cascade size of the test set. Table III shows data information in a similar format to I from the Github section.

Political Discussion Cascade Data Information				
Framework	**What is Predicted**	**#Users**	**#Casc.**	**Avg. Casc. Size**
CommentLDA [23]	user-to-cascade classification	16,849	240	N.M.
PTE [26]	user-to-cascade classification	26,096	7,053	5.6
DyDiff-VAE [27]	user-to-cascade classification	5,069	2,980	38.2

TABLE II: The test dataset information for the Political Discussion Cascade frameworks. Various pieces of information are shown such as the prediction task, the number of users in the test set, the number of cascades in the test set, and the average cascade size of the test set. "N.M." stands for "Not Mentioned". Information from the largest dataset by number of cascades are reported.

VII. Model Performance Results

There are many ways to evaluate success in social media prediction. In this section, the various ways of doing so are discussed.

A. Github - Model-Agnostic Results

The *FARM* framework was used to simulate the performance of 5 different models - a Bayesian model, a link prediction model, and 3 sampling-based models [13]. These models were evaluated on how well they predicted the popularity of the users and repositories. Rank Biased Overlap *(RBO)*, R^2, and *RMSE* metrics were used. The authors of [13] noted the sampling models performed the best. For example, for the R^2 metric for measuring the event count for issues per repository, the 3 sampling models all had R^2 scores of 0.74, 0.74, and 0.75; while the Bayesian and link prediction models had scores of 0.05 and 0.58, respectively (higher is better).

The *Matrix* framework was not evaluated on accuracy metrics, but was instead evaluated on runtime performance under multiple conditions. They found that runtime decreased approximately linearly with the addition of more CPU cores. Furthermore, they found that runtime performance increased approximately linearly with the addition of more users. The smallest number of users *Matrix* simulated was 300,000, and that took about 6.6 minutes. The largest number of users simulated was 3 million, which took about 52 minutes.

B. Github - Volume-to-User Results

For the *CVE-ATP* model, the authors measured the distribution of Fork and Watch events across Github repositories [3]. They noted that both distributions of events were close to the ground truth distributions. The JS divergence scores for Forks and Watches were 0.0029 and 0.0020, respectively (lower is better). For the R^2 metric, the scores were 0.6300 and 0.6067, respectively [3].

In the *Cyber-ATP* model in [18], the authors used a myriad of measurements and metrics to measure how well the overall simulated pattern of user-to-repo Github activity matched that of the ground truth. The model performed particularly well at predicting *Repo activity disparity* for Fork events. This is a measure of how well the simulated and ground truth Fork event distributions match up, and it was measured with Absolute Difference. The *Cyber-ATP* model's absolute difference was around 1, while the baseline "shifted" model's absolute difference was much higher, at 4.

SocialCube predicted the Github PushEvent, PullRequest, and ForkEvent 31-day time series using ARIMA [19]. *SocialCube* then used these initial predictions to perform the more fine-grained task of predicting the "affinity rates" of all the user-repo pairs. The affinity rate is the rate (between 0 and 1) at which a user u interacts with a repo r within the overall 31-day prediction period. The authors of [19] used Absolute Error to measure the ground truth affinity rates with the predicted affinity rates. They found that *SocialCube* strongly outperformed the baseline "Stationary" model, which was constructed by shifting the past 31 days of affinity rates into the next 31 days. The absolute error for *SocialCube* was around 2.1% while the baseline's error was around 2.4%. Furthermore, *Socialcube* gave better performance for 78.6% of the users in the test set, while the baseline gave better perfomance for 21.4% of users.

C. Github - Clustering-Based Results

The *Archetype-ABM* model was used to predict which repo a user would engage with. The authors of [16] used a parameter, σ to control the number of candidate repos that *Archetype-ABM* could choose from when predicting a repo for a particular user. In their experiments, the authors tested with various values of σ ranging from 1 to 32 [16].

