Simulating New and Old Twitter Activity with XGBoost and Probabilistic Hybrid Models - Supplemental Materials

Anonymous

Supplemental information to the main VAM paper

CONTENTS

I. Supplemental Material Information

This document contains supplemental information to the main VAM paper.

II. Problem Statements (More Detail)

As noted in the main paper there 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 because the focus of this work is to predict the overall volume of Twitter activities. Let q be some topic of discussion on a social media platform such that $q \in Q$, in which Q consists of all topics. 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) the active old user volume time series, which is the number of previously seen users performing an action in a time interval; and (3) the active new user volume time series, which is the number of new users that perform an action in a time interval. Note that in this work $S = 24$ in order to represent 24 hours [1].

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 [1].

Now we discuss the *User-Assignment Problem*. The goal is to assign a user to each activity predicted by the Volume Prediction Module, and to then assign edges between pairs of users. For tweets 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 there are 3 volume time series as discussed in the *Volume Prediction Problem*. The 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 problem. These predicted user-user interactions are contained in a temporal graph $\{G^{future}\}_{t=1}^{S}$ such that $G^{future} = \{G_1^{future}, G_2^{future}, ...G_S^{future}\}$ [1].

III. Full Data Collection Details

A. Raw Data Collection and Labelling

Data was collected and anonymized by Leidos. Annotators and subject matter experts (SMEs) worked together to annotate an initial set of 4,997 tweet and YouTube comments. These posts were related to 21 different topics, which are shown in Table I. This table contains the Weighted Average Inner-Annotator agreements on each of these topics. All topics are related to the Chinese-Pakistan Economic Corridor. The time period was from April 2, 2020 to August 31, 2020.

A BERT model [2] was trained and tested on this annotated data with a train/test split of 0.85 to 0.15. The F1 scores per topic are also shown in Table I. There was a wide range of F1 scores, with the highest being 0.97 and the lowest being 0. As a result, in order to avoid having an overtly "noisy" dataset, we only chose topics for our final Twitter dataset that had a Weighted Average Inner-Annotator Agreement of 0.8 or higher, and a BERT F1 score of 0.7 or higher. By doing this, we ended up with 10 topics, which are shown in bold in the table.

Twitter Topic Annotation Set Information			
Topic	**Weighted Average IAA**	**Label Count in Annotation Set**	**F1**
controversies/pakistan/students	**0.9308**	220	**0.97**
controversies/china/border	**0.9126**	309	**0.77**
leadership/sharif	**0.8980**	236	**0.86**
controversies/pakistan/baloch	**0.8589**	276	**0.71**
controversies/china/uighur	**0.8567**	25	**0.86**
leadership/bajwa	**0.8464**	722	**0.88**
benefits/development/roads	**0.8326**	571	**0.83**
benefits/covid	0.8276	242	0.67
benefits/development/energy	**0.8171**	335	**0.73**
benefits/jobs	**0.8124**	216	**0.75**
opposition/propaganda	**0.8046**	439	**0.75**
benefits/connections/afghanistan	0.7599	64	0.29
opposition/kashmir	0.7550	99	0.55
controversies/pakistan/bajwa	0.7533	165	0.73
controversies/china/exploitation	0.7379	210	0.57
leadership/khan	0.7376	246	0.63
controversies/pakistan/army	0.7269	129	0.19
controversies/china/naval	0.7261	24	0
controversies/china/funding	0.6225	46	0.4
benefits/development/maritime	0.6215	324	0.65
controversies/china/debt	0.6053	79	0.57

TABLE I: Twitter Topic Annotation Set Information. IAA stands for Inner Annatator Agreement. Topics were chosen for the Twitter dataset if the Inner-Annatator Agreement was at least 0.8 and if the F1 score of the BERT classifier on the test set was at least 0.7. The final chosen topics are in bold.

This BERT model was then used to label topics for 3,166,842 Twitter posts (tweets/retweets/quotes/replies) and 5,620 YouTube posts (videos and comments). Table II shows the counts of the Twitter and YouTube posts per topic. BERT was not applied to the Reddit data, so the Reddit data used as additional features in this work is not split by topics.

