Popularity Prediction for Social Media over Arbitrary Time Horizons

Daniel Haimovich
Core Data Science, Meta Platforms
danielha@fb.com

Dima Karamshuk
Core Data Science, Meta Platforms
karamshuk@fb.com

Thomas J. Leeper
Core Data Science, Meta Platforms
thomasleeper@fb.com

Evgeniy Riabenko
Core Data Science, Meta Platforms
riabenko.e@gmail.com

Milan Vojnovic
Department of Statistics, LSE
m.vojnovic@lse.ac.uk

ABSTRACT

Predicting the popularity of social media content in real time requires approaches that efficiently operate at global scale. Popularity prediction is important for many applications, including detection of harmful viral content to enable timely content moderation. The prediction task is difficult because views result from interactions between user interests, content features, resharing, feed ranking, and network structure. We consider the problem of accurately predicting popularity both at any given prediction time since a content item's creation and for arbitrary time horizons into the future. In order to achieve high accuracy for different prediction time horizons, it is essential for models to use static features (of content and user) as well as observed popularity growth up to prediction time.

We propose a feature-based approach based on a self-excited Hawkes point process model, which involves prediction of the content's popularity at one or more reference horizons in tandem with a point predictor of an effective growth parameter that reflects the timescale of popularity growth. This results in a highly scalable method for popularity prediction over arbitrary prediction time horizons that also achieves a high degree of accuracy, compared to several leading baselines, on a dataset of public page content on Facebook over a two-month period, covering billions of content views and hundreds of thousands of distinct content items. The model has shown competitive prediction accuracy against a strong baseline that consists of separately trained models for specific prediction time horizons.

PVLDB Reference Format:
Daniel Haimovich, Dima Karamshuk, Thomas J. Leeper, Evgeniy Riabenko, and Milan Vojnovic. Popularity Prediction for Social Media over Arbitrary Time Horizons. PVLDB, 15(4): 841-849, 2022.
doi:10.14778/3503585.3503593

1 INTRODUCTION

Popularity prediction can be a useful system component for management of user-generated content in online platforms. For example, in content moderation platforms, such as the one used by Facebook [46], potentially harmful content items are flagged either by users or machine learning filters. These flagged content items are examined either automatically or are placed into a queue for manual review. To make sure that the most important posts are seen first by the reviewers, a content moderation platform may take into account their virality. Other applications of popularity prediction include optimizing content distribution, e.g. for video streaming [44]. In these applications, popularity prediction is used to prioritize content item processing with the goal to improve the quality of user experience. These applications require accurate and scalable methods for popularity prediction.

State of the art popularity prediction algorithms are accurate but mostly do not scale to handle large-scale social media content workload because they typically have per-content-item computation cost that increases linearly with the number of observed events (see discussion in Sec. 4). While different popularity prediction methods have been proposed, e.g., [10, 12, 39, 51], they do not satisfy at least one of the following design considerations for application at a planetary scale: (a) prediction of the number of views acquired up to a future time horizon, not just a classification of virality, (b) prediction method has a low computation and memory complexity, (c) prediction method can generate accurate predictions for any given time horizon, or (d) prediction method leverages both static features (e.g. content author and content item features) and temporal features (observed up to given prediction time). We discuss this further as follows.

First, some work in the information cascades literature adopts a classification-based approach to defining virality (e.g., cascades smaller/larger than a given size; cascades doubling in a given time frame), these have limited use for applications that require comparison or prioritization among likely popular items. We focus here on approaches that provide *real number predictions of popularity*.

Second, while low computational costs and memory constraints may not be prohibitive for offline or adhoc demonstration, scalability of this sort can be especially relevant when making predictions in real time. This is particularly true when evaluating large numbers of content items in parallel. To the best of our knowledge, only some previous work focused on the design of popularity prediction methods with the *scalability as the main design goal* for applications in large-scale online platforms. Specifically, [44] proposed a method for video popularity prediction that uses a constant state per content item. Some prediction methods, e.g., Reinforced Poisson Process model [40] and SEISMIC [51], may be deemed to be computationally simple, but they still do not satisfy our target scalability constraints (see Sec. 4 for details). Other methods, such as HIP [39],

Proceedings of the VLDB Endowment, Vol. 15, No. 4 ISSN 2150-8097.
doi:10.14778/3503585.3503593

have scalability issues at prediction time and do not address the requirement of combining static and temporal features.

