Simulating CPEC User-Level Twitter Activity with XGBoost and Probabilistic Hybrid Models

Fred Mubang [1] and Lawrence O. Hall[1]

***Abstract*— The *Volume-Audience-Match* simulator, or *VAM* was applied to predict future activity on Twitter related to the Chinese-Pakistan Economic Corridor (CPEC). *VAM* was applied to do time-series forecasting to predict the: (1) number of total activities, (2) number of active old users, and (3) number of newly active users over the span of 24 hours from the start time of prediction. *VAM* then used these volume predictions to perform user link predictions. A user-user edge was assigned to each of the activities in the 24 future timesteps. VAM outperformed our baseline model in both the time series and user-assignment tasks.**

I. INTRODUCTION

Recent research strongly suggests that social media activity can serve as an indicator for future offline events. For example, the authors of [1] showed that Twitter user data could be used to predict the spatiotemporal spread of COVID-19. The authors of [2] found a strong correlation between the number of tweets mentioning each candidate in a given state, and the state's election results.

Clearly, more attention should be focused upon creating a simulator that can predict future social media activity at user and topic granularity. To that end, in this work we use the *Volume Audience Match* Simulator, or *VAM*, which was first introduced in [3]. *VAM* is a machine-learning and sampling driven simulator that predicts both overall activity volume and user level activity in a given social media network. *VAM* is comprised of 2 modules.

The first module is the *Volume Prediction Module*. This module predicts, over the next 24 hours, the future (1) activity volume time series, (2) active old user volume time series, and (3) active new user volume time series in some social media platform, for some given topic of discussion.

The second module in *VAM* is the *User-Assignment Module*. This module uses the 3 time series predicted by the *VP-Module*, as well as historical user-interaction information in order to predict, for a given topic, the user-to-user interactions over the next 24 hours from some start time T. We tested *VAM*'s predictive power on a dataset of tweets related to the China-Pakistan Economic Corridor (CPEC). As a baseline for comparison, we used a *Persistence Baseline*, which is created by shifting the events from the previous 24 hours over to the next 24 hours as a prediction. We found that *VAM* outperformed this baseline in both the *Volume Prediction* and *User-Assignment* tasks. Figure 1 contains a pictoral representation of *VAM* [3].

II. PROBLEM STATEMENTS

As previously mentioned are 2 problems *VAM* attempts to solve, the *Volume Prediction Problem* and the *User-Assignment Problem.*

A. The Volume Prediction Problem

The *Volume Prediction Problem* is to predict the overall volume of Twitter activities. Note that we do not distinguish whether a particular action is a tweet, retweet, quote, or reply. Let q be some topic of discussion on a social media platform such that $q \in Q$. Furthermore, let T be the current timestep of interest. The *Volume Prediction* task is to predict 3 time series of length S between $T+1$ and $T+S$. These time series, for a topic q, are the future (1) activity volume time series, which is the count of actions per time interval, (2) active old user volume time series which is the number of previously seen users taking action in a time interval, and (3) active new user volume time series which is the number of new users that take an action in a time interval. Note that in this work $S = 24$ in order to represent 24 hours [3].

B. The User-Assignment Problem

Before describing the *User-Assignment Problem* we must first define several terms. Let G be a sequence of temporal weighted and directed graphs such that $G = \{G_1, G_2, ...G_T\}$. Each temporal graph, G_t, can be represented as a set (V_t, E_t). V_t is the set of all users that are active at time t. E_t is the set of all user-to-user interactions, or links, at time t. Each element of E_t is a tuple of form $(u, v, w(u, v, t))$. u is the *child* user, or the user performing an action (such as a tweet or retweet). v is the parent user, or user on the receiving end of the action. The term $w(u, v, t))$ represents the weight of the outdegree between u and v at time t [3].

Now we discuss the *User-Assignment Problem.* The goal is to assign a user to each activity by the Volume Prediction Module, and to then assign edges between pairs of users. An edge between user A and B represents the act of user A retweeting a post by user B.

Given this information, let us say, for topic q you have the 3 volume time series as discussed in the *Volume Prediction Problem.* Your task is now to use these volume predictions, as well as the temporal graph sequence G to predict the user-to-user interactions for topic q between $T+1$ and $T+S$. This can be viewed as a temporal link prediction

*This work is partially supported by DARPA and Air Force Research Laboratory via contract FA8650-18-C-7825.

