Predicting Longitudinal User Activity at Fine Time Granularity in Online Collaborative Platforms

Renhao Liu and Frederick Mubang and Lawrence O. Hall and Sameera Horawalavithana and Adriana Iamnitchi and John Skvoretz

***Abstract*— This paper introduces a decomposition approach to address the problem of predicting different user activities at hour granularity over a long period of time. Our approach involves two steps. First, we used a temporal neural network ensemble to predict the number of each type of activity that occurred in a day. Second, we used a set of neural networks to assign the events to a user-repository pair in a particular hour. We focused this work on a subset of the public GitHub dataset that records the activities of over 2 million users on over 400,000 software repositories. Our experiments show we were able to predict hourly user-repo activity with reasonably low error. Our simulations are accurate for 1-3 weeks (168–504 hours) after inception, with accuracy gradually falling off. It was shown that activity on Twitter and Reddit increases the accuracy of activity prediction on GitHub for most events.**

I. Introduction

Simulating human behavior in complex online environments is challenging due to multiple factors that also include the tension between scale and accuracy [17], the complexity of user's actions and interactions, and time granularity. At the same time, accurately predicting users' online activities for long periods in the future can be used for adapting to variable computational loads, recognizing anomalies, and intervening in case of emergent undesired social phenomena.

This paper addresses the challenge of predicting user's hourly activities over up to a month in the future in a complex online collaborative environment, GitHub, with help from other social media platforms, Twitter and Reddit. GitHub [3] is a web-based software repository platform that provides a graphical interface for version control and various collaborative features. In Github, users contribute to software repositories via different types of actions (e.g., push, pull, issue comment). In order to limit the very large GitHub dataset, we selected only the repositories related to cyber security issues. We used the same criterion to extract related activities from Twitter (a highly popular microblogging service) and Reddit (an online topic-based discussions platform).

Given the nature of the GitHub platform, hour granularity brings significant challenges: first, the open-source repositories in the public GitHub dataset are contributed to by volunteers, thus without a predefined daily activity pattern. Second, user actions may be the result of significant time and effort investment (e.g., fixing bugs, introducing new features, identifying bugs), thus with wide time variation. Consequently, the naive approach of simulating activities in GitHub at the hourly level, where (user, repo, event) triplets are predicted, has the issue that in most hours a particular user will do no event to most of the (relatively few) repos they work on. So, a highly accurate prediction is that nothing will happen. The classes (actions for a user-repo for an event in an hour and no action) are highly imbalanced.

For these reasons, we decomposed the problem into two tasks. We first predict the number of each type of GitHub event performed in each day between August 1st 2017 and August 31st 2017. We will refer to this task as the Daily-Level Prediction Task. Using the predicted daily counts, we then predict which user-repository pairs (user-repo pairs) performed which actions in a given hour. We will refer to this task as the Hourly User-Level Prediction Task. To accomplish both tasks, we used LSTM recurrent neural networks [4] to capture the temporal aspects of the data.

We will show that trained models that perform long-term forecasting/simulation with 2 million GitHub users take a reasonably small time for predictions. Most GitHub events have their own weekly pattern, which can be learned for efficient daily count prediction on the platform level. Adding external features from Twitter and Reddit usually improved prediction performance of the number of events in a day. Finally, the simulator handles highly imbalanced data and shows promising user-level prediction performance for user engagement and repo popularity metrics.

II. Related Work

Work on using deep learning or recurrent neural networks for simulation over time for non-image data is not something we are aware of. There has been some work in trying to predict or forecast future events using learned models as discussed in the proceeding.

There are various papers that show positive results using websites and/or social media to predict the actions of a population within some paradigm. In [21], Pagolu et. al successfully observed a correlation between the sentiment scores of Twitter tweets, and stock market movements. They extracted sentiment features using N-Gram representation and Word2Vec and then fed these sentiment features into Random Forests to build a classifier. The classifier predicted if the previous day stock price is more than the current day stock price. In [24] it was also shown that sentiment could be shown to point to changes in stock prices. Cryptocurrencies price direction was predicted from the sentiments of

*Partially supported by DARPA SocialSim and NSF grant 1513126.

Department of Computer Science and Engineering, ENG 060, University of South Florida, Tampa, FL 33620, lohall@mail.usf.edu

comments in related online communities in [13].

Pedestrian motion was predicted in [8] from a short history of their and neighbors past behavior. The approach used learning for the predictions. The prediction of trajectory of people in crowded spaces was addressed in [1] using LSTM based neural networks, as done here. There has been work on predicting how a patient will do over time in the ICU using a learned model [19].

There has also been work on agent based simulations of social systems [9] which has a learning component for the agents, but it is a different type of low-level approach than taken here. Here, we use purely learned models and look at simulating results over time using daily predictions to make predictions further in the future.

As a comparison approach, LightGBM [11] is a fast gradient boosting framework that uses a regression tree based learning algorithm. It has performed well in numeric prediction, as done here.

III. Datasets

The focus of our experiments is the activity on public domain GitHub repositories (repos) related to cyber (computer) security. A set of keywords in comments that showed a focus on cyber security or that it was affected by security issues gave us quite a few repos and associated users. Related subReddit's are used and Tweets that have one of a specific set of (3476) keywords are also used [16].