Third, flexibility in prediction time and the prediction time horizon are desirable. A simple approach to popularity prediction might involve a point-based prediction of growth up to a fixed content age based on features observable at a single point in time (e.g., content creation), but that model would not update in response to new information about the content (e.g., temporal features) and is limited in being able to predict for a fixed time horizon. Supporting *multiple time horizons* by using one model per horizon disallows predictions for previously unseen horizons. Alternatively, many recent approaches to popularity prediction aim to provide estimates of total cascade size (at infinite time), limiting their utility for forecasting the "urgency" of cascade growth. Once again considering the content moderation application, queries about expected popularity may be made at multiple points in the content lifecycle from creation onward. In cases where content is removed—by platforms or by users themselves—cascades are truncated, making the evaluation of prediction accuracy for only a fixed prediction horizon difficult or impossible. Such truncated cascades are also unusable as training data in fixed or infinite horizon models.

Finally, the last consideration — to *leverage both static and temporal features* — is important to ensure high prediction accuracy throughout the content lifecycle. Content views can result from complex interactions between content resharing and engagement, time zone use patterns, feed ranking algorithms, and organic features of the content and social network structure, so predictions benefit from insight into as much of this information as possible to the extent that the signals can be efficiently incorporated into the model. Clearly, any approach that relies only on event histories are likely to be inaccurate or unusable at early content ages. Similarly, approaches that rely only on static features will not adapt to new information provided by content engagement, resharing, and views.

In this paper, we propose a new popularity prediction model that (a) provides real number predictions, (b) has constant computation complexity and uses a small space per content item, (c) can produce predictions for any given prediction time horizon specified at any given prediction time, and (d) leverages both static and temporal features. The model is based on a self-excited Hawkes point process model with exponentially decaying intensity, combined with prediction of model parameters by using both static and temporal features. This combination allows us to reduce the computation complexity of making predictions to constant time for any cascade size, but benefit from the analytically tractable estimators of the popularity over arbitrary future time horizons.

Specifically, our basic prediction model uses only two point predictors, one for prediction of the number of points over a fixed reference time horizon (this is a hyper-parameter of the model) and one for the effective growth exponent which reflects the point process growth rate over time. This allows to use any point predictor developed and trained for making predictions for a specific time horizon, and then generalize this to support predictions for any given prediction time horizon by adding one extra point predictor. We also propose an extension that allows combining several point predictors of content view counts at different reference time horizons, increasing prediction accuracy while still using only a constant space per content item.

We demonstrate the accuracy and feasibility of our prediction method using a large-scale dataset of public Facebook posts over a two-month period. Our results demonstrate that high prediction accuracy can be achieved over different prediction time horizons, by using a few point predictors and that our models achieve performance that is comparable or better than a strong baseline that consists of using predictors designed and trained for specific prediction time horizons.

In Section 2 we discuss related work. Section 3 lies down a framework for making predictions using self-excited point process models. Section 3.2 defines our prediction models. Section 4 provides a discussion of our results. Experimental results are presented in Section 5. In Section 6, we provide concluding remarks. Appendix is provided in the companion technical report [23].

2 RELATED WORK

Early work on predicting the popularity of online content considered various classification and regression models for fixed prediction time horizons using different types of features [12, 43]. Much work has been devoted to understanding how information spreads in online social networks [1, 21, 34, 48] and the role of social networks for information diffusion [5, 14]. We refer the reader to surveys on web content popularity prediction [37] and information cascade analysis [52]. Models have been proposed for both popularity prediction (shares, views) and prediction of the number of users reached in an information cascade. We distinguish feature based models, generative models, and deep learning models, which we discuss in turn.

Feature based methods. Feature based prediction models use different types of features, including *temporal features* (observation time, creation time, first view time), *structural features* (cascade graph), *user-item features*, and *content features*. Several works considered prediction of an information cascade size by using information observed over an initial time period [2, 6, 7, 15, 22, 28, 33, 45]. Classification models [12, 16, 24, 25, 28] and regression models [4, 28, 43, 45] have been studied for prediction of information cascade sizes and prediction of occurrence of activity bursts [13, 42, 47]. Temporal features have been found to be important for prediction of content popularity and information sharing [3, 12, 43]. Using network structural features is often not considered scalable [42].

Generative models. Generative models assume events are generated according to a stochastic point process, which includes simple Poisson processes, survival analysis models, Hawkes point processes, and epidemic models. Different self-excited point process models have been used, including cascades of Poisson processes [41], reinforced Poisson processes [40], and Hawkes point processes and their variations [26, 36, 39, 51]. Another class of models are multi-dimensional Hawkes processes, which allow to model different types of events and their mutual excitation [31, 49, 53, 54]. Finally, epidemic models have also been used for modelling information diffusions [27, 38]. Most similarly, Hawkes point processes with exponentially decaying intensity were used for feature generation fed into a neural network predictor for predicting infinite-horizon watch time of Facebook videos [44]. Our work has similarities with these previous works in using a generative model and differs in emphasizing both scalability and making popularity predictions for arbitrary time horizons as the main design goals.