[1] Department of Computer Science , University of South Florida, 4202 E Fowler Ave, Tampa, FL 33620, USA fmubang@usf.edu lohall@usf.edu

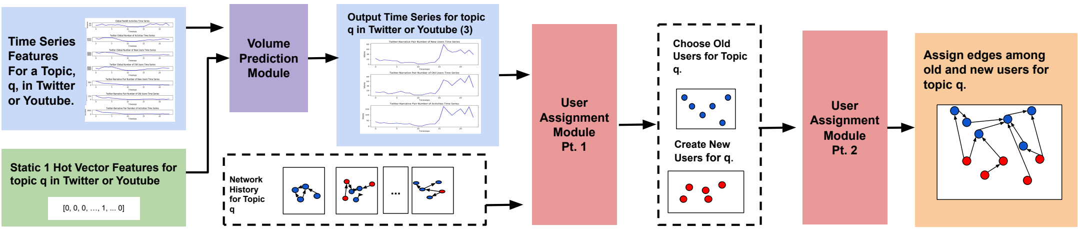

Time Series Features For a Topic, q, in Twitter or Youtube.
Static 1 Hot Vector Features for topic q in Twitter or Youtube
[0, 0, 0, ..., 1, ... 0]
Volume Prediction Module
Output Time Series for topic q in Twitter or Youtube (3)
Network History for Topic q
...
User Assignment Module Pt. 1
Choose Old Users for Topic q.
Create New Users for q.
User Assignment Module Pt. 2
Assign edges among old and new users for topic q.

Fig. 1: Framework for the Volume-Audience Match Algorithm (VAM). First, time series features and a 1 hot vector are inserted into the Volume Prediction Module. These features are used to predict the new user, old user, and activity counts. These volume counts as well as the temporal network history are then fed into the User Assignment Module to first choose the most likely old users and to create new users along with their attributes. Lastly, edges are assigned among these new and old users.

problem. These predicted user-user interactions are contained in a temporal graph $\{G^{future}\}_{t=1}^{S}$ such that $G^{future} = \{G_1^{future}, G_2^{future}, ...G_S^{future}\}$ [3].

III. Background

A. The Volume-Audience-Match Simulator

The *VAM* Simulator was first discussed in the technical report in [3], which contains all details. In this work, we focus on the performance of *VAM* on our Twitter CPEC dataset.

B. Social Media Time Series Prediction

The authors of [4] performed time series regression in the social media networks Facebook, Twitter, and Linkedin. They used various curve fitting models such as Polynomial, Logarithmic, and Exponential Regression. Similar to our work here, the authors also compared their models to a "temporal shift" baseline that utilizes the previous timesteps' values to predict the future timesteps' values. They were able to successfully outperform this baseline for some of their models.

In [5], the authors used LSTM networks with Twitter and Reddit features to predict the future time series of the Github social network. They found that using Twitter and Reddit features helped predict the Github time series more accurately. In [6], the authors used a sequence-to-sequence LSTM model along with Twitter and Reddit features to predict user activity in Github, and then aggregated the sequence blocks in order to derive the overall future time series predictions. In [7], the authors also used LSTMs to predict Github activity, but focused on predicting bursts. Lastly, in [8], time series prediction of Twitter and Github datasets, using Graph Convolutional Networks rather than LSTMs, was done.

C. Temporal Link Prediction

There have been several previous works on temporal link prediction algorithms. Some utilize neural networks that embed each node in a given network into a low dimensional space. In [9] the authors introduced *dyngraph2vec*, a temporal node embedding framework that utilizes autoencoders made up of LSTMs and fully-connected neural networks. There is also the tNodeEmbed framework as discussed in [10]. This embedding framework utilizes *node2vec* [11] and matrix rotation operations in order to create temporal embeddings.

The embeddings from [9] or [10] can then be used for temporal link prediction or node classification. In our work, we do not employ node embedding as it can be computationally expensive in terms of training time and space.

There are also matrix factorization approaches to temporal link prediction, which are discussed in [12], [13], and [14]. However, these approaches also struggle with scalability due to high computational cost. Lastly, there are temporal link prediction approaches which employ probabilistic methods, such as [15] and [16]. These methods have been shown to be effective but suffer from computational complexity in terms of space [16], and time in the case of [15] and [16]. And lastly, none of these approaches can predict the growth of new users, which is important for certain social networks in which activity is strongly driven by new users.

D. Additional Social Media Prediction Approaches

There are various works that predict social media activity in the online platform Github, that contain similar elements to those used of *VAM*. For example, there are approaches that utilize the two-step volume prediction/user-level prediction approach such as

IV. Data Collection

Data was collected and anonymized by Leidos. Annotators and subject matter experts (SMEs) worked together to annotate an initial set of 5,461 tweet and YouTube comments. All topics are related to the Chinese-Pakistan Economic Corridor. The time period was from April 2, 2020 to August 31, 2020. The final weighted average Cohen's Kappa (inter-annotator-agreement) for the 10 annotated topics was 0.84.

A BERT model was then trained and tested on this annotated data with a train/test split of 0.85 to 0.15. The average precision of the 10 topics was 0.83 and the average Recall was 0.78. The average F1 score was 0.8.

This BERT model was then used to label topics for 6,145,957 Twitter posts (tweets/retweets/quotes/replies) and 392,788 YouTube posts (videos and comments). Table I shows the counts of the Twitter and YouTube posts per topic.

Twitter and YouTube Topic Counts		
Topic	**# Tweets**	**# Youtube Posts**
other	3,282,978	387,803
controversies/china/border	1,509,000	1,081
opposition/propaganda	309,378	455
benefits/development/roads	189,082	937
leadership/sharif	185,851	648
controversies/china/uighur	173,431	440
benefits/development/energy	160,874	436
leadership/bajwa	144,277	494
benefits/jobs	112,769	267
controversies/pakistan/bajwa	78,317	227

TABLE I: Twitter and YouTube post counts per topic. Tweet counts refer to tweets, retweets, quotes, and replies. YouTube posts refer to videos and comments.