The performance of *Archetype-ABM* was evaluated by measuring, with the Gini Coefficient, how well (1) activities were distributed among users and (2) how well activities were distributed among repos. They divided the activties into 3 categories: (1) Contributions, (2) Watches, and (3) Forks. Contributions are comprised of all Github activities that are not Forks or Watches such as Pushes, Pulls, etc. As a baseline model, the authors of [16] used a *mean model*, which predicts the number of activities in the future by using the historical mean of activities. Overall, the *Archetype-ABM* outperformed the baseline model. *Archetype-ABM* performed particularly well at predicting the distribution of Contribution Events over users. When $\sigma = 32$, *Error of Gini coefficient* for the baseline was around 0.5, while the error for *Archetype-ABM* was less than 0.1.

Archetype-ABM also outperformed the baseline when predicting the distribution of events over repos, albeit by not as much. When $\sigma = 32$, the baseline had an error of around 0.125 while *Archetype-ABM* had an error of around 0.1.

The *Proposed Community Features Model (PCFM)* [17] measured performance on a wide variety of metrics that fell into 4 different resolutions: (1) user (2) content (repo), (3) community (a subset of users), and (4) population (the entire Github network). The metrics used were (1) RMSE, (2) KS-test, (3) JS-Divergence, (4) Absolute Difference, and (5) Rank-Biased Overlap (RBO). *PCFM* was then compared with 3

Non-Cascade Political Model Test Data Information							
Framework	**What is Predicted**	**#Users**	**#Activities**	**#Testing Timesteps**	**Timestep Granularity**	**Prediction (Test) Period Dates**	**Prediction Runtime**
NAAM [28]	# user activities	202	N.M.	69	daily	11/1/2016 to 1/8/2017	N.M.
UA-DCRNN [9]	# user activities	6,376	241,778	79	daily	2/11/2019 to 4/30/2019	N.M.

TABLE III: The test dataset information for the non-cascade prediction political frameworks. Various pieces of information are shown such as the number of users in the test set, the number of activities in the test set, etc. "N.M." stands for "Not mentioned".

other baseline models. One overall metric score was calculated for each model by normalizing all metric results between 0 and 1 by measurement group and metric type. This final score was called the "normalized metric error" [17]. *PCFM* was found to have the lowest error scores out of the 4 models. For example, for the "user" category *PCFM* model had a normalized metric error of around 0.2, while the best baseline had an error of around 0.4.

D. Github - Follower/Followee Driven Results

The one Github model that performed a classification task was the *DeepFork* model [10]. Recall the prediction task was to predict whether a followee-follower-repo triplet would be active at future timestep $T + 1$. *DeepFork* outperformed its baselines, with an average F1 score around 70%. Perhaps the reason it achieved this somewhat high score is due to the setup of the prediction task. The authors used time T as the training period and it ranged from August 1 2016 to Dec 31 2016 (5 months). The granularity of $T + 1$ (the test period) was 1 entire month (Jan 1 2017 to Feb 1 2017). So, the task was for DeepFork to predict if a triplet would be active anytime within an entire month. Perhaps if $T + 1$ was a smaller granualrity (such as 1 day), the accuracy may have been much lower.

E. Github - Non-Follower/Followee Driven Results

Recall that for the *RA-DCRNN* model [9], the authors predicted sequences of repo-to-repo adjacency matrices over 7-day periods. They used RMSE and MAE of each graph instance to measure success. In order to alleviate the data sparsity issue of repo time series, they clustered repos according to different characteristics. However, they found that the more they clustered the repo nodes, the worse the metric performance became. For example, the most-clustered dataset (the one with the fewest nodes) had an MAE and RMSE of 6.37 and 29.92, respectively. However, the dataset with the least amount of clustering (so, the largest number of nodes), had an MAE and RMSE of 1.41 and 7.07, respectively. However, the authors do note that the model trained on the most-clustered dataset was best at predicting spikes in the time series. The problem was that these predicted spikes did not perfectly line up with the ground truth time series. So, perhaps different metrics could have been used to account for this fact [9].