B. New and Old User Information

Lastly, Table III contains the average hourly proportion of new to old users in the Twitter dataset. As shown in

Twitter and YouTube Topic Counts		
Topic	**Twitter Counts**	**Youtube Counts**
controversies/china/border	1,509,000	1,081
controversies/pakistan/baloch	344,289	856
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/students	37,891	6

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

Twitter Hourly Active New/Old Frequencies		
Topic	**Avg. New User Freq (%)**	**Avg. Old User Freq (%)**
controversies/china/uighur	78.72	21.28
controversies/pakistan/students	75.0	25.0
benefits/jobs	66.67	33.33
opposition/propaganda	59.74	40.26
controversies/pakistan/baloch	50.0	50.0
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
leadership/sharif	28.26	71.74

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

the table, 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 we aim to use *VAM* to predict both new and old user activity, unlike most previous works that only focus on old/previous user activity prediction.

C. Twitter Network Counts

Table IV contains the node and edge counts of each of the 10 Twitter networks. The largest network in terms of nodes is the *controversies/china/border* network with 443,666 nodes. The smallest network in terms of nodes is the *controversies/pakistan/students* network, with 10,650 nodes.

Furthermore note that Table IV also contains columns for *Edges* and *Temporal Edges*. 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).

IV. Full 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

Twitter Topic Network Counts			
Topic	**Nodes**	**Edges**	**Temporal Edges**
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
controversies/pakistan/baloch	133,343	253,247	294,114
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/students	10,650	20,456	27,182

TABLE IV: Twitter network information by topic.

timestep of interest T. The different possible time series used for features are shown in Table V. Also, a 1 hot vector of size 10 was used to indicate which topic each sample represented.

Table VI 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 V 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 [1].

We trained 12 different VAM 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 V. 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 V. For example, the *VAM-TRY-24* model was the model trained on Twitter, Reddit, and YouTube time series, all of length 24.

B. XGBoost Parameter Selection

Similar to [1], we used the *XGBoost* [3] and *sk-learn* [4] 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*

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 V: All possible time series feature categories.

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

TABLE VI: Twitter volume model input sizes.

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.

C. ARIMA Models

As previously mentioned in the main paper, ARIMA, ARMA, AR, and MA models were used as baselines against VAM. The models were trained in the following way. The ARIMA model has $p > 0$, $d > 0$, and $q > 0$. The AR model has $p > 0$, $d = 0$, and $q = 0$. The ARMA model has $p > 0$, $d = 0$, and $q > 0$. Lastly, the Moving Average (MA) model has $p = 0$, $d = 0$, and $q > 0$.

In order train each of these ARIMA-based models, a grid search was performed with p and q's possible values being 0, 24, 48, and 72, and d's possible values being 0, 1, and 2. This is the same grid search approach used in [1]. A different model was trained per topic/output-type pair. So, for example, the *(Benefits/Jobs, # of new users)* pair had its own ARIMA, ARMA, AR, and MA models. The validation data was used to select the best model parameters for the test period and the *RMSE* metric was used to select the best model parameters.

V. Full User-Assignment Methodology

In this section, we discuss the methodology for the User-Assignment Module of VAM. Once the volume time series have been predicted (the number of actions, new users, and old users), they are then fed into the User-Assignment Module. The UA-Module then uses these time series to predict the user-to-user interactions over time.

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 truncated version of the temporal sequence of graphs, G, G^{recent}. Using this information we now describe the user-assignment algorithm [1].

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, with 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 **??** is a visual representation of the User Assignment algorithm. For more details, see [1].

VI. Volume Prediction Results (Additional Details)

A. Metrics Used

In order to properly assess VAM's predictive power in the time series prediction task, various metrics were used. We used RMSE and MAE metrics in order to assess how well VAM could predict time series in terms of volume and exact timing.

Predicting the exact timing of a time series is a difficult task. It is possible for a model to approximate the overall "shape" of a time series, while not correctly predicting the number of events or exact temporal pattern. In order to account for this phenomenon, we also use the Normalized RMSE metric. It is calculated in the following way. The ground truth time series and simulated time series are both converted into cumulative time series. Each time series is then divided by its respective maximum value. The result is 2 time series whose values range from 0 to 1. Finally, the standard RMSE metric is applied to these normalized time series.

In order to measure VAM's accuracy in terms of pure volume of events, without regard to temporal pattern, we used the Symmetric Absolute Percentage Error, or *S-APE*. This measures how accurate the total number of events was for each model, without regard to the temporal pattern. The formula is as follows. Let F be the forecast time series, and let A be the actual time series:

$$SAPE = \frac{|sum(F) - sum(A)|}{sum(F) + sum(A)} * 100\%$$

The last 2 metrics were used in order to measure how well the volatility of a predicted time series matches that of the ground truth. These metrics are *Volatility Error* (VE) and *Skewness Error* (SkE). The Volatility Error is measured by calculating the absolute difference between the actual and predicted time series' standard deviations. The Skewness Error is measured by calculating the absolute difference between the actual and predicted time series' skewness. The skewness statistic used in this work utilizes the adjusted Fisher-Pearson standardized moment coefficient [5].