Data for all platforms was available from January 1, 2017 to August 31, 2017. Test data was the from August 1, 2017 to August 31, 2017. The 14 GitHub events predicted were (1) Push, (2) Create, (3) Watch, (4) Issue Comment, (5) Pull Request, (6) Issues, (7) Fork, (8) Delete, (9) Pull Request Review, (10) Gollum, (11) Commit Comment, (12) Public, (13) Member, and (14) Release.

A. GitHub

Github is primarily an open-source software collaboration platform where users contribute to Github repositories via (code) commits, pushes, pull-requests, and issues raised. Users can also "watch" repositories to receive alerts on updates, and can "fork" (i.e., copy) public repositories to make their own software modifications.

GitHub is home to over 100 million public repositories and 30 million users. The dataset of events on the public repositories is publicly available [3]. In total, our Cyber dataset includes over two million users who contributed to over 400,000 software repositories via more than 65 million actions (Table I).

B. Reddit

Reddit is a popular website where users post content on a bulletin board system, comment on each other's posts, and vote them up or down. Content is organized into topic-specific subreddits. Users can post content, comment, or vote once they are logged into their account. Users can also befriend each other (similar to an online social network), in which case they receive updates on their friends' actions on Reddit. Users can also subscribe to subreddits to personalize what content they see. Our dataset covers the complete conversation threads related to 36 subreddits that are usually on cyber-security topics, such as *r/hacking*, *r/security*, *r/privacy*, *r/piracy*.

TABLE I: Dataset activities: posts (7.3%) and comments (92.7%) in Reddit, tweets (30.30%), re-tweets (62.05%) and replies (7.65%) in Twitter, and 14 events in GitHub

Dataset	Activities	Users	Communities
Reddit	8,166,033	788,598	36
Twitter	12,303,032	3,678,215	263,678
Github	65,520,077	2,496,045	403,287

C. Twitter

Twitter is a micro-blogging platform where users broadcast messages (i.e., tweets) publicly or share privately to their follower network. Twitter allows tweets to be tagged with hashtags, and users can post any messages, URLs, images, etc., under one or multiple hashtags. We used 3476 keywords to filter tweets related to cyber threats.

Generally, communities are of interest in and across platforms. Communities are subreddits in Reddit, hashtags in Twitter, and software repositories in GitHub.

D. The GitHub Imbalanced Data Problem

The imbalanced data problem is prevalent in many areas of machine learning [5]. In our experiments, we were confronted with this problem when trying to predict Github user-level activity. In Github, as with many social media platforms, user activity is sparse at the hour-level. In other words, a user may perform 1 or more actions in Github in a given hour, and then do nothing for possibly hours, days, weeks, months, or even years at a time. So, it around 99% accurate to say a user did nothing to a repo in an hour.

IV. Simulation approach

In order to ameliorate the imbalanced data problem, we decomposed our prediction task into two parts. First, we sought to predict the number of events in a day on Github, and then we sought to predict which user-repository pair performed which event in a particular hour.

A. Daily-Level Prediction Task

The Daily Level Prediction Task predicts the number of each of the 14 Github events that occurred each day in August 2017. Our data was collected in a collaboration with Leidos and included the time period spanning January 1st 2017 to August 31,2017 because data for this time period across all 3 platforms was available to us.

B. Using Reddit and Twitter Activity to Predict Github Activity

We experimented with various sets of features for the daily level task. We sometimes achieved our lowest prediction error when including event count information from Twitter and Reddit. At times adding features for the sentiment of

the information posted to the two platforms also helped. So, it was usually the case more accurate predictions of Github activity could be obtained when using Twitter and Reddit features in addition to Github features.

Both Twitter and Reddit had 6 types of features that we used for our predictions. For Twitter they were: (1) tweet count, (2) tweeting user count, (3) retweet count, (4) retweeting user count, (5) twitter polarity mean, and (6) twitter subjectivity mean. For Reddit they were: (1) post count, (2) author count, (3) subreddit count, (4) reddit polarity mean, (5) reddit subjectivity mean, and (6) comment count.

C. Performing Cold-Start Forecasting/simulation with an LSTM

"Cold-Start" forecasting is a problem within the realm of time series prediction [26]. This type of forecasting involves predicting the target information for multiple timesteps while only using 1 "initial" timestep for these predictions. One can view this as a simulation of activity using initial conditions.

A predictor within the cold-start paradigm uses the features of the initial timestep to predict the values at the next time step, and then uses that prediction in generating the prediction for the next time step, repeating this process until activity for the entire period of interest has been predicted.

In most literature discussing neural network time series prediction, a given prediction is made using the ground truth of the previous time step. Our work seeks to actually "simulate" multiple time steps, given data for only 1 initial time step.

D. Hourly User-Level Prediction Task

We first created a model that would predict user-repo activity without the current daily count information, with the hope that the network could predict user activity at the granularity of an hour without this prior information. In this previous approach, each sample was a representation of a specific user-repo pair's activities within a 24-hour window. The target value was the number of activities the user-repo pair would perform in the 25th hour.