Deep learning models. Deep learning models use neural networks as prediction models or for learning numerical vector representation (embeddings) of temporal or structural features for popularity prediction. Several works extended self-excited point process models with neural networks, including DeepHawkes [8], Neural-Hawkes [35], and SIR-Hawkes [38]. Neural networks have been used for representations of event histories [19], incidence curves [50], information diffusion networks [29, 30], fusion of content and temporal features [32], representations of structural and temporal information [11], and social network interactions [9]. Deep learning based models for popularity prediction are not scalable for our intended scenarios, as they typically require inputs that grow linearly in the number of past events and are complex or expensive to use for making predictions over arbitrary time horizons.

3 METHODOLOGY

In this section, we first present some results on self-excited point processes in Section 3.1, which are used to define our prediction method in Section 3.2.

3.1 Self-excited point processes

3.1.1 Background. We consider generative models defined as point processes, with points representing occurrance times of view events of a content item. A realization of a *point process* on $\mathbb{R}_+$ is a sequence of points $0 \leq T_1 \leq T_2 \leq \cdots$ that can be equivalently represented by a counting variable $N(t)$ defined as the number of points in $[0, t)$, i.e. $N(t) = \sum_{i\geq 1} \mathbf{1}_{\{0\leq T_i<t\}}$, for any $t \in \mathbb{R}_+$. A *stochastic point process* has the *stochastic intensity function* defined by

$$\lambda(t) = \lim_{\epsilon \downarrow 0} \frac{\mathbb{E}[\, N(t+\epsilon) - N(t) \,|\, \mathcal{F}_t]}{\epsilon},$$

where $\mathcal{F}_t$ is the history of the point process up to time t. Intuitively, we can think of $\lambda(t)$ as the conditional probability that there is a point in $[t, t+\epsilon)$, conditional on the history $\mathcal{F}_t$, for small ϵ.

A *Hawkes point process* is defined by the stochastic intensity function

$$\lambda(t) = \lambda_0(t) + \sum_{i=1}^{\infty} \phi_{Y_i}(t - T_i)\mathbf{1}_{\{0\leq T_i<t\}},$$

where λ_0 and ϕ_y are given functions and $y \in \mathbb{R}_+$. Here $Y_0, Y_1, \ldots$ are assumed to be independent and identically distributed random variables (referred to as *marks*) according to distribution F_Y, which are independent of the points $T_1, T_2, \ldots$. Following standard definition, we assume that $\phi_y(x)$ is of the form $\phi_y(x) = y\phi(x)$, where $\phi(x)$ is a *kernel function.* Under this assumption, Y_i is the size of a jump in the stochastic intensity function.

Let μ be the expected contribution of a point to the value of the stochastic intensity function defined by $\mu = \mathbb{E}_{Y\sim F_Y}\left[\int_0^\infty \phi_Y(t)dt\right]$. We assume that $\mu < 1$, which ensures stability of the point process.

The framework of self-excited point processes accommodates different instances of stochastic point processes. Here we consider two notable examples.

Exponentially decaying kernel. The Hawkes point process *with exponentially decaying intensity* is defined by the kernel function

$$\phi(x) = e^{-\beta x}, \tag{1}$$

where $\beta > 0$ is a parameter and assuming that $\lambda_0(t) = \lambda(0)\phi(t)$, for some initial value $\lambda(0) > 0$. In this case, we have

$$\lambda(t) = \lambda(0)e^{-\beta t} + \sum_{i=1}^{\infty} Y_i e^{-\beta(t-T_i)}\mathbf{1}_{\{0\leq T_i<t\}}.$$

We will use the change of variable such that $Y_i = \beta Z_i$ for a random variable Z_i with distribution G. We may interpret Z_i as a population size (neighbors of a node in a social network) and β as a rate parameter (rate of interactions between nodes in a social network). Let ρ_r denote the r-th moment of Z_i, i.e. $\rho_r = \int_0^\infty z^r dG(z)$. Note that $\mathbb{E}[Y_1] = \beta\rho_1$ and $\mu = \rho_1$.

We will later discuss that Hawkes point processes with exponentially decaying intensity have certain desirable properties for scalable popularity prediction over arbitrary time horizons.

Power-law decaying kernel. Another commonly used kernel function is the *power-law kernel* defined as

$$\phi(x) = \begin{cases} \phi(0) & \text{if } 0 \leq x \leq \tau, \\ \phi(0)\left(\frac{\tau}{x}\right)^{1+\theta} & \text{if } x > \tau, \end{cases} \tag{2}$$

where $\phi(0) > 0$, $\tau > 0$ and $\theta > 0$ are parameters. For instance, this kernel was used in [51] and [39] to model information cascades.

The framework presented in this section has the following interpretation in the context of popularity prediction of content items. We may interpret each point as a content view event that excites subsequent content view events. For the Hawkes point process with exponentially decaying kernel, the random variable Z_i can be interpreted as the number of potential users that can be reached resulting from the content view event at time T_i. The parameter β is the rate at which users consume content. The parameter μ is the expected number of subsequent content view events triggered by a content view event. The kernel function models the time-decay of the stochastic intensity function components triggered by content view events, capturing their diminishing influence over time.