Twitter Topic Network Counts			
Topic	Nodes	Edges	Temporal Edges
other	806,741	1,985,880	2,762,331
controversies/china/border	443,666	1,170,374	1,438,123
opposition/propaganda	170,942	281,023	296,690
controversies/china/uighur	133,542	164,484	171,590
benefits/development/roads	74,042	148,345	179,432
benefits/jobs	71,914	98,038	110,304
benefits/development/energy	69,836	128,115	153,246
leadership/sharif	47,775	130,333	169,864
leadership/bajwa	35,320	87,836	99,783
controversies/pakistan/bajwa	35,219	63,567	74,610

TABLE II: Twitter network info by topic.

BERT was not applied to the Reddit data, so the Reddit data used as extra features in this work is not split by topics.

Table II contains the counts of each of the 10 Twitter networks. The largest network is *other*, with over 800,000 nodes, over 2 million edges, and over 2.8 million temporal edges. Note, an edge is defined as a user-user interaction (u, v), while a temporal edge is defined as a user-user interaction at some timestep t, or (u, v, t).

Lastly, Table III shows the average hourly proportion of new to old users in the Twitter dataset. Note that for some topics, there is a particularly high frequency of average new users per hour. For example, in controversies/china/uighur, on average, every hour 78.72% of the active users were new and 21.28% were old.Topics such as this are the reason that *VAM* seeks to predict both new and old user activity, unlike most previous works that only focus on old user activity prediction.

V. VOLUME PREDICTION METHODOLOGY

A. Data Processing

Our training period was from April 2, 2020 to August 10, 2020 (4 months). The validation period was August 11 to August 17th, 2020 (1 week). Lastly, the test period was August 18, 2020 to August 31, 2020 (2 weeks).

Each sample represents a *topic-timestep* pair. The input features represent multiple time series leading up to a given timestep of interest T. The different possible time series used for features are shown in Table IV. Also, a 1 hot vector of size 10 was used to indicate which topic each sample represented.

Twitter Hourly Active New/Old Frequencies		
Topic	**Avg. New User Freq (%)**	**Avg. Old User Freq (%)**
controversies/china/uighur	78.72	21.28
benefits/jobs	66.67	33.33
opposition/propaganda	59.74	40.26
controversies/pakistan/bajwa	52.63	47.37
leadership/bajwa	47.62	52.38
benefits/development/energy	47.5	52.5
benefits/development/roads	42.55	57.45
controversies/china/border	34.94	65.06
other	29.76	70.24
leadership/sharif	28.26	71.74

TABLE III: This table shows the hourly proportion of new to old users per topic.

Table V shows the feature sizes for each model trained. The *model* column shows the name of the model. The abbreviation represents the platform features used to train the particular model. "T", "Y", and "R" represent Twitter, YouTube, and Reddit respectively. The numbers represent the hourly length of the time series input to each model. However, note that the 3 output time series of each model are each of length 24 in order to maintain consistency in evaluation. For example, the *VAM-TR-72* model is a model trained on Twitter and Reddit time series that are all of length 72. Using table IV these time series indices would be 1-3, 7-9, and 13, or 7 different time series. Also recall, the 10 static features (for the 1 hot vector). So in total, this model had 7*72 + 10 = 514 features, as shown in the table.

There were 31,210 training samples used for each model, 1,450 validation samples, and 140 test samples. There are 140 test samples because of 10 topics and 14 days for testing. However, for training and validation, we wanted to generate as many samples as possible so our models had adequate data. So, for those datasets, we created samples by creating "days" both in terms of hour and day. We call this a "sliding window data generation" approach, similar to [3].

We trained 12 different models. Each model was trained on a different combination of platform features which were some combination of Twitter, Reddit, and YouTube. The time series features used for each platform are shown in Table IV. The names of the different models used are shown in Table VI. Furthermore, we also used different *volume lookback factors* (L^{vol}). The L^{vol} parameter determines the length of each time series described in Table IV. For example, the *VAM-TRY-24* model was the model trained on Twitter, Reddit, and YouTube time series, all of length 24.

B. XGBoost

VAM's *Volume Prediction module*, which we call Φ, is comprised of multiple XGBoost models. It takes an input vector, $\mathbf{x}$ and produces a matrix, $\hat{\boldsymbol{Y}} \in \mathbb{R}^{3\times S}$. In other words, $\Phi(\mathbf{x}) = \hat{\boldsymbol{Y}}$. Each row of this matrix represents one of the 3 volume time series (actions, new users, old users). Each column represents a timestep between T + 1 and $T + S$, with $S = 24$ hours in our experiments. As a result, there are 72 XGBoost models contained within the Volume Prediction Module Φ, each one "specializing" on an hour-output-type pair (e.g. number of new users in hour 1, or number of activities at hour 18, etc.). For more details see [3].