This approach failed to deliver, most likely because, when predicting events on Github at the hourly level using a learned model, there is a large class imbalance problem. Most users do nothing to any repo in a given hour. So, statistically, the best prediction for the model to make regarding the number of events for a user-repo pair is 0. While some methods can be used to deal with imbalance, we could not add artificial events due to memory limitations and had to subsample [5].

So we first predict daily counts for all events and then use a separate model to predict which users did which event to which repo in a given hour. Through our experiments, we found our two-step process allowed us to predict hourly user-repo activity with reasonable success. In the proceeding, we describe our results in detail and show the various metrics used to rate our performance.

V. Simulation Process

The two-part decomposition approach to predict the hourly user activity is discussed next. First, the daily event counts prediction model is presented followed by user-repo-hour assignment to events system.

A. The Daily Count Models

In order to predict the number of events in Github on the daily level, we experimented with ensembles of models trained on three different datasets. Each ensemble was made up of 10 models with each model created using a different random seed for weight initialization. The model predictions were averaged in order to obtain one set of event counts. The neural networks had an LSTM layer with 500 units, followed by a fully connected layer of 10 units and then a linear output layer that used a single set of weights to each of the linear output units. The ensembles were created as follows.

The Github Only Ensemble (GO): During training and testing, the feature vector, X, for this ensemble contained the daily counts of the 14 Github events over the span of a week. The target vector, Y, contained the delta of each event count for each day in the next week. When training and testing, X contained 98 features (14 event counts over 7 days) and Y contained 98 values (14 event deltas over 7 days). Since there were 10 models in this ensemble, there were 10 different values per each of the 98 model outputs, or 980 model outputs in total. These were then averaged by event to yield the final 98 model outputs.

The Github-Twitter-Reddit-with-Sentiment Ensemble (GTR-WS): For this ensemble the feature vector, X, contained the daily counts of each of the 14 Github events, 4 Reddit events and associated 2 average sentiment features, and 4 Twitter events and associated 2 average sentiment features over the span of a week. Similar to the GO ensemble, the target vector, Y, contained the delta of each event count for each day in the next week. So, the feature vector X and outputs Y contained 182 features (26 event counts over 7 days).

The Github-Twitter-Reddit-with-NO-Sentiment Ensemble (GTR-NS): During training and testing, the feature vector, X, for this ensemble contained the daily counts of each of the 14 Github events, 4 Reddit events, and 4 Twitter events over the span of a week. Everything is the same as the previous set of models except no average sentiment features were used.

1) Daily Model Training and Testing Data: The training set for each of the model sets consisted of 26 weeks. The 26 weeks for these samples were in the range of January 6th, 2017 to July 6, 2017. The target vectors for each of the feature vectors were made from the weeks spanning January 13th, 2017 to July 13, 2017. One sample (1 week) was used for the validation set. The feature vector was the week spanning from 7/14/2017 to 7/20/2017, and the target vector was the week starting from 7/21/2017 and ending on 7/27/2017.

The test set was comprised of 5 samples spanning the last few days of July up until the end of August, or the weeks spanning from 7/28/2017 to 8/31/2017. However, the

focus of our analysis was just on the month of August, so **when evaluating, we ignored July 28 to July 31**. Since we performed experiments with cold-start forecasting, we needed a week of initial conditions, so for that we used the week of 7/21/2017 to 7/27/2017.

B. Pair-Assignment Models

To predict the activity of the user-repo pairs at the hourly level, we trained 14 models, one for each Github event. The event-specific model was used to predict how many times each user did a particular event on a repo at some hour.

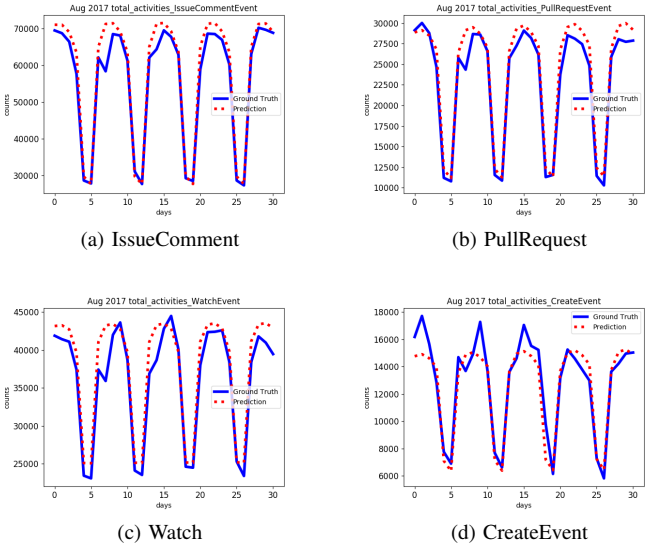
Each training sample for the pair-model was a representation of each user-repo pair's activity in a particular 2 day period. The feature vector was comprised of 336 values, corresponding to 24 hours * 14 Github events. Each value represented the percentage of global activity the user-repo pair performed for each of 24 hours for a particular Github event. The target vector for each sample was comprised of 24 values, each representing the proportion of overall activity that particular pair would engage in during each of the next 24 hours, for a particular event.

For example, suppose that there is a user-repo pair, called UR. Now, suppose that on some day, D, and in some hour, H, the pair UR performed 5% of all Push events done by all users. Furthermore, suppose that the number of all Pushes done by all user-repo pairs on day D, at hour H, was 100. This means that pair UR performed 0.05 * 100, or 5 Pushes on day D, at hour H.