B. Per Topic Analysis

Table VII shows a per-topic break down of VAM-TR-72 vs. ARMA's Overall Normalized Metric Error (ONME) results. Each ONME result for a particular model and topic is the mean of 72 values, which were the ONME results of 14 test days * 3 output types (activities, new users, and old users).

The Wilcoxon Signed Rank Test was used to determine significance of these results. The p-values from this test are shown per topic and statistically significant results are in bold. An alpha of 0.05 was used to determine significance. The "VAM-TR-72 is Winner" column contains a 1 if the VAM model's Overall Normalized Metric Error was lower, or better than ARMA's, and 0 otherwise. The "VAM-TR-72 is Statistically Significant Winner" column contains 1 if the win was statistically significant, and 0 otherwise. As one can see, the VAM-TR-72 model outperformed ARMA on all 10 topics, with 7 out of these 10 wins being statistically significant. There were noticeably large wins for several topics. For example, the *controversies/china/border*, *benefits/development/energy*, *benefits/development/roads* topics had Percent Improvement From Baseline scores of 50.6%, 41.79%, and 38.42%, which are quite large improvements from ARMA's results.

C. Full Cluster Analysis

We wanted to better understand the attributes of the time series VAM-TR-72 performed well on, relative to ARMA. To that end, we clustered the time series in the test set, and analyzed the average Overall Normalized Metric Error of those clusters. Recall that there were 10 topics, 3 output-types, and 14 test days in the test set, so 420 time series in total. We clustered these time series in terms of 2 attributes, "skewness" and "sparsity". We define the sparsity of a time series as the frequency of zeros within that time series. We wanted to analyze these attributes in particular because we wanted to know how well VAM performed on time series with both a high amount of 0's and low amount of 0's (measured by sparsity); and how well VAM performed on assymetrical, or potentially bursty, time series (as measured by skewness).

For all 420 we calculated these 2 values. We then clustered the time series based on the medians of these values. The time series whose value was equal to or below the median went into one cluster and any time series whose value was above the median went into another other cluster.

Using this methodology, four clusters were created. For the skewness attribute, "high-skewness" and "low-skewness" clusters were created. For the "sparsity" attribute, "high-sparsity" and "low-sparsity" clusters were created.

The median sparsity value was 0. There were 205 time series with sparsity of 0, meaning 205 time series contained no 0's, and only numbers 1 or higher. These time series were obviously placed into the "low-sparsity" cluster. There were 215 time series in the "high-sparsity" cluster, meaning they contained at least 1 or more 0's.

The median skewness value was about 1.27. There were 210 time series with a skewness equal to or lower than this value placed in the "low-skewness" cluster. There were 210 time series with a skewness above this value placed into the "high-skewness" cluster.

Note that since we were comparing skewness values among multiple time series, we normalized all time series between 0 and 1 using MinMax scaling before calculating the medians, in order to have a fair comparison among time series. No normalization was needed for the time series when calculating sparsity because sparsity is trivially calculated

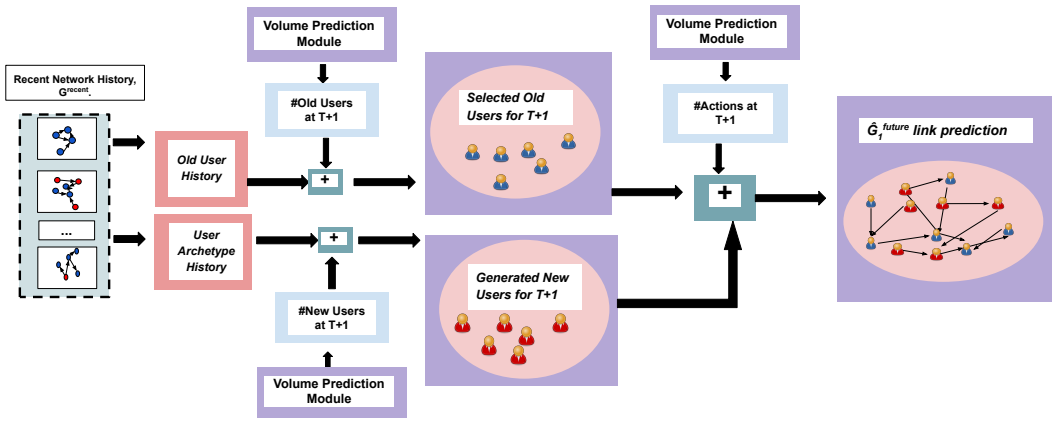


Fig. 1: This is an overview of the user-assignment module for 1 future timestep prediction at $T+1$. The recent network history (G^{recent}) is used to obtain *Old User History* and *User Archetype History*. This information, along with the counts from the *Volume Prediction* module, is used to predict 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 $T+1$.