3.1.2 Counts over future time horizons. For popularity prediction for a content item, we are interested in predicting the number of content view events over a given time horizon at a given prediction time, having observed the history of the content views up to the prediction time. Using the framework introduced in previous section, given a prediction time s and a time horizon up to time instance $t > s$, we are interested in predicting the value of $N(t) - N(s)$, having observed the history $\mathcal{F}_s$.

Infinite time horizon. For any stable Hawkes point process, the conditional expected number of points over an *infinite* time horizon originating at a time instance $s \geq 0$, conditional on the history $\mathcal{F}_s$, is given as

$$\lim_{t\to\infty} \mathbb{E}[\, N(t) - N(s) \,|\, \mathcal{F}_s] = \frac{1}{1-\mu} \lim_{t\to\infty} \Lambda(s,t) \tag{3}$$

where

$$\Lambda(s,t) = \Lambda_0(t) - \Lambda_0(s) + \sum_{i\geq 1} y_i \left(\Phi(t-T_i) - \Phi(s-T_i)\right)\mathbf{1}_{\{0\leq T_i<s\}},$$

and Λ_0 and Φ are the primitive functions of λ_0 and ϕ, respectively, i.e. $\Lambda_0(x) := \int_0^x \lambda_0(u)du$ and $\Phi(x) := \int_0^x \phi(u)du$. Here $\Lambda(s,t)$ is the conditional expected number of points in $[s,t]$, induced by the intensity function λ_0 and the intensity function components excited by points in $[0,s]$, conditional on the history $\mathcal{F}_s$.

For the Hawkes point process with exponentially decaying intensity, the expression in (3) boils down to

$$\lim_{t\to\infty} \mathbb{E}[\,N(t) - N(s) | \mathcal{F}_s] = \frac{1}{\alpha}\lambda(s) \quad (4)$$

where $\alpha = \beta(1-\rho_1)$. For the reasons explained shortly, we refer to α as *the effective growth exponent.* Note that (4) is a function only of the intensity $\lambda(s)$ and the effective growth exponent α.

Arbitrary time horizons. It is not tractable to have an explicit formula for the conditional expected count over an *arbitrary* time horizon—which is our objective—for all Hawkes point processes. However, we offer the following bounds.

Proposition 3.1. *For any stable Hawkes point process, for every* $0 \le s \le t$, *we have*

$$\Lambda(s,t) \le \mathbb{E}[\,N(t) - N(s) | \mathcal{F}_s] \le \frac{1}{1-\mu}\Lambda(s,t).$$

Proof of this proposition is given in Appendix A.5 [23]. Note that for any fixed value of $\mu < 1$, $\mathbb{E}[\,N(t) - N(s) | \mathcal{F}_s]$ is within a constant factor of $\Lambda(s,t)$. Intuitively, the bounds in Proposition 3.1 are tighter the nearer the value of μ is to zero (small expected number of points excited by a point).

Arbitrary time horizons for exponential kernel. For the Hawkes point process with exponentially decaying intensity, we can characterize the conditional expected number of points over an *arbitrary* time horizon, conditional on the observed history up to a time instance, as stated in the following proposition. This is a key proposition for defining our prediction model in Section 3.2.

Proposition 3.2. *For the Hawkes point process with exponentially decaying intensity, for every* $0 \le s \le t$, *we have*

$$\mathbb{E}[\,N(t) - N(s) | \mathcal{F}_s] = \frac{1}{\alpha}\left(1 - e^{-\alpha(t-s)}\right)\lambda(s). \quad (5)$$

Proof is given in Appendix A.4 [23]. From (5), observe that the conditional expected count of points converges exponentially to its limit value with rate α, which provides a justification for referring to α as the effective growth exponent.

The effective growth exponent α admits the following intuitive interpretation. Note that we can write (5) as

$$\mathbb{E}[\,N(t) - N(s) | \mathcal{F}_s] = \mathbb{E}[\,N(+\infty) - N(s) | \mathcal{F}_s]\,(1 - e^{-\alpha(t-s)}).$$

For any given $\gamma \in (0,1)$, let τ_γ be the length of the time horizon at which the conditional expected count is equal to factor γ of its limit value. It is easy to derive that

$$\tau_\gamma = c_\gamma \frac{1}{\alpha}, \quad (6)$$

with constant $c_\gamma = \log(1/(1-\gamma))$. Hence, we can interpret the reciprocal value of α as a *characteristic time.*

A notable property of Hawkes point processes with exponentially decaying intensity is that $\Lambda(s,t)$ and $\mathbb{E}[N(t) - N(s) \mid \mathcal{F}_s]$ depend on the history $\mathcal{F}_s$ only through the value of the stochastic intensity $\lambda(s)$ at time instance s. This can be leveraged for making scalable predictions by using low-complexity estimators of $\lambda(s)$. This stands in contrast to other Hawkes point processes, which require using more expensive computations.