1) Pair-Assignment Models Training and Validation Data: For training and validation, we used user-repo activity information spanning January 1st 2017 to July 31st 2017. We did not use all the available pairs for training, instead opting for the most active pairs in this 7 month period. We defined a pair as most-active if, on average, the user did more total events on the repo than averaging 1 per hour would yield from January 1st 2017 to July 31st, 2017. In total, this amounted to 410 pairs. We chose this subset of pairs for two reasons. The main purpose of the pair assignment model is finding the general hourly activity pattern for pairs. In our task there were many Github users who did very few events so a majority of the pairs don't contain enough useful hourly data to train a model. If we included them, our data would have contained a lot of barely active users and a large number of repos with very few events (e.g. a student's repo for a class). Then our model would have been inclined to under-predict user activity for the interesting cases of active users and well used repos. Also, including more user-repo-hour triplets than the current 1,485,840 samples, our training data would have exceeded the available memory.

For training, we used 151 days * 410 samples of user-repo pairs for 24 hours. The 151 days spanned January 1st to May 31st, 2017.

For validation, we used 61 days * 379 user-repo pairs, as some training pairs were inactive during the validation period. The range for the validation set was comprised of the 61 days spanning from June 1st 2017 to July 31st 2017.



(a) IssueComment (b) PullRequest

(c) Watch (d) CreateEvent

Fig. 1: Graphs of predicted event counts vs. ground truth for simulations (without ground truth) using Github only features

The pair model parameters were as follows. We used a fully connected neural network with 1 LSTM hidden layer with 500 units. We had a sequence length of 24 with each feature vector the count of the 14 events done in the previous day by the particular user-repo pair. There were 24 outputs which consisted of the percent of activities for the day done by the user to the repo in an hour. The output activation was linear. There was a fixed set of 11 weights (1 bias) used to output each of the 24 percentages using the TimeDistributed layer of Keras. There was a separate model created for each event. Each model was trained 100 epochs with a batch size of 1. We used the Adam optimizer and MAE loss function.

VI. Simulation Results

Experiments were done using just ground truth for the last week in July that began on the same day of the week as August 1, 2017 to drive initial predictions. Then the predictions were used to drive predictions for the second week, the predictions for the third week were based on those from the second week, etc. Experiments were also done using ground truth to drive the simulation throughout August. In that case, the actual counts for a week were used to predict the counts for the next week. Activity on Twitter and Reddit was included in learned models to see how it might improve the prediction of the daily count of events. In that case, we experimented both with and without including average sentiment features. The mean absolute percent error (MAPE) per event is reported. This was used in place of RMSE because the events of Github are not evenly distributed. Results are summarized using a weighted MAPE (WMAPE), where the weight is the percent of activities from the total number of activities for a particular event. Table II shows the number of events of each type in August 2017 and their percentage of the overall total number of events.

TABLE II: August 2017 Cyber Domain Event Occurrences

Event	Event Count	Frequency
Push	2,328,126	28.94%
Create	398,205	4.95%
Watch	1,118,66	13.90%
Issue Comment	1,732,18	21.53%
Pull Request	714,33	8.81%
Issues	631,428	7.80%
Fork	323,368	4.20%
Delete	239,044	2.97%
Pull Request Review Comment	456,259	5.67%
Gollum	39,986	0.49%
Commit Comment	27,930	0.34%
Public	450	0.006%
Member	7,202	0.09%
Release	26,408	0.33%
Total	8,043,588	100%

TABLE III: August 2017 No Ground Truth MAPE Results. GO stands for Github-Only, GTR WS means Github,Twitter, Reddit with Sentiment; and GTR NS means GTR without sentiment. C - Comment, PR - Pull request

Event	GO	GTR-NS	GTR-WS	Winner
Push	10.09%	23.69%	9.11%	GTR-WS
Create	6.65%	17.43%	9.09%	GO
Watch	5.41%	4.11%	4.29%	GTR-NS
Issue C	4.24%	4.21%	8.05%	GTR-NS
Pull Request	5.62%	3.89%	13.95%	GTR-NS
Issues	10.33%	10.95%	18.01%	GO
Fork	15.92%	12.22%	15.80%	GTR-NS
Delete	12.19%	20.09%	9.13%	GTR-WS
PR Review C	5.29%	5.68%	15.69%	GO
Gollum	13.49%	11.51%	7.98%	GTR-WS
Commit C	55.59%	38.42%	49.58%	GTR-NS
Public	96.44%	93.48%	47.74%	GTR-WS
Member	8.56%	10.01%	11.66%	GO
Release	8.57%	7.46%	8.55%	GTR-NS
WMAPE	7.829%	12.041%	10.117%	GO

A. Daily Count Model Evaluation

Complete results of the 3 Daily Model Sets are in [16], with limited space we focus on the no ground truth simulation. Figure 1 shows the predicted event counts for August 2017 compared to ground truth. Table III shows the results for each of the 3 ensembles when **not** using the ground truth vector as the input feature vector at each time step. Each model set used ground truth only for the first week's prediction. Ground truth was used to get the initial delta predictions which were added to the ground truth event counts. The predictions were then input for the second week's predictions, whose predictions were input for the third weeks predictions, etc.