F. Political Discussion - Cascade Prediction Results

The cascade prediction models were all used to solve classification problems. For example, in [23] the authors used precision to measure how many true positive commenting users were predicted out of the sum of true positive and false positive commenting users. The *LinkLDA* and *CommentLDA* were compared to a probabilistic baseline model. The *LinkLDA* model mainly outperformed the *CommentLDA* and baseline models. However, on its best performing dataset, the *LinkLDA* model still had a low precision score of 37%. For comparison, the *CommentLDA* model had a precision of 35% and the baseline had a precision score of 33%. These low precision scores are most likely due to the inherent difficulty of the task of predicting whether a user (out of a pool of many) will be the next user to respond to a cascade.

For the *DyDiff-VAE* model, the authors of [27] measured success with Recall and Mean Average Precision (MAP) of the predicted cascade users. Similar to the *CommentLDA* model, *DyDiff-VAE* outperformed its baselines, but still had very low scores. For example, on its best performing dataset for the recall metric, *DyDiff-VAE* had a score of 28.92%. The best performing baseline on this dataset had a recall of 25.41%. For MAP, *DyDiff-VAE* had a score of 7.64%, compared to the baseline's score of 4.54%.

Lastly, the authors of the [26] created 2 *Political Tweet Engagement (PTE)* models. One, was made with a Support Vector Machine and had an accuracy of 98%, and the other was made with the Logistic Regression algorithm and had an accuracy of 94%. [26]. Notice these scores are much higher than the other classification results of the other models. These scores are likely very high because the authors only sought to predict whether a user *who has already been engaged with a thread* will continue engaging with that thread. Note this is a much simpler task than what was discussed in the other papers, whose task was to predict if a user would engage with a thread, even if this user never had before.

G. Political Discussion - Non-Cascade Prediction Results

The non-cascade models were used to solve regression problems, similar to most of the Github models. For example, the *UA-DCRNN* model from [9] was used to predict the adjacency matrices of Twitter users over multiple 7-day periods in a similar matter to the *RA-DCRNN* model from the same work [9]. Similar to the *RA-DCRNN* model, the authors found that clustering users degraded performance. The model trained on the dataset with the fewest clusters had an MAE and RMSE of 15.26 and 56.71, respectively. However the model trained on the dataset with the most clusters had an MAE and RMSE of 0.80 and 2.07. Also, similar to the *RA-DCRNN* model, the authors noted that the model trained on the the dataset with the fewest clusters performed the best at predicting spikes, even though these spikes were misaligned with the ground truth.

The *NAAM* model [28] was used to perform next-day predictions of user-to-user adjacency matrices in Twitter. Mean Squared Error, Mean Absolute Error, Median Absolute Error, Root Mean Squared Error, Pearson Correlation, and R^2 metrics were used. The *NAAM* model outperformed the baseline models on most of these metrics. For example, the *NAAM* model had an R^2 score of 0.79 for retweet prediction, while the "previous day shifted" baseline had an R^2 of 0.72 (higher is better).

VIII. Future Work

A. The Volume-Audience-Match Simulator

Future work would involve more approaches that can predict the creation of new users as well as what activities they will do over a certain period of time. There are 2 works currently in progress within this domain ([29] and [30]). They are currently under review. They utilize a framework called the *Volume Audience Match* framework, or *VAM*.

VAM is comprised of 2 modules: the (1) *Volume Prediction Module (VP-Module)* and the *User-Assignment Module (UA-Module)*. The *VP-Module* takes as input the recent history of temporal graph G and predicts 3 time series of length S: (1) number of activities, (2) number of new users, and (3) number of old users. The *UA-Module* then assigns users to these predicted activities. Old users are assigned activities based on their historical behavior, and new users are assigned activities based on a *User Archetype Table*. This is a table that models the historical behavior of the old users, and is used to assign attributes to newly generated users.

In order to account for the new user prediction, VAM's predictive success is measured in the following way. The time series predictions are trivially measured with RMSE and MAE. The *UA-Module* predictions are measured in a less trivial fashion with the following 2 metrics: (1) Earth Mover's Distance [31] and (2) Relative Hausdorff Distance [32], [33]. For EMD, the Page Rank [34] distributions of the VAM-simulated network and ground truth network are created and then measured against one another. For RHD, the indegree distribution of the VAM-simulated network and the ground truth network are created and then measured against one another. Intuitively, these metrics are used to measure the "pattern" of activity between the simulated and ground truth networks.