3.2 Prediction method

In this section we present our model for predicting popularity of social media items over arbitrary time horizons. The model is designed with *scalability* as the main design requirement. The idea behind our approach is to use a Hawkes model with parameters determined by a learned mapping between a vector representation of the content features and point process parameters. This approach allows us to reduce the computation complexity of making predictions to constant time with respect to the observed events in the cascade $N(s)$, and benefit from the analytically tractable estimators of the popularity over arbitrary future time horizons.

3.2.1 Prediction model. The model is based on the following expression for the conditional expected number of points up to future time $s+\delta$, for given prediction time s and prediction time horizon $\delta \ge 0$, and an arbitrarily fixed *reference horizon* $\delta^* > 0$,

$$\mathbb{E}[\,N(s+\delta) | \mathcal{F}_s] = N(s) + \frac{1-e^{-\alpha\delta}}{1-e^{-\alpha\delta^*}}\left(\mathbb{E}\left[\,N(s+\delta^*)\middle|\mathcal{F}_s\right] - N(s)\right)$$

which follows from Proposition 3.2.

The expression above has two unknown parameters: (a) the conditional expected number of points at the reference time horizon, $\mathbb{E}[\,N(s+\delta^*) | \mathcal{F}_s]$, and (b) the effective growth exponent α. These unknown parameters need to be inferred for any given features of a content item by using training data.

Let $\hat{N}(\delta;s)$ denote the predictor of $N(s+\delta)$ given history $\mathcal{F}_s$ and $\hat{\alpha}$ denote the predictor of α. Let us also use a logarithmic transformation of the prediction variable by defining $Y(\delta;s) = \log(\hat{N}(\delta;s) - N(s))$. Then, we can write

$$Y(\delta;s) = Y(\delta^*;s) + \log\left(\frac{1-e^{-\hat{\alpha}\delta}}{1-e^{-\hat{\alpha}\delta^*}}\right) \quad (7)$$

with $Y(\delta^*;s)$ and $\hat{\alpha}$ being values of two predictors defined as follows. The first predictor is for the log-transformed number of points over the reference time horizon,

$$Y(\delta^*;s) = f(x, \tau(\mathcal{F}_s); \theta), \quad (8)$$

where x is the vector of static features and $\tau(\mathcal{F}_s)$ is the vector of temporal features derived from $\mathcal{F}_s$ of the content item, and θ is the regression model parameter. The second predictor is for the effective growth exponent:

$$\hat{\alpha} = g(x, \tau(\mathcal{F}_s); \theta'), \quad (9)$$

where θ' is the regression model parameter. We use temporal features $\tau(\mathcal{F}_s)$ that can be computed in constant time, which is required by our scalability requirement.

In summary, our prediction method amounts to predicting popularity of a content item at time $s+\delta$, at prediction time s, and any given prediction time horizon δ, by using equation (7) with $Y(\delta^*;s)$

and $\hat{\alpha}$ defined by functions of the static feature vector x and the temporal feature vector $\tau(\mathcal{F}_s)$ as given in (8) and (9), respectively.

3.2.2 Training details. Functions f and g, in (8) and (9) respectively, can be implemented by using standard machine learning algorithms. In our evaluations in Section 5 we used gradient boosted decision trees, trained independently for f and g. For training parameters of f, we use (x_i, y_i) as training examples where x_i is the vector of static and temporal features and y_i is the number of points observed over the reference time horizon for training example i. Similarly, for training parameters of g, we use (x_i, y_i) as training examples with x_i defined as before and y_i defined to be an estimate of the effective growth exponent for content item i. We discuss estimators of the effective growth exponent in Section 3.2.4.

A notable property of our prediction model is that it requires using only two point predictors, while allowing for making predictions for any given prediction time horizon. With scalability in mind, we consider point predictors which can be computed in constant time with respect to the observed history of cascade. Notice that the predicted value for the length of prediction horizon $\delta = \delta^*$ is equal to $Y(\delta^*; s)$. In this case, our predictor is guaranteed to be as accurate as the predictor optimized for the reference time horizon δ^*. For $\delta \neq \delta^*$, the predictor may have a worse accuracy than a predictor optimized for the time horizon δ. We will evaluate this empirically in Section 5, where we will see that the proposed method can achieve competitive performance to predictors optimized for specific prediction time horizons.

3.2.3 Combining multiple point predictors. We can extend our prediction method to using one or more point predictors, which may increase prediction accuracy. Let $\hat{N}(\delta_1^*; s), \ldots, \hat{N}(\delta_m^*; s)$ be point predictors for given values of reference horizons $\delta_1^* < \delta_2^* < \cdots < \delta_m^*$, for some given $m \geq 1$. The prediction method is defined by combining outputs of these point predictors.