The mean absolute percent error (MAPE) is shown in Table III. Table II shows the number and relative frequency of each event occurrence in the month of August 2017. As one can see it is quite imbalanced with Push making up 28.94% of all the events in August, while Gollum made up only 0.49% of all the events in August. Hence, the Weighted MAPE (WMAPE) of each of the 3 model set test results can be seen at the bottom of Table III. This weighted sum was calculated by multiplying the *Frequency of Occurrence* column in Table II with the corresponding event MAPE in Table III, and then summing each weighted event MAPE to produce the sum at the bottom of each column.

1) Daily Count Model Result Analysis: Overall, the model ensemble, GO (Github Only), had the lowest WMAPE compared to GTR (Github, Twitter and Reddit) NS (no sentiment) and GTR WS (with sentiment) both when feeding in the ground truth (7.610%) and when not feeding in the ground truth (7.829%). Recall that the GO ensemble is comprised of models trained only on the 14 Github events. The GTR-WS ensemble had the 2nd lowest WMAPE of 10.165% when feeding in the ground truth after each time step and 10.117% when not feeding in the ground truth at each time step. In third place was the GTR NS (no sentiment) ensemble, with a WMAPE of 12.481% when feeding in ground truth and 12.041% when not feeding in ground truth.

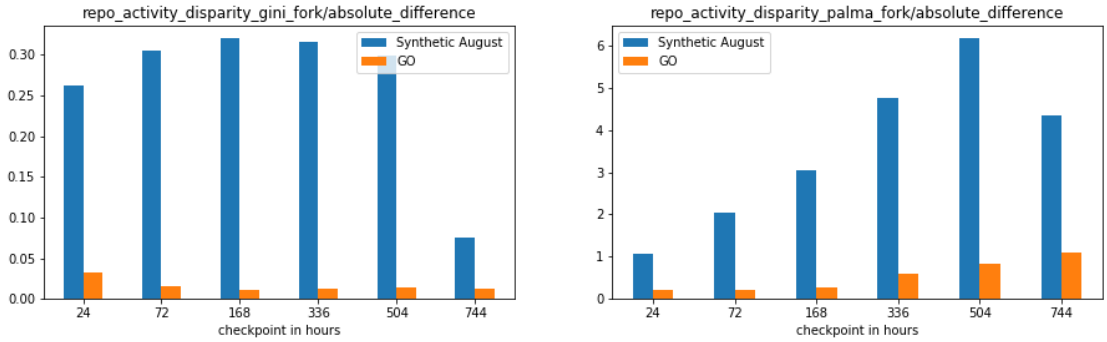
Based on these results, it would seem that in order to achieve the lowest overall WMAPE one should only train on Github features in order to predict Github features. However, for most Github events, having Twitter and Reddit features can help increase daily prediction accuracy.

Twitter and Reddit features are particularly useful (Table III) for the following. The the GTR-WS ensemble had a MAPE of 9.11% vs. the GO ensemble MAPE of 10.09% for Push. For Watch, GO's MAPE was 5.41%, but GTR-NS had an even lower MAPE of 4.11%. For Pull Request, GO had a MAPE of 5.62%, whereas GTR-NS had a lower MAPE of 3.89%. For Gollum, GO had a MAPE of 13.49%, whereas GTR-WS had a MAPE of 7.89%. For Commit Comment, GO had a MAPE of 55.59%, whereas GTR-NS had a MAPE of 38.42%. For Public, GO had a MAPE of 96.44% whereas GTR-WS had a MAPE of 47.74%. For Release, GO had a MAPE of 8.57%, whereas GTR-NS had a MAPE of 7.46%.

Clearly using the Reddit and Twitter features helps predict daily Github activity within the Cyber domain for events that could be influenced by other platforms. For example, the Push event could happen after some issues are raised in tweets or on Reddit. If they are important (sentiment matters) there may be an influence. A similar argument could be made for a Pull request and related Commit comment, except we see sentiment is not helpful for them. The Watch and Fork events are popularity based and would be influenced by social media, where mentions of a repo indicate popularity. Conversely, the Create, Issue Comment and Pull Request Review Comment events all reflect no influence by the other platforms. This seems reasonable as creating a repo and commenting on internal issues would likely have no impact from activities outside GitHub. Member events, which involve the addition or removal or permission changes of a repo collaborator, show no influence from external activity.

B. Pair assignment model results

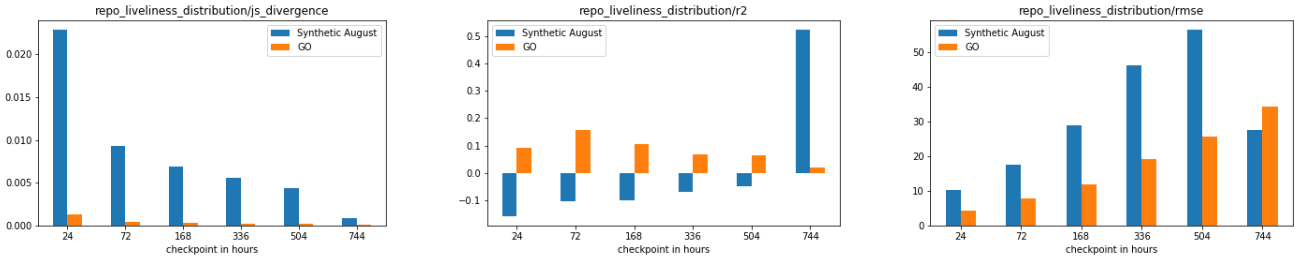
Assignment of user-repo pairs to an event in a particular hour (pair assignment) is a much harder task because of the highly imbalanced data and the huge total size of the data.