C. Parameter Selection

Similar to [3], we used the *XGBoost* [17] and *sk-learn* [18] libraries to create our XGBoost models. The subsample frequency, gamma, and L1 regularization parameters were set to 1, 0, and 0 respectively. A grid search over a pool of candidate values was done for other parameters using the validation set. For the *column sample frequency*, the candidate values were 0.6, 0.8, and 1. For the *number of trees* parameter, the candidate values were 100 and 200. For the *learning rate*, the values were 0.1 and 0.2. For *L2 Regularization*, the values were 0.2 and 1. Lastly, for *maximum tree depth*, the values were 5 and 7.

Mean Squared Error was the loss function and log normalization was used.

Time Series Index	**Time Series Description**
1	New user volume time series for a given topic in Twitter.
2	Old user volume time series for a given topic in Twitter.
3	Activity volume time series for a given topic in Twitter.
4	New user volume time series for a given topic in YouTube.
5	Old user time series for a given topic in YouTube.
6	Activity volume time series for a given topic in YouTube.
7	Activity volume time series across all topics in Twitter.
8	New user volume time series across all topics in Twitter.
9	Old user volume time series across all topics in Twitter.
10	Activity volume across all topics in YouTube.
11	New user volume time series across all topics in YouTube.
12	Old user volume time series across all topics in YouTube.
13	Activity volume time series in Reddit.

TABLE IV: All possible time series feature categories.

Model Input Feature Sizes	
Model	**Features**
VAM-T-72	442
VAM-TR-24	178
VAM-T-48	298
VAM-TR-72	514
VAM-TRY-72	946
VAM-TRY-24	322
VAM-TY-72	874
VAM-T-24	154
VAM-TY-24	298
VAM-TY-48	586
VAM-TRY-48	634
VAM-TR-48	346

TABLE V: Twitter volume model input sizes.

VI. VOLUME PREDICTION RESULTS

Table VI contains the RMSE and MAE results for the 12 *VAM* Volume Prediction models and the *Persistence Baseline*. Similar to [3], there are also *Percent Improvement From Baseline (PIMFB)* columns for both metrics. The PIMFB score shows by how much a given model's RMSE or MAE improved from the baseline. The formula for this value is given as follows:

$$PIMFB = 100\% * \frac{BaselineError - ModelError}{BaselineError}.$$

The upper bound of $PIMFB$ is 100%, which occurs if the VAM model's RMSE or MAE is 0. This is clearly the best possible result. The lower bound for $PIMFB$ is negative infinity because any given model could potentially perform infinitely worse than the baseline.

As seen in Table VI, the models are ranked from best to worst, by RMSE (lower is better). The *Persistence Baseline* came in 12th place, with an RMSE and MAE of 251.975 and 78.999, respectively. The best model was the *VAM-T-72*. This was the model trained on only Twitter features, with a *lookback factor*, or L^{vol} of 72. It had an RMSE of 221.209 and MAE of 62.276. The PIMFB scores of the RMSE and MAE are 12.21% and 21.168%, respectively.

Table VII shows the RMSE results for the *VAM-T-72* model, the top ranked *VAM* model. It was better than baseline for 9 out of 10 topics. Figure 2 shows some sample plots of time series predicted by the *VAM-T-72* model.

A. Temporal Feature Importances

In figure 3 we show a bar plot of the temporal feature importances of the XGBoost models for the *number of actions* output category for the *VAM-T-72* model. In this figure we refer to that output category as *Num. Twitter Actions For Topic*.

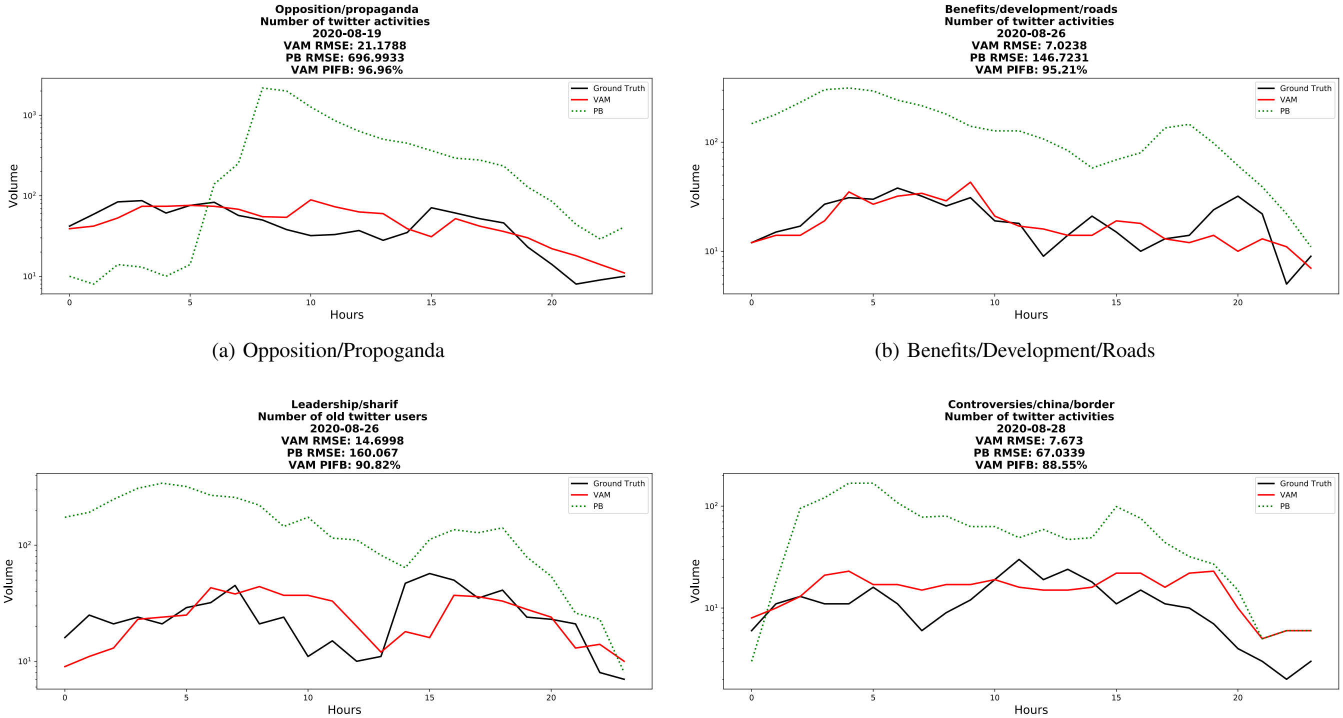
Along the Y-axis one can see the name of each feature category. There are 6 time series feature categories, 3 for the "global count" time series (the ones labelled with "All Topics"), and 3 categories for the "Twitter-topic" pair time series (the ones labelled with "For Topic"). We normalized all the feature category importances so that their values lied between 0 and 1, which is what you see in the bar plot.