B. VAM Literature Justification

Note that components of *VAM* have support and justification from the literature. For example, *VAM* is a *Volume-to-User Decompositional Approach* similar to the *CVE-ATP* [3] , *Cyber-ATP* [18] , and *Socialcube* [19], [20] frameworks. As discussed previously, these decompositional approaches were shown to perform well on their respective predictive tasks.

Similar to the *Archetype-ABM* in [16], *VAM* utilizes user archetypes for user-level activity prediction. *Archetype-ABM* was was shown to also perform well on its predictive task. Lastly, *VAM* uses historical sampling for its user-level predictions, similar to the sampling-based models used with *FARM* in [12]–[14] and the *Proposed Community Features Model* [17]. These approaches also yielded positive results in their predictive tasks as well.

C. VAM in Socialsim

VAM has been used to predict social network activity as a part of DARPA's Socialsim program. It is a research project started in 2017 in which teams from various universities research methods of efficient social media prediction [35].

Furthermore, VAM was used in a DARPA Socialsim competition in 2020. The prediction task was to predict the activity of users in Twitter and Youtube networks related to the Venezuelan Politcal Crisis. VAM placed 3rd of 52 model submissions by various universities [35].

In [30], VAM was used to predict Twitter and Youtube activity related to the Venezuelan Political Crisis. In [29] VAM was used to predict Twitter activity related to the Chinese Pakistan Economic Corridor (CPEC). These datasets were provided by DARPA as a part of the Socialsim Program [35].

The networks in both of these works are comprised of multiple sub-networks, in which each sub-network represents a conversation topic. For example, in the Venezuelan Political Crisis subnetworks, the various conversations included topics such as *Anti Maduro* (conversations denouncing Maduro) or *Anti Guaido* (conversations denoucing Guaido). In the CPEC dataset, the various conversations included topics such as the *Pakistan Sharif Leadership* or *CPEC Opposition*. The motivation behind predicting the future activity of these topics is that it can potentially allow for better understanding of political sentiment among the masses, and it can aid with tracking the potential spread of misinformation.

D. Future Uses of VAM

Future applications of VAM include using it for social media discussions relating to the Chinese Belt and Road Initiative in East Africa, which is the current research focus of the DARPA Socialsim Project. The social media platforms of interest are Twitter, Youtube, Reddit, and Jamii (a social media platform used in East Africa). A problem within this domain is that the activity of the Youtube and Jamii forums is quite sparse, so the use of data augmentation will be explored as a preprocessing step.

Lastly, with some modifications, *VAM* will be used on the Github datasets also provided by DARPA. *VAM* will be different than most of the Github prediction approaches discussed here in that it will predict new user activity in addition to old user activity. Most of the works in this survey only predicted old user activity.

IX. Conclusion

In this review we discussed recent social media prediction approaches within the domains of Github prediction and political discussion prediction. Within the realm of Github, the main approaches can be divided into the *Model Agnostic* and *Model-Driven* categories. Within the *Model-Driven* approaches, the different types discussed were *Volume-to-User*, *Clustering-Based*, *Follower/Followee Driven*, and *Non-Follower/Followee Driven*. For political discussion prediction, we discussed *Cascade* versus *Non-Cascade* prediction approaches.

We also discussed the challenges of social media prediction, such as scalability, noisy or uncertain data, and sparse data. Lastly, we discussed *VAM*, a prediction approach currently in progress.

This review can be used to guide researchers who are interested in creating social media prediction frameworks. Although we focused on Github and political discussion prediction, one should note that the ideas presented in these works are applicable (with perhaps some modifications) to other temporal network prediction domains.

References

[1] "Github homepage." [Online]. Available: https://github.com/

[2] T. Brewster, "How hackers broke equifax: Exploiting a patchable vulnerability," *Forbes*. [Online]. Available: https://www.forbes.com/sites/thomasbrewster/2017/09/14/equifax-hack-the-result-of-patched-vulnerability/?sh=b6a317d5cda4

[3] S. Horawalavithana, A. Bhattacharjee, R. Liu, N. Choudhury, L. O. Hall, and A. Iamnitchi, "Mentions of Security Vulnerabilities on Reddit, Twitter and GitHub," in *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (WI'19)*, Thessaloniki, Greece, Oct 2019.