(a) Absolute Difference Gini (lower better)

(b) Absolute Difference Palma (lower better)

Fig. 2: Plots of repo level activity vs. Synthetic August for fork event. Values are displayed at 1 day, 3 days, 1, 2, 3 weeks and the end of the month. Differences in Gini coefficient and Palma ratio

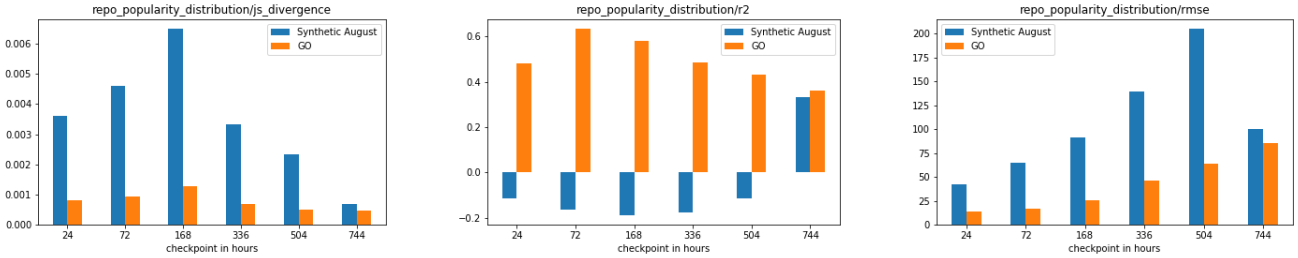


(a) js divergence (lower better)

(b) r2 (higher better)

(c) RMSE (lower better)

Fig. 3: Plots of fork activity across repos vs. Synthetic August using js divergence, r2 and RMSE (Root Mean Square Error)

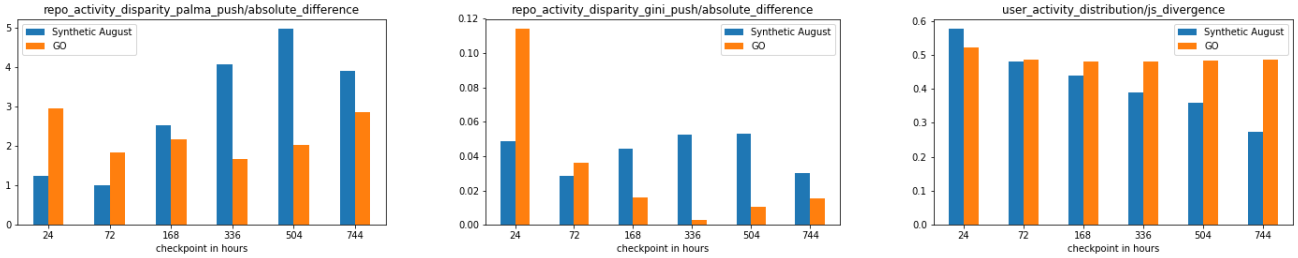


(a) js divergence (lower better)

(b) r2 (higher better)

(c) RMSE (lower better)

Fig. 4: Plots of repo popularity distribution measured by watch event vs. Synthetic August using js divergence, r2, and RMSE

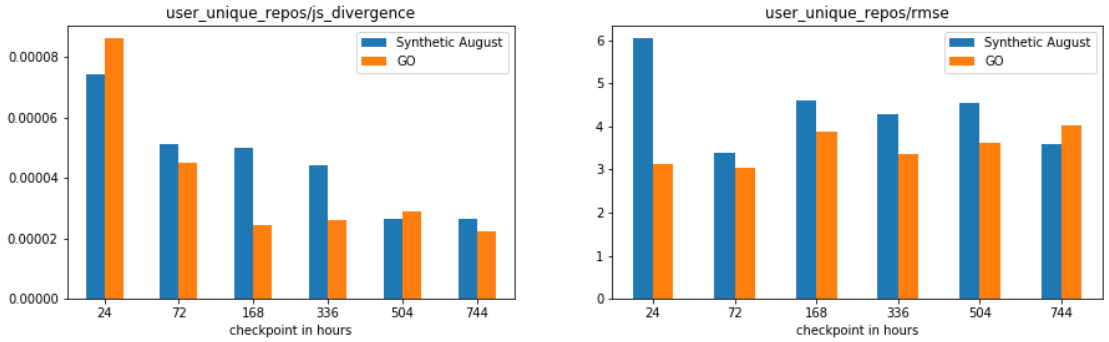


(a) Absolute Difference (lower better)

(b) Absolute Difference (lower better)

(c) js divergence (lower better)

Fig. 5: Plots of push event Palma ratio and Gini coefficient comparison vs. Synthetic August, plus user activity distribution



(a) js divergence (lower better)

(b) RMSE (lower better)

Fig. 6: Plots of user unique repos comparison vs. Synthetic August using js divergence and RMSE

All users who did events in the last 2 weeks of July to some repo were simulated (this was 962,404 pairs).