As one can see, for the *VAM-T-72* model, the *Num. Twitter Actions For Topic* input time series is the most helpful for predicting the output category *Num. Twitter Actions For Topic*. In other words, according to this plot, if one wished to predict the number of actions for *benefits/jobs* at some future timestep, the most useful input time series would be the number of actions time series for *benefits/jobs*, which obviously makes sense. In second place, the old user time series *Num. Twitter Old Users For Topic* is most helpful for predicting *Num. Twitter Actions For Topic*, and in third place is the *Num. Twitter Old Users For All Topics*.

VII. USER-ASSIGNMENT METHODOLOGY

A. User-Assignment Lookback Factor

Similar to how the *Volume Prediction* modules utilized lookback factors (L^{vol}), we also utilized a lookback factor parameter for the *User-Assignment* task, L^{user}. We set this value to 24 hours. So, in other words, VAM's user-assignment module only uses the past 24 hours of user interaction history when making predictions. The assumption here is that recent user-interaction history is all that is needed to make accurate user-to-user predictions. We call this new



(a) Opposition/Propoganda

(b) Benefits/Development/Roads

(c) Leadership/Sharif

(d) Controversies/China/Border

Fig. 2: These are some time series plots showing 24-hour periods in which the *VAM-72* model performed particularly well against the baseline. The plots show the RMSE scores for both the VAM (solid red) and Persistence Baseline (dotted green) time series. The black curves represent the ground truth. The *PIFB* score shown in the title of each plot is the *Percent Improvement From Baseline* score of VAM against the baseline.

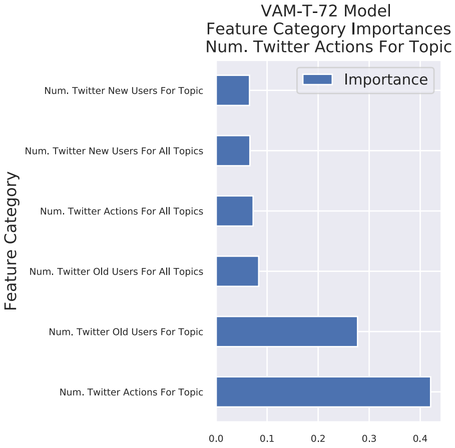


Fig. 3: The feature importances for the *VAM-T-72* volume prediction module.

VAM Volume Module Results					
Rank	**Model**	**RMSE**	**MAE**	**RMSE PIMFB (%)**	**MAE PIMFB (%)**
1	VAM-T-72	221.209	62.276	12.21	21.168
2	VAM-TR-24	221.892	62.721	11.939	20.605
3	VAM-T-48	223.225	62.752	11.41	20.566
4	VAM-TR-72	223.414	62.531	11.335	20.846
5	VAM-TRY-72	224.695	63.425	10.827	19.714
6	VAM-TRY-24	226.287	63.445	10.195	19.689
7	VAM-TY-72	226.856	65.082	9.969	17.617
8	VAM-T-24	228.646	63.525	9.258	19.588
9	VAM-TY-24	230.549	64.496	8.503	18.359
10	VAM-TY-48	233.602	67.347	7.292	14.75
11	VAM-TRY-48	241.944	65.695	3.981	16.841
12	Persistence Baseline	251.975	78.999	0.0	0.0
13	VAM-TR-48	582.253	79.749	-131.076	-0.948

TABLE VI: VAM Twitter Model Comparisons with Persistence Baseline. All VAM models outperformed the baseline except for the VAM-TR-48 model.

truncated version of the temporal sequence of graphs, G, G^{recent}. Using this information we now describe the user-assignment algorithm, which is the same as [3].