VAM-TR-72 vs. ARMA Overall Normalized Metric Error Results						
Topic	**VAM-TR-72**	**ARMA**	**p value**	**VAM-TR-72 is Winner**	**VAM-TR-72 is Statistically Significant Winner**	**PIFB (%)**
controversies/china/border	**0.33047**	0.66953	2e-06	1	**1**	**50.6415**
benefits/development/energy	**0.367926**	0.632074	1e-05	1	**1**	**41.7907**
benefits/development/roads	**0.381091**	0.618909	9.3e-05	1	**1**	**38.4254**
benefits/jobs	**0.390389**	0.609611	0.000572	1	**1**	**35.961**
opposition/propaganda	**0.427561**	0.572439	0.021059	1	**1**	**25.3089**
controversies/pakistan/baloch	**0.427578**	0.572422	0.000656	1	**1**	**25.3037**
leadership/bajwa	**0.449846**	0.550154	0.002982	1	**1**	**18.2327**
controversies/pakistan/students	0.46092	0.53908	0.054945	1	0	14.4988
leadership/sharif	0.462587	0.537413	0.081114	1	0	13.9234
controversies/china/uighur	0.491394	0.508606	0.586504	1	0	3.3842

TABLE VII: VAM-TR-72 vs. ARMA Overall Normalized Metric Error Results

by dividing the number of 0's in a time series by the total number of values in a time series.

Figure 2a and 2b show the Overall Normalized Metric Error results of these clusters. As one can see, VAM outperformed ARMA on all 4 clusters. In other words, VAM outperforms ARMA on both highly-sparse, and lowly-sparse time series; as well as highly-skewed, and lowly-skewed time series. This shows that VAM is a versatile time series prediction method, which is ideal for highly-variable social media conversations on a platform such as Twitter.

VII. User Assignment Results (Additional Information)

A. Jaccard Similarity Explained

As mentioned in the main paper, in order to measure the accuracy of the old user prediction task, the Weighted and Unweighted Jaccard Similarity metrics were used [6]. Note that the Weighted Jaccard Similarity is also sometimes known as the Ruzicka Similarity, as it is referred to in [6]. It was used to measure how well VAM predicted the old users in each hour, as well as how "influential" they were. In this case, influence is defined quantitatively as the number of retweets, replies, and quotes a user's tweets received.

We used the Weighted and Unweighted Jaccard Similarity metrics in a similar fashion to the work of [1]. Let A represent the actual old user set within a particular hour, and let P represent the predicted set of old users within a particular hour.

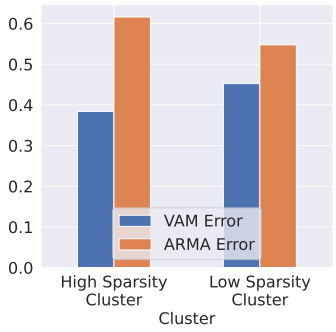
The Unweighted Jaccard similarity is trivially calculated as the cardinality of the intersection of A and P divided by the cardinality of the union of A and P. In other words, the Unweighted Jaccard Similarity is defined as follows:

$$J(A,P) = \frac{|A \cap P|}{|A \cup P|}$$

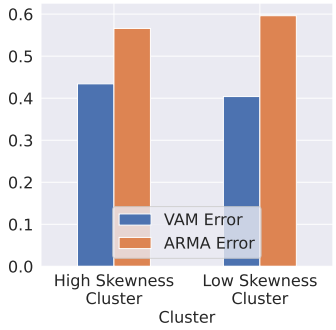
Furthermore, let $\mathbf{a}$ and $\mathbf{p}$ represent vectors that contain the weights of each user in the A and P sets, respectively. For example, $\mathbf{a}_k$ represents the weight of user A_k from the A set. With this in mind, the Weighted Jaccard Similarity is defined as follows:

$$J(\mathbf{a},\mathbf{p}) = \frac{\sum_k min(\mathbf{a}_k, \mathbf{p}_k)}{\sum_k max(\mathbf{a}_k, \mathbf{p}_k)}$$

.