[4] N. Choudhury, K. W. NG, and A. Iamnitchi, "Strategic Information Operation in YouTube: The case of White Helmets," in *Proceedings of International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation (SBP-BRiMS'20)*, Washington, USA, Oct 2020.

[5] S. Horawalavithana, K. W. NG, and A. Iamnitchi, "Twitter is the Megaphone of Cross-Platform Messaging on the White Helmets," in *Proceedings of International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation (SBP-BRiMS'20)*, Washington, USA, Oct 2020.

[6] K. W. NG, S. Horawalavithana, and A. Iamnitchi, "Multi-platform Information Operations: Twitter, Facebook and YouTube against the White Helmets," in *Proceedings of The Workshop Proceedings of the 14th International AAAI Conference on Web and Social Media (ICWSM) (SocialSens'21)*, Atlanta, USA, June 2021.

[7] S. Nair and A. Iamnitchi, "The Polarized Web of the Voter Fraud Claims in the 2020 US Presidential Election," in *Proceedings of The Workshop Proceedings of the 14th International AAAI Conference on Web and Social Media (ICWSM) (SocialSens'21)*, Atlanta, USA, June 2021.

[8] S. Horawalavithana, K. W. NG, and A. Iamnitchi, "Drivers of Polarized Discussions on Twitter during Venezuela Political Crisis," in *Proceedings of WebSci '21: The 13th International ACM Conference on Web Science (WebSci'21)*, Southampton, UK, June 2021.

[9] A. Hernandez, K. W. NG, and A. Iamnitchi, "Using Deep Learning for Temporal Forecasting of User Activity on Social Media: Challenges and Limitations," in *Proceedings of Temporal Web Analytics Workshop, Companion Proceedings of The 2020 World Wide Web Conference (TempWeb'20)*, Taipei, Taipei, April 2020.

[10] R. Akula, I. Garibay, and N. Yousefi, "Deepfork: Supervised prediction of information diffusion in github," in *Conference: Proceedings of the International Conference on Industrial Engineering and Operations ManagementAt: Bangkok, Thailand, March 5-7, 2019*, 03 2019.

[11] N. Tavabi, A. Abeliuk, N. Mokhberian, J. Abramson, and K. Lerman, "Challenges in forecasting malicious events from incomplete data," in *Companion Proceedings of the Web Conference 2020*, ser. WWW '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 603–610. [Online]. Available: https://doi.org/10.1145/3366424.3385774

[12] J. Blythe and A. Tregubov, "Farm: Architecture for distributed agent-based social simulations," in *Massively Multi-Agent Systems II*, D. Lin, T. Ishida, F. Zambonelli, and I. Noda, Eds. Cham: Springer International Publishing, 2019, pp. 96–107.

[13] J. Blythe, J. Bollenbacher, D. Huang, P.-M. Hui, R. Krohn, D. Pacheco, G. Murić, A. Sapienza, A. Tregubov, Y.-Y. Ahn, A. Flammini, K. Lerman, F. Menczer, T. Weninger, and E. Ferrara, "Massive multi-agent data-driven simulations of the github ecosystem," in *PAAMS*, 2019.

[14] J. Blythe, E. Ferrara, D. Huang, K. Lerman, G. Murić, A. Sapienza, A. Tregubov, D. Pacheco, J. Bollenbacher, A. Flammini, P.-M. Hui, and F. Menczer, "The darpa socialsim challenge: Massive multi-agent simulations of the github ecosystem," in *AAMAS*, 2019.

[15] P. Bhattacharya, S. Ekanayake, C. Kuhlman, C. Lebiere, D. Morrison, S. Swarup, M. Wilson, and M. Orr, "The matrix: An agent-based modeling framework for data intensive simulations," in *AAMAS*, 2019.