We consider two different predictors that combine outputs of point predictors by using different combining functions.

Arithmetic mean aggregation. The first predictor combines outputs of different point predictors $(\hat{N}(\delta_1^*; s), \ldots, \hat{N}(\delta_m^*; s))$ by using the arithmetic mean aggregation, which amounts to the following predictor for the log-transformed prediction variable:

$$Y(\delta; s) = \log\left(\frac{1}{m}\sum_{i=1}^{m}\frac{1}{1-e^{-\hat{\alpha}\delta_i^*}}e^{Y(\delta_i^*; s)}\right) + \log\left(1-e^{-\hat{\alpha}\delta}\right).$$

Geometric mean aggregation. The second predictor combines outputs of point predictors $(\hat{N}(\delta_1^*; s), \ldots, \hat{N}(\delta_m^*; s))$ by using the geometric mean aggregation, which amounts to the following predictor for the log-transformed prediction variable:

$$Y(\delta; s) = \frac{1}{m}\sum_{i=1}^{m} Y(\delta_i^*; s) + \log\left(\frac{1-e^{-\hat{\alpha}\delta}}{\left(\prod_{i=1}^{m}\left(1-e^{-\hat{\alpha}\delta_i^*}\right)\right)^{1/m}}\right). \quad (10)$$

We will evaluate the accuracy of prediction models with one or more point predictors in Section 5.

3.2.4 Estimating the effective growth exponent. To train the predictor of the effective growth exponent in equation (9), we need training examples with the response variable corresponding to the effective growth exponent. One way to compute the effective growth exponent is to use MLE for given observed data. This is computationally expensive so we discuss two simpler estimators.

Mean value based estimator. By Proposition 3.2, for every $t \geq 0$,

$$\mathbb{E}[\,N(+\infty) - N(t) \,|\, \mathcal{F}_t] = \frac{\lambda(t)}{\alpha}.$$

We can show that

$$\mathbb{E}\left[\int_s^{\infty}(N(+\infty) - N(t))dt \,\middle|\, \mathcal{F}_s\right] = \frac{1}{\alpha}\mathbb{E}[\,N(+\infty) - N(s) \,|\, \mathcal{F}_s]$$

which follows from derivations in Appendix A.9. [23] This leads us to define the following estimator

$$\hat{\alpha} = \frac{N(+\infty) - N(s)}{\int_s^{\infty}(N(+\infty) - N(u))du}.$$

Suppose $s = 0$ and $N(s) = 0$ and let $T_1, T_2, \ldots, T_n$ denote the observed points. It can be shown that

$$\int_0^{\infty}(n - N(t))dt = \sum_{i=1}^{n} T_i$$

which follows by some simple calculations provided in Appendix A.9 [23]. Hence, we have

$$\hat{\alpha} = \frac{1}{\frac{1}{n}\sum_{i=1}^{n} T_i}.$$

This shows that $\hat{\alpha}$ is the reciprocal of the mean point time.

Quantile value based estimator. An alternative estimator can be defined based on computing a quantile value as described next. For any fixed value $\gamma \in (0, 1)$, let

$$T_\gamma = \inf\{t > 0 : N(t) \geq \gamma N(+\infty)\}.$$

Notice that if $\gamma = 1/2$, then we can interpret $T_{1/2}$ as the median value of the observed point times. Intuitively, we may think of T_γ as of an estimator of τ_γ, defined by $\mathbb{E}\left[\,N(\tau_\gamma)\middle|\,\mathcal{F}_0\right] = \gamma\mathbb{E}[\,N(+\infty) \,|\, \mathcal{F}_0]$. We already noted in (6) that $\tau_\gamma = \log(1/(1-\gamma))/\alpha$. Hence, this leads us to define $\hat{\alpha} = 1/T_\gamma$, provided that $T_\gamma > 0$.

In Appendix A.10 [23], we provide a theoretical bound on the bias of the quantile value based estimator. In Section 5, we empirically compare the two estimators on real-world data.

4 DISCUSSION

In this section we discuss the computation complexity of some previously proposed methods based on point process models as well as of our prediction method presented in Section 3.2.

In order to make predictions by using expressions for $\mathbb{E}[N(t) - N(s) \mid \mathcal{F}_s]$ or $\Lambda(s,t)$ discussed in Section 3.1 for different point process models, we need to compute these values which has certain computation cost. This computation cost is incurred both at *training time* (for computing values of prediction variables used for supervised learning) and at *prediction time*. Moreover, additional computation cost is incurred for estimating unknown model parameters at training time.