We predicted GitHub activities across the days in August 2017 and compared the predictions with ground-truth data. We used the following measures to evaluate performance from the code repository [22].

The Jensen-Shannon (js) divergence [15] is a finite measure applicable to finite random variables. It can be used to quantify how 'distinguishable' the predicted distribution is from the ground truth distribution. The coefficient of determination (r2 metric) [20] is a statistical measure that provides information on the goodness of fit of a model. The rank-biased overlap (RBO) [25] is a similarity measure that is appropriate for indefinite rankings.

For comparison purposes, we shifted the activities from June 27, 2017 to July 27, 2017 onto August and called this Synthetic August. To capture daily patterns, Synthetic August starts on a Tuesday, when August 1, 2017 falls.

A set of figures comparing our predictions, using the GitHub only daily counts, against Synthetic August with analysis were created [16]. Results, checkpoints, are shown for 1 day, 3 days, 1, 2, 3 weeks, and the full month in hours.

The repo activity disparity Gini/Palma fork measures the absolute difference of the Gini coefficient and Palma ratio for fork events on repos between predictions and ground truth. Figure 2 shows our fork forecasting per repo is much better than the shifting baseline.

The repo liveliness distribution measures distribution of fork events across repos. In Figure 3 the repo liveliness distribution shows our fork events forecasting is much better than the shifting baseline for at least 3 weeks of forecasting.

The repo popularity distribution measures the distribution of watch events across repos in Figure 4. As a repo popularity measurement, it shows our watch event forecasting is much better than the shifting baseline of Synthetic August for at least 3 weeks.

The distribution of pushes across repos is shown to be captured reasonably well in Figure 5. The user activity distribution shows the distribution over user activity for all users. It shows we capture the general activity of users better than the shifted baseline in the first 3 days predictions.

In Figure 6 user unique repos measures the number of unique repos to which each user contribute. Our simulation method has better performance for most checkpoints in this user level measurement.

Overall these results show the current pair assignment model works well on user/repo level popularity distribution measurement and user engagement measurement. It does tail off in accuracy as it uses its own predictions to make future predictions for some cases, which is not unexpected. More evaluations can be found in [16].

C. Additional Experiments

As an additional experiment, we predicted Github daily event counts using LightGBM models. Our parameters were as follows. For boosting type, we used "gbdt". The objective parameter was set to "regression". The metrics used were "L2" and "L1". We used 31 leaves and a learning rate of 0.05. Our feature fraction was 0.9, the bagging fraction was 0.8, and the bagging frequency was 5. The number of boost rounds was 100 and lastly, we used 5 early stopping rounds.

Our proposed LSTM daily count model outperformed LightGBM for Fed-in Ground Truth in terms of MAPE (7.610% VS 8.916%) and was much better in the No Ground Truth scenario (7.829% VS 25.94%). When using its own previous predictions Ground Truth LightGBM drifted from ground truth quickly to be quite inaccurate. Detailed results are available in [16].

VII. Discussion

Training and simulation using daily count models is really fast on CPU's. One month of GitHub simulation for daily counts takes minutes. User level pair assignment is considerably more time-consuming. In the pair assignment model training we used one Nvidia GTX 1080ti GPU and training used 16 hrs. In simulation we completed predictions for August in 3hr with that GPU.

The percentage prediction in our pair assignment model is important for the imbalanced dataset. Previous experiments, which predicted number of activities by pairs in an hour,

were less accurate after subsampling to contain just the most-active pairs. The difference is the ground truth values when predicting activities per day are big numbers compared to the predicted percentage of activities. The figures show that at the hourly level our predictions are often not far from ground truth on this complex, big data problem. An LSTM model proved to be useful to capture temporal patterns for both big and small numbers.

A limitation on the current simulation system is only user-repo pairs (users active on a repo) that exist in past data can appear. The current system can easily be extended to new users/repos to get daily counts by adding features from them into our time-series data set and re-training. However, how to measure the performance at the user level is unclear since matching predicted new users to the correct new user IDs for comparison with future ground truth is daunting.

VIII. CONCLUSIONS

Our decomposition approach using an ensemble of LSTM models to predict the number of daily events on GitHub followed by another LSTM model to predict what user does an event to what repo at what time was found to provide solid simulation performance and be scalable. Results are reasonably close to ground truth on this big data simulation of human activity. Importantly, it was shown that activity on other social media platforms, Twitter and Reddit, can positively influence predictive performance for Github events subject to external influence.

We applied our temporal learning models to do long-term simulation with over 2 million Github users and over 400,000 repos with reasonable time (hours) to do a forecast. We found most Github events have their own weekly pattern, which can enable efficient overall daily event count prediction. Our simulator can handle highly imbalanced data. As more predictions are made on top of earlier predictions with errors, accuracy degrades. This is a focus for future work and could be addressed by blending predictions at different granularities, as well as direct predictions of future activities from an initial set of activity information. Generally, early in the simulation the performance is strong on this big data problem of predicting who does what event to what repo per hour.