B. User Assignment Explained

A recent history table called H^{recent} is created from the history sequence of graphs, G^{recent}. This table contains event records, which each record being defined as a tuple containing (1) the timestamp, (2) the name of the child user, (3) the name of the parent user, (4) the number of interactions between the two users, (5) a flag indicating if the child user is new, and (6) a flag indicating if the parent user is new.

Using this table and the volume count of old users from the

Volume Prediction, module, *VAM* utilizes weighted random sampling to predict the set of active old users at T + 1, $\hat{O}^{T+1}$. Using the new user volume prediction counts, *VAM* is also able to create the set of active new users at T + 1, $\hat{N}^{T+1}$. Multiple data structures for each set of users are used to keep track of 4 main user attributes: (1) the user's probability of activity, (2) the user's probability of influence, (3) the user's list of parents it is most likely to interact with, and (4) the probability a user would interact with each parent in their respective parent list.

It is easy to obtain these 4 attributes for the old users because their history is available in the H^{recent} table. However, for new users, *VAM* must infer what their attributes would most likely be. In order to do this, *VAM* uses a *User Archetype Table*, which is created with the use of a random sampling algorithm applied to the set of old users in the H^{recent} table. The assumption is that new users in the future are likely to have the same attributes as old users in the recent past.

VAM then uses weighted random sampling in order to assign edges among the users in the $\hat{O}^{T+1}$ and $\hat{N}^{T+1}$ set. *VAM* "knows" how many total actions to assign among all users because the activity volume time series was predicted in the *Volume-Prediction* task. The final set of nodes and edges predicted at $T+1$ is known as G_1^{future}. *VAM* updates the history table H^{recent} with the new graph G_1^{future}, and then repeats the process of predicting old users, new users, and user-user interactions until it has predicted the full sequence $G^{future} = G_1^{future}, G_2^{future}, ...G_S^{future}$. Figure 4 contains an overview of the *User-Assignment* module. For more details, see [3].

VIII. USER-ASSIGNMENT RESULTS

A. Defining Success

Since our task involves predicting the creation and activity of new users, in addition to activity of old users, defining and measuring predictive success has complexities. The names of a new user are unknown before they appear in the ground truth. Hence, it is impossible to exactly match a new user that *VAM* generates with a new user in the ground truth. So, in order to work around this issue, we measure success using more macroscopic views of the network. Specifically, we used the Page Rank Distribution [19] of the weighted indegree of the network and the Complementary Cumulative Degree Histogram (CCDH) [20] of the unweighted indegree of the network. In order to measure the distance between the predicted and actual Page Rank distributions we used the Earth Mover's Distance Metric [21]. In order to measure the distance between the CCHD's of the predicted network and the ground truth network, the Relative Hausdorff (RH) Distance [20] was used.

VAM-T-72 RMSE Results

Topic	VAM-T-72 RMSE	Persistence Baseline RMSE	RMSE PIMFB (%)
leadership/sharif	**60.917**	86.407	29.5
leadership/bajwa	**456.096**	546.734	16.58
controversies/china/uighur	**67.59**	87.643	22.88
controversies/china/border	213.421	**200.408**	-6.49
benefits/development/roads	**42.98**	62.012	30.69
benefits/jobs	**49.175**	60.216	18.34
opposition/propaganda	**318.367**	350.179	9.08
benefits/development/energy	**47.648**	51.189	6.92
controversies/pakistan/bajwa	**168.916**	174.309	3.09
other	**301.756**	342.906	12.0

TABLE VII: VAM-T-72 RMSE Results

VAM-T-72V-24U Twitter Earth Mover's Distance Results

Topic	VAM Avg. EMD	Persistence Baseline Avg. EMD	VAM Percent Improvement From Baseline (%)
opposition/propaganda	**0.047847**	0.066306	27.84
benefits/development/roads	**0.063813**	0.088412	27.82
controversies/china/uighur	**0.05298**	0.067156	21.11
controversies/china/border	**0.072741**	0.087008	16.4
benefits/development/energy	**0.128978**	0.150268	14.17
other	**0.001282**	0.001485	13.7
leadership/sharif	**0.049239**	0.053797	8.47
leadership/bajwa	**0.157152**	0.160491	2.08
benefits/jobs	**0.158926**	0.161771	1.76
controversies/pakistan/bajwa	0.141172	**0.139003**	-1.56

TABLE VIII: VAM-T-72V-24U Twitter Earth Mover's Distance Results

B. User-Assignment Results

Table VIII shows the results for the Earth Mover's Distance metric. As one can see in the table we refer to this model as the *VAM-T-72V-24U* model. This is a *VAM* model that has a volume lookback factor (L^{vol}) of 72 hours and a user-assignment lookback factor (L^{user}) of 24 hours. The numbers in bold represent the best results. As one can see, *VAM* outperformed the baseline on this metric for 9 out of 10 topics. *VAM* performed particularly well for the *opposition/propaganda*, *benefits/development/roads*, and *controversies/china/uighur* topics. The percent improvement scores for those topics were 27.84%, 27.82%, and 21.11%, respectively.