[16] S. Saadat, C. Gunaratne, N. Baral, G. Sukthankar, and I. Garibay, "Initializing agent-based models with clustering archetypes," in *Social, Cultural, and Behavioral Modeling*, R. Thomson, C. Dancy, A. Hyder, and H. Bisgin, Eds. Cham: Springer International Publishing, 2018, pp. 233–239.

[17] N. Hajiakhoond Bidoki, M. Schiappa, G. Sukthankar, and I. Garibay, "Modeling social coding dynamics with sampled historical data," *Online Social Networks and Media*, vol. 16, p. 100070, 03 2020.

[18] R. Liu, F. Mubang, L. O. Hall, S. Horawalavithana, A. Iamnitchi, and J. Skvoretz, "Predicting longitudinal user activity at fine time granularity in online collaborative platforms," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 2535–2542.

[19] T. Abdelzaher, J. Han, Y. Hao, A. Jing, D. Liu, S. Liu, H. Nguyen, D. Nicol, H. Shao, T. Wang, S. Yao, Y. Zhang, O. Malik, S. Dipple, J. Flamino, F. Buchanan, S. Cohen, G. Korniss, and B. Szymanski, "Multiscale online media simulation with socialcube," *Computational and Mathematical Organization Theory*, vol. 26, 06 2020.

[20] S. Yao, Y. Hao, D. Liu, S. Liu, H. Shao, J. Wu, M. Bamba, T. Abdelzaher, J. Flamino, and B. Szymanski, "A predictive self-configuring simulator for online media," in *2018 Winter Simulation Conference (WSC)*, 2018, pp. 1262–1273.

[21] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[22] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proceedings of the 5th International Conference on Learning Representations*, ser. ICLR '17, 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[23] T. Yano, W. W. Cohen, and N. A. Smith, "Predicting response to political blog posts with topic models," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Boulder, Colorado: Association for Computational Linguistics, Jun. 2009, pp. 477–485. [Online]. Available: https://www.aclweb.org/anthology/N09-1054

[24] E. Erosheva, S. Fienberg, and J. Lafferty, "Mixed-membership models of scientific publications," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101 Suppl 1, pp. 5220–7, 05 2004.

[25] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, p. 993–1022, Mar. 2003.

[26] S. Shugars and N. Beauchamp, "Why keep arguing? predicting engagement in political conversations online," *SAGE Open*, vol. 9, 03 2019.

[27] R. Wang, Z. Huang, S. Liu, H. Shao, D. Liu, J. Li, T. Wang, D. Sun, S. Yao, and T. Abdelzaher, "Dydiff-vae: A dynamic variational framework for information diffusion prediction," 06 2021.

[28] P. Shrestha, S. Maharjan, D. Arendt, and S. Volkova, "Learning from dynamic user interaction graphs to forecast diverse social behavior," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, ser. CIKM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2033–2042. [Online]. Available: https://doi.org/10.1145/3357384.3358043

[29] F. Mubang and L. O. Hall, "Simulating CPEC User-Level Twitter Activity with XGBoost and Probabilistic Hybrid Models," in *(In Review) 2021 IEEE International Conference on Systems, Man, and Cybernetics*, Oct 2021.

[30] ——, "VAM: An End-to-End Simulator for Time Series Regression and Temporal Link Prediction in Social Media Networks," *(In Review) IEEE Transactions on Computational Social Systems*, 2021.

[31] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases." in *IEEE Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, 1998, pp. 59–66.

[32] S. G. Aksoy, K. E. Nowak, E. Purvine, and S. J. Young, "Relative hausdorff distance for network analysis," *Appl Netw Sci 4*, p. 80, 2019.
[33] O. Simpson, C. Seshadhri, and A. McGregor., "Catching the head, tail, and everything in between: A streaming algorithm for the degree distribution." *2015 IEEE International Conference on Data Mining*, pp. 979–984, 2015.
[34] S. Brin and L. Page., "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, pp. 30(1–7):107–117, 1998.
[35] "Darpa socialsim challenge." [Online]. Available: https://www.darpa.mil/program/computational-simulation-of-online-social-behavior