(a) Sparsity Cluster Comparisons



(b) Skewness Cluster Comparisons

Fig. 2: VAM-TR-72 vs. ARMA cluster comparisons.

B. Old User Prediction Results

Tables VIII and IX show the Weighted and Unweighted Jaccard Similarity results, respectively. As previously mentioned, two models compared were the VAM-TR-72 and Persistence Baseline models. The format of the table is similar to the volume prediction result table VII. The VAM-TR-72 and Persistence Baseline Jaccard Similarity results are shown per topic. Instances in which VAM-TR-72 had a statistically significant and better score than the Persistence Baseline are in bold (higher is better). Similar to the Volume Prediction result table, (Table VII), the Wilcoxon Signed Rank Test was used to determine significance, with an alpha of 0.05. The p-values from the test are shown in the table.

For the Weighted Jaccard Similarity results, VAM-TR-72 outperformed the Persistence Baseline on 8 out of 10 topics, with all 8 of these wins being statistically significant. As one can see in the *PIFB* (Percent Improvement From Baseline) column, the VAM model had several quite considerable wins. For example, the PIFB scores for *benefits/development/roads*, *leadership/sharif*, and *controversies/china/uighur* were about 220%, 214%, and 120%, respectively.

For the Unweighted Jaccard Similarity results, VAM-TR-72 outperformed the Persistence Baseline on 8 out of 10 topics, with 7 of them being statistically significant. The PIFB scores are not as large as the weighted results, but still quite large all the same. For example, the *leadership/sharif*, *benefits/development/roads*, and *controversies/china/uighur* topics, VAM-TR-72 had PIFB scores of about 81%, 46%, and 43%, respectively.

In summary, VAM-TR-72 is considerably better than the Persistence Baseline model at predicting how influential old users will be over time.

C. Network Structure Results

Tables X and XI show the Weighted and Unweighted Earth Mover's Distance results, respectively. The format is similar to Tables VIII and IX. The model results per topic are shown, with statistically significant results indicated in bold.

For both the Weighted and Unweighted Earth Mover's Distance results, VAM-TR-72 outperformed the Persistence Baseline on 10 out of 10 topics, with 8 out of 10 wins being statistically significant.

In terms of PIFB scores, VAM-TR-72 had similar results between the two metrics. For example, for the Weighted EMD, the top performing topics were *benefits/development/roads*, *controversies/pakistan/baloch*, and *controversies/china/uighur*, with PIFB scores of about 29.8%, 23.67%, and 23.4% respectively. Furthermore, for the Unweighted EMD, top performing topics were once again *benefits/development/roads*, *controversies/pakistan/baloch*, and *controversies/china/uighur*. The PIFB scores were 32.71%, 23.91%, and 23.43%.

As one can see, VAM-TR-72 is also much better than the Persistence Baseline at predicting the network structure.

References

[1] 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*, 2022. [Online]. Available: https://fmubang.github.io/pdfs/VAM_Venezuela_5_11_22.pdf

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *ArXiv*, vol. abs/1810.04805, 2019.

[3] 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.

[4] 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.

[5] D. Doane and L. Seward, "Measuring skewness: A forgotten statistic?" *J. Stat. Educ.*, vol. 19, 07 2011.

[6] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *Int. J. Math. Model. Meth. Appl. Sci.*, vol. 1, 01 2007.

VAM-TR-72 vs. Persistence Baseline Weighted Jaccard Similarity Results						
Topic	**VAM-TR-72**	**Persistence Baseline**	**p value**	**VAM-TR-72 is Winner**	**VAM-TR-72 is Statistically Significant Winner**	**PIFB (%)**
benefits/development/roads	**0.1192**	0.0373	0.000000	1	**1**	**219.571**
leadership/sharif	**0.1352**	0.043	0.000000	1	**1**	**214.4186**
controversies/china/uighur	**0.1621**	0.0738	0.000000	1	**1**	**119.6477**
controversies/pakistan/baloch	**0.0567**	0.0308	0.000000	1	**1**	**84.0909**
opposition/propaganda	**0.0958**	0.056	0.000000	1	**1**	**71.0714**
benefits/development/energy	**0.0744**	0.0455	0.000000	1	**1**	**63.5165**
controversies/china/border	**0.0851**	0.0572	0.000000	1	**1**	**48.7762**
leadership/bajwa	**0.1008**	0.0878	0.002954	1	**1**	**14.8064**
benefits/jobs	0.068	0.0688	0.112933	0	0	-1.1628
controversies/pakistan/students	0.062	0.1118	0.042056	0	0	-44.5438