Table IX shows *VAM*'s Relative Hausdorff Distance results. It beat the baseline on 8 out of 10 topics. It performed particularly well on the *leadership/sharif*, *controversies/pakistan/bajwa*, and *benefits/development/roads* topics. The percent improvement scores for those topics were 22.98%, 22.62%, and 13.16%, respectively.

Figure 5 shows the Earth Mover's Distance and Relative Hausdorff Distance for the *VAM-T-72V-24U* model. This is a *VAM* model that has a volume lookback factor (L^{vol}) of 72 hours and a user-assignment lookback factor (L^{user}) of 24 hours.

Since the user-assignment algorithm is probabilistic, we performed 5 trials, and averaged their resulting distance metric results. These average results are what is shown in the bar plots.

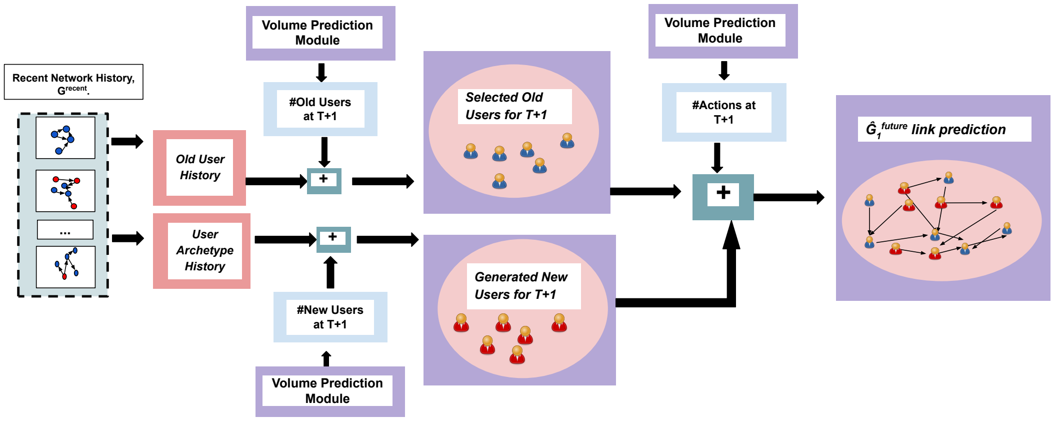
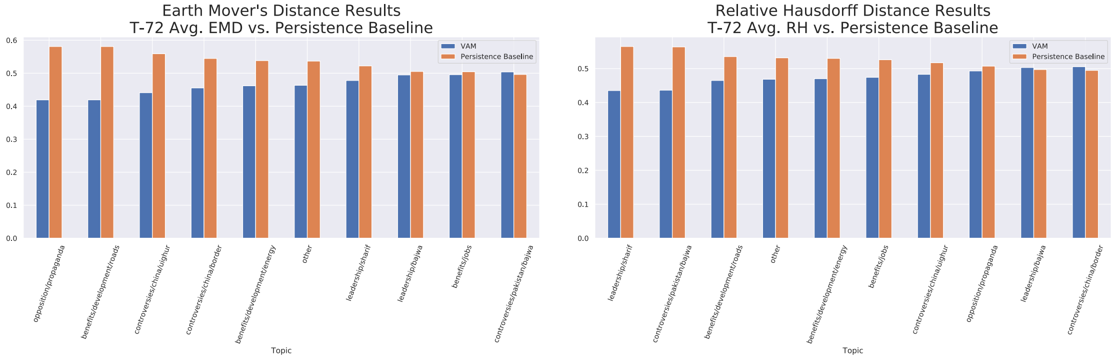


Fig. 4: This is an overview of the user-assignment module for 1 future timestep prediction at $T+1$. The recent network history (G^{recent}) is used to obtain *Old User History* and *User Archetype History*. This information, along with the counts from the *Volume Prediction* module, is used to predict the active old and new users at time $T+1$. These user sets, and the action volume counts are used to predict the links in the G_1^{future} set of edges for $T+1$.



(a) Twitter EMD Results

(b) Twitter RH Results

Fig. 5: These barplots show the Earth Mover's Distance and Relative Hausdorff Distance results for the Twitter *VAM-TR-24V-12U* vs. the *Persistence Baseline* models. Blue bars represent VAM results and orange bars represent baseline results. A lower bar for a model means a model outperformed the other for that particular topic.