For general Hawkes point processes, the computation of $\mathbb{E}[N(t) - N(s) \mid \mathcal{F}_s]$ or $\Lambda(s,t)$ can be prohibitively expensive for implementation in large-scale online platforms. Evaluation of these quantities have $\Omega(N(s))$ computation complexity, i.e. it is at least linear in the number of points in the observed history. For popularity prediction in social media platforms, this number can be large, in the order of millions and possibly even larger.

We next discuss computation complexity of these evaluations for several well-known methods (namely, Reinforced Poisson Process, SEISMIC, Hawkes Intensity Process, and Hawkes with exponential kernel). We do not discuss computation complexity of deep learning extensions of these models as they have same or higher complexity.

Reinforced Poisson Processes. RPP model [40] has the stochastic intensity function $\lambda(t) = pf(t)N(t)$ where p is a positive-valued infection-rate parameter and $f(t)$ is a probability density function. The model assumes f to be a log-normal density function, which has two parameters. This model does not exactly fall in the framework of Hawkes point processes, but it is a self-excited point process model. The conditional expected number of points over an arbitrary time horizon is given by

$$\mathbb{E}[N(t) - N(s) \mid \mathcal{F}_s] = N(s)\left(e^{p(F(t)-F(s))} - 1\right).$$

The model requires to track the total number of points observed by any given time, which can be efficiently tracked in a streaming computation setting. However, the model is computationally expensive as it requires to fit model parameters for each content item using a Maximum Likelihood Estimator (MLE), which requires using an iterative optimization method. Specifically, the time complexity of this approach $\Omega(M \times N(s))$ is proportional to the number of iterations M of the optimization method (which can be considerably large in practice) times the number of points in the history $N(s)$.

SEISMIC. This model [51] is a Hawkes point process model with a power-law kernel $p\phi(x)$ where $\phi(x)$ is given in (2). The model is defined by letting marks Y_i be the degrees d_i of nodes re-sharing information in an online social network. The two parameters of the function $\phi(x)$ are assumed to be hyper-parameters, and an MLE is used to estimate parameter p by using the observed part of a cascade. This estimator can be expressed in a closed form as

$$\hat{p} = \frac{N(s)}{\sum_{i=1}^{N(s)} d_i \Phi(s - T_i)}.$$

The paper [51] uses a variant of this estimator that involves some smoothing. Clearly, the computation complexity for evaluating the value of estimator $\hat{p}$ is $\Omega(N(s))$.

Hawkes Intensity Process. The HIP method [39] assumes a Hawkes point process with a power-law kernel function and is based on estimating the model parameters by fitting the expected value of the stochastic intensity function to observed data at fixed time instances. For general Hawkes point processes, the expected value of the stochastic intensity function obeys a convolutional equation, which is leveraged by the proposed approach. This approach still requires using an iterative optimization method and has the time complexity comparable to RPP.

Hawkes with exponential kernel. For the Hawkes point process model with exponentially decaying intensity, by Proposition 3.2, we need to evaluate the value of the stochastic intensity $\lambda(s)$ in order to compute the value of $\mathbb{E}[N(t) - N(s) | \mathcal{F}_s]$. The stochastic intensity $\lambda(s)$ can be approximated by a *velocity statistic* which measures the local rate of points at time s. For instance, we may define the velocity as the rate of points observed over $[s - d, s]$ for some fixed value $d > 0$. Velocity can be efficiently tracked and queried in constant time by using a sliding-window algorithm over the stream of observed points [18]. For estimating the other two parameters of the model, namely ρ_1 and β, one may use an MLE optimization method. This approach, as in the methods mentioned above, may induce significant computation costs. An alternative approach is to use an estimator for the effective growth exponent α. This parameter is both sufficient for prediction purposes (see Proposition (3.2)) and an estimator of this parameter be efficiently computed (see Section 3.2.4).

5 EXPERIMENTAL RESULTS

In this section we present our numerical results. We first provide basic information about datasets that we used for training and evaluation, the models we chose for comparison and our evaluation metrics. We then provide results on the accuracy of predictions over infinite and then varied time horizons. Our choice of baseline models includes previously proposed popularity prediction models based on self-excited point processes, and separately trained machine learning models for specific prediction time horizons. Overall our results show that our proposed method can provide more accurate predictions than other self-excited point process models, and that our method achieves competitive performance to models trained for specific prediction horizons.

5.1 Datasets, models and evaluation metrics

Datasets. For our experiments, we used datasets containing de-identified public Facebook posts created by pages (Facebook accounts of companies, brands, celebrities, and other public entities) and collected over different time periods. These datasets cover a large number of view and reshare events – in the order of billions – and hundreds of thousands of posts. Specifically, we used a dataset containing 100 thousand public page posts which were reviewed by moderators but deemed to not violate Facebook Community Standards. These posts were created within 2 weeks in October 2020; we tracked their reshares and views for up to 2 months after creation. The number of views recorded on these posts is in the order of hundreds of billions. We also used a second dataset containing 200 thousand randomly sampled public page posts created within 1 week in November 2019, and also tracked their reshares and views for up to 2 months after creation, collecting timestamps of several billions of such events. We used the first dataset to evaluate prediction accuracy of different models for infinite horizons. For validating performance on the varied prediction horizons we used both datasets and obtained similar results. Hence, for varied prediction horizons we only present results on the second dataset. We believe that datasets we use are typical and hence the claims made in this section would generalize to other datasets.