REFERENCES

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L Fei-Fei, and S Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016.

[2] Ioannis Arapakis, Berkant Barla Cambazoglu, and Mounia Lalmas. On the feasibility of predicting popular news at cold start. *Journal of the Association for Information Science & Technology*, 68(5):1149 – 1164, 2017.

[3] Github Archive. Gh archive. http://www.gharchive.org/, 2018.

[4] Mohammad Assaad, Romuald Boné, and Hubert Cardot. A new boosting algorithm for improved time-series forecasting with recurrent neural networks. *Information Fusion*, 9(1):41 – 55, 2008. Special Issue on Applications of Ensemble Methods.

[5] N. Chawla, D. Cieslak, L. Hall, and A. Joshi. Automatically countering imbalance and its empirical relationship to cost. *Data Mining and Knowledge Discovery*, 17:225–252, 2008.

[6] J.F. Díez-Pastor, J.J. Rodríguez, C. García-Osorio, and L.I. Kuncheva. Random balance: Ensembles of variable priors classifiers for imbalanced data. *Knowledge-Based Systems*, 2015.

[7] J.F. Díez-Pastor, J.J. Rodríguez, C.I. García-Osorio, and L.I. Kuncheva. Diversity techniques improve the performance of the best imbalance learning ensembles. *Information Sciences*, 325:98 – 117, 2015.

[8] T. Fernando, S. Denman, S. Sridharan, and C. Fookes. Soft + hard-wired attention: An LSTM framework for human trajectory prediction and abnormal event detection. *Neural Networks*, 108:466 – 478, 2018.

[9] Jesse Hoey, Tobias Schröder, Jonathan Morgan, Kimberly B. Rogers, Deepak Rishi, and Meiyappan Nagappan. Artificial intelligence and social simulation: Studying group dynamics on a massive scale. *Small Group Research*, 49(6):647 – 683, 2018.

[10] D.-S. Huang, X.-P. Zhang, and G.-B. Huang. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In Hui Han, Wen-Yuan Wang, and Bing-Huan Mao, editors, *Advances in Intelligent Computin*, volume 3644 of *Lecture Notes in Computer Science*, pages 878–887. Springer Berlin Heidelberg, 2005.

[11] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T-Y Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, and et. al. S. Bengio, editors, *Advances in Neural Information Processing Systems 30*, pages 3146–3154. Curran Associates, Inc., 2017.

[12] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154, 2017.

[13] Young Bin Kim, Jun Gi Kim, Wook Kim, Jae Ho Im, Tae Hyeong Kim, Shin Jin Kang, and Chang Hun Kim. Predicting fluctuations in cryptocurrency transactions based on user comments and replies. *PLoS ONE*, 11(8):1 – 17, 2016.

[14] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.

[15] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, Jan 1991.

[16] R. Liu, F. Mubang, L.O. Hall, A. Iamnitchi, and J. Skvoretz. Simulating github events for cyber-security related repos. Tr: ISL119: http://www.cse.usf.edu/˜ lohall/isl119.pdf, Univ of South Florida, 2019.

[17] Ilias N. Lymperopoulos and George D. Ioannou. Understanding and modeling the complex dynamics of the online social networks: a scalable conceptual approach. *Evolving Systems*, 7(3):207–232, Sep 2016.

[18] Moreno Mancosu and Giuliano Bobba. Using deep-learning algorithms to derive basic characteristics of social media users: The brexit campaign as a case study. *PLoS ONE*, 14(1):1 – 20, 2019.

[19] C Meiring, A Dixit, S Harris, N.S. MacCallum, D.A. Brealey, P.J. Watkinson, A. Jones, S. Ashworth, R. Beale, S.J. Brett, M Singer, and A Ercole. Optimal intensive care outcome prediction over time using machine learning. *PLoS ONE*, 13(11):1 – 19, 2018.

[20] N. J. D. Nagelkerke. A note on a general definition of the coefficient of determination. *Biometrika*, 78(3):691–692, 1991.

[21] V. S. Pagolu, G. Reddy, K.N.and Panda, and B. Majhi. Sentiment analysis of twitter data for predicting stock market movements. In *Signal Processing, Communication, Power and Embedded System (SCOPES), 2016 International Conference on*, pages 1345–1350. IEEE, 2016.

[22] PNNL. Pacific northwest national laboratory, socialsim. https://github.com/pnnl/socialsim, 2018.

[23] P Singer, F Flöck, C Meinhart, E Zeitfogel, and M Strohmaier. Evolution of reddit: From the front page of the internet to a self-referential community? In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, pages 517–522, New York, NY, USA, 2014. ACM.

[24] Huiwen Wang, Shan Lu, and Jichang Zhao. Aggregating multiple types of complex data in stock market prediction: A model-independent framework. *Knowledge-Based Systems*, 164:193 – 204, 2019.

[25] W. Webber, A. Moffat, and J. Zobel. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4):1–38, November 2010.

[26] Christopher Xie, Alex Tank, Alec Greaves-Tunnell, and Emily Fox. A unified framework for long range and cold start forecasting of seasonal profiles in time series. *arXiv preprint arXiv:1710.08473*, 2017.