VAM-T-72V-24U Twitter Relative Hausdorff Distance Results			
Topic	**VAM Avg. RH**	**Persistence Baseline Avg. RH**	**VAM Percent Improvement From Baseline (%)**
leadership/sharif	**0.795144**	1.032364	22.98
controversies/pakistan/bajwa	**0.976125**	1.26144	22.62
benefits/development/roads	**0.770693**	0.887479	13.16
other	**0.897848**	1.018806	11.87
benefits/development/energy	**0.6449**	0.726908	11.28
benefits/jobs	**0.663843**	0.736474	9.86
controversies/china/uighur	**0.75128**	0.804439	6.61
opposition/propaganda	**1.160409**	1.193958	2.81
leadership/bajwa	1.106825	**1.093718**	-1.2
controversies/china/border	0.958664	**0.937972**	-2.21

TABLE IX: VAM-T-72V-24U Twitter Relative Hausdorff Distance Results

C. Hardware and Runtime Information

The 5 trials were run in parallel across 5 nodes on a Sun Grid Engine Cluster. The CPU used on each node in the cluster was the Intel(R) Xeon(R) CPU E5-2630 v3 with a clock speed of 2.40GHz. It was comprised of 2 sockets, 8 cores per socket, and 2 threads per core. Each node had 128 GB of RAM. The average time of a given trial was about 52 minutes.

IX. CONCLUSIONS

In this work, we discussed the *VAM* Simulator [3], an end-to-end approach for time series prediction and temporal link prediction. We showed that it could outperform a *Persistence Baseline* model on both the *Volume Prediction* and *User-Assignment* tasks.

A simulator of the CPEC data is relevant for the following reasons. If a simulator can predict that there will be an increase in tweets related to the *opposition/progaganda* topic, that lets some government or corporate entity be aware that there might be growing opposition to the CPEC Initiative among the masses. On the other hand, if a simulator predicts that there will be an increase in tweets related to the *benefits/development/jobs* or *benefits/development/roads* topics, this lets some government or corporate entity know that people are potentially focusing on perceived benefits of the CPEC initiative such as an more jobs or better roads.

Future work would involve utilizing a machine-learning model for the *User-Assignment* module, as well as trying LSTM neural networks for both the *Volume Prediction* and *User-Assignment* modules.

ACKNOWLEDGMENT

The authors thank Leidos for providing the Twitter, YouTube, and Reddit data.

REFERENCES

[1] D. Bisanzio, M. Kraemer, I. Bogoch, T. Brewer, J. Brownstein, and R. Reithinger, "Use of twitter social media activity as a proxy for human mobility to predict the spatiotemporal spread of covid-19 at global scale," *Geospatial Health*, vol. 15, 06 2020.

[2] M.-H. Tsou and J.-A. Yang, "Spatial analysis of social media content (tweets) during the 2012 us republican presidential primaries," in *Seventh International Conference on Geographic Information Science (GIScience'12)*, 09 2012.

[3] F. Mubang and L. Hall, "*VAM*: An end-to-end simulator for times series regression and temporal link prediction in social media networks," *Technical Report ISL-41521*, 2021. [Online]. Available: www.cse.usf.edu/%7Elohall/ISL-41521.pdf

[4] M. Jayaram, G. Jayatheertha, and R. Rajpurohit, "Time series predictive models for social networking media usage data: The pragmatics and projections," *Asian Journal of Research in Computer Science*, pp. 37–50, August 2020.

[5] 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)*, Oct 2019, pp. 2535–2542.

[6] R. Liu, F. Mubang, and L. Hall, "Simulating temporal user activity on social networks with sequence to sequence neural models," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, October 2020.

[7] N. H. Bidoki, A. V. Mantzaris, and G. Sukthankar, "An lstm model for predicting cross-platform bursts of social media activity," *Information*, vol. 10(12), pp. 1–13, 2019.

[8] A. Hernandez, K. Ng, and A. Iamnitchi, "Using deep learning for temporal forecasting of user activity on social media: Challenges and limitations," in *Companion Proceedings of the Web Conference 2020*, April 2020, pp. 331–336.

[9] P. Goyal, S. R. Chhetri, and A. Canedo, "dyngraph2vec: Capturing network dynamics using dynamic graph representation learning," in *Knowledge-Based Systems,Volume 187*, January 2020.

[10] U. Singer, I. Guy, and K. Radinsky, "Node embedding over temporal graphs," in *Proceedings of the 28th International Joint Conference on AI (IJCAI-19)*, August 2019.

[11] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," vol. 2016, 07 2016, pp. 855–864.

[12] D. Dunlavy, T. Kolda, and E. Acar, "Temporal link prediction using matrix and tensor factorizations." in *ACM Trans. Knowl. Discov. Data (TKDD) 5(2)*, 2011.

[13] X. Ma, P. Sun, and G. Qin, "Nonnegative matrix factorization algorithms for link prediction in temporal networks using graph communicability," *Pattern Recognition*, vol. 71, p. 361–374, 2017.

[14] S. Gao, L. Denoyer, and P. Gallinari, "Temporal link prediction by integrating content and structure information," *In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, p. 1169–1174, 2011.

[15] P. Sarkar, D. Chakrabarti, and M. Jordan, "Nonparametric link prediction in large scale dynamic networks," *Electronic Journal of Statistics*, vol. 8, pp. 2022–2065, 2014.

[16] N. Ahmed and L. Chen, "An efficient algorithm for link prediction in temporal uncertain social networks." *Information Science*, vol. 331, pp. 120–136, 2016.

[17] T. Chen and C. Gestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2016, pp. 785–794.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[19] 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.

[20] 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.

[21] 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.