TABLE VIII: VAM-TR-72 vs. Persistence Baseline Weighted Jaccard Similarity Results

VAM-TR-72 vs. Persistence Baseline Unweighted Jaccard Similarity Results						
Topic	**VAM-TR-72**	**Persistence Baseline**	**p value**	**VAM-TR-72 is Winner**	**VAM-TR-72 is Statistically Significant Winner**	**PIFB (%)**
leadership/sharif	**0.2026**	0.112	0.000000	1	**1**	**80.8929**
benefits/development/roads	**0.1381**	0.0948	0.000000	1	**1**	**45.6751**
controversies/china/uighur	**0.2725**	0.1899	0.000000	1	**1**	**43.4966**
opposition/propaganda	**0.19**	0.1457	0.000000	1	**1**	**30.4049**
controversies/pakistan/baloch	**0.1279**	0.0986	0.000000	1	**1**	**29.716**
controversies/china/border	**0.1883**	0.1523	0.000000	1	**1**	**23.6376**
benefits/development/energy	**0.1334**	0.1155	0.018520	1	**1**	**15.4978**
leadership/bajwa	0.1221	0.1177	0.138814	1	0	3.7383
benefits/jobs	0.1022	0.1149	0.509420	0	0	-11.0531
controversies/pakistan/students	0.0957	0.1533	0.000318	0	0	-37.5734

TABLE IX: VAM-TR-72 vs. Persistence Baseline Unweighted Jaccard Similarity Results

VAM-TR-72 vs. Persistence Baseline Weighted Earth Mover's Distance						
Topic	**VAM-TR-72**	**Persistence Baseline**	**p value**	**VAM-TR-72 is Winner**	**VAM-TR-72 is Statistically Significant Winner**	**PIFB (%)**
benefits/development/roads	**0.6108**	0.8701	0.000000	1	**1**	**29.8012**
controversies/pakistan/baloch	**0.6235**	0.8169	0.000000	1	**1**	**23.6749**
controversies/china/uighur	**0.7629**	0.996	0.000000	1	**1**	**23.4036**
benefits/jobs	**0.5953**	0.75	0.000000	1	**1**	**20.6267**
benefits/development/energy	**0.6833**	0.8282	0.000000	1	**1**	**17.4958**
controversies/china/border	**0.6816**	0.7924	3.6e-05	1	**1**	**13.9828**
leadership/sharif	**0.6468**	0.7415	0.000706	1	**1**	**12.7714**
controversies/pakistan/students	**0.7788**	0.8458	0.008321	1	**1**	**7.9215**
opposition/propaganda	0.8391	0.8858	0.588399	1	0	5.2721
leadership/bajwa	0.6912	0.7277	0.168877	1	0	5.0158

TABLE X: VAM-TR-72 vs. Persistence Baseline Weighted Earth Mover's Distance

VAM-TR-72 vs. Persistence Baseline Unweighted Earth Mover's Distance						
Topic	**VAM-TR-72**	**Persistence Baseline**	**p value**	**VAM-TR-72 is Winner**	**VAM-TR-72 is Statistically Significant Winner**	**PIFB (%)**
benefits/development/roads	**0.5812**	0.8638	0.000000	1	**1**	**32.7159**
controversies/pakistan/baloch	**0.6074**	0.7983	0.000000	1	**1**	**23.9133**
controversies/china/uighur	**0.7617**	0.9948	0.000000	1	**1**	**23.4318**
benefits/jobs	**0.5524**	0.721	0.000000	1	**1**	**23.3842**
controversies/pakistan/students	**0.5027**	0.6453	0.000000	1	**1**	**22.0982**
benefits/development/energy	**0.6633**	0.8068	0.000000	1	**1**	**17.7863**
controversies/china/border	**0.6764**	0.7849	6.5e-05	1	**1**	**13.8234**
leadership/sharif	**0.6414**	0.7379	0.000464	1	**1**	**13.0777**
opposition/propaganda	0.8075	0.8713	0.243718	1	0	7.3224
leadership/bajwa	0.5996	0.6438	0.068773	1	0	6.8655

TABLE XI: VAM-TR-72 vs. Persistence Baseline Unweighted Earth Mover's Distance. Highlighted results are where VAM's wins were statistically significant.