Our prediction model. The Hawkes model we propose is defined in Section 3.2. We use gradient boosted decision trees from the scikit-learn library [20] for point predictors of the view counts for given reference horizons and the effective growth parameters. We use a set of 1889 features, which could be categorized into *content*

Table 1: Prediction performance for the proposed Hawkes model vs. SEISMIC-CF, overall, and conditional on content popularity (Low, High) or prediction time (Early, Late).

Dataset	Hawkes			SEISMIC-CF		
	MAPE	τ	RMSE	MAPE	τ	RMSE
Overall	0.565	0.821	2.0e6	0.698	0.769	6.5e6
Low	0.651	0.713	9.8e4	0.802	0.633	1.6e7
High	0.552	0.796	2.2e6	0.685	0.744	2.4e7
Early	0.451	0.824	1.4e6	0.667	0.752	9.9e7
Late	0.573	0.821	2.3e6	0.737	0.762	2.8e7

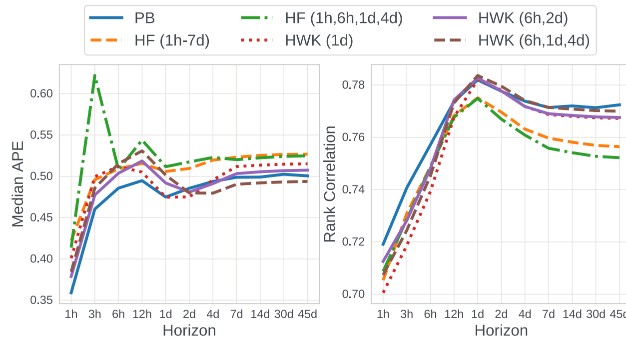

Figure 1: Prediction performance for different horizons: (left) median absolute prediction error and (right) rank correlation. The results are for the proposed Hawkes models with one reference time horizon (HWK (1d)), two reference time horizons (HWK (6h,4d)) and three reference time horizons (HWK (6h,1d,4d)), point-based models (PB), and horizon-as-feature models, one trained on all considered horizons between 1 hour and 7 days (HF (1h-7d)) and another one trained only on a subset of them (HF (1h,6h,1d,4d)).

features (properties of the post), *page features* (properties of the account that created the post), and *engagement features* (patterns of users' interactions with the post and the page). Appendix A.16 [23] provides details on these groups of features and their cumulative importance for both regressors. As expected, engagement features have the highest importance scores for both regressors. However, the long term patterns of a cascade's growth – as indicated in the case of predicting effective growth exponent α – are better explained by the characteristics of the page and the *page-level engagement* features. In contrast, the *content engagement* features are by far the most important for popularity prediction over shorter horizons.

Baselines. We compare prediction accuracy of our model against several carefully chosen baselines drawn from relevant literature. Our first set of baselines are taken from the class of generative models based on self-excited point processes. In particular, we compare against a variant of SEISMIC [51] adapted for predicting popularity of Facebook posts following [44] and the RPP model [40]. We used the source code of SEISMIC model from the original paper [1]. For RPP, we have not found the original source code of the model and opted for reproducing it in Python. These baseline models are representative of the family of generative models based on self-excited point processes, and their computation complexity is not so high as to make them unusable on our data (in contrast with other more complex models like those that combine deep learning with self-excited processes). Our second set of baselines consists of prediction models separately trained for specific reference prediction horizons (hereafter "PB"), and a prediction model that uses the the horizon as the feature (hereafter "HF"). We will provide some more discussion about the baseline models in the following sections.

Evaluation metrics. Following [51], we evaluated prediction accuracy using Median Absolute Percentage Error (MAPE) and τ Rank Correlation; we also added some evaluation results using Root Mean Squared Error (RMSE).

5.2 Predictions for infinite horizons

In this section we present our numerical results on the prediction accuracy of our model and compare with two baselines, namely, a variant of SEISMIC and RPP models, which we introduced in Section 4. The presented numerical results demonstrate that our model can achieve superior prediction accuracy than these baseline models, by leveraging static features, and that this can be achieved at a much smaller computation time cost.

We compare our approach against a SEISMIC-CF variant of the model proposed for Facebook cascades in [44]. We used default values for the constant node degree parameter proposed for SEISMIC-CF and for the kernel function parameters. We have explored various other settings of parameter values and obtained similar results. As it can be seen in Table 1, our model outperforms SEISMIC-CF on both Median APE and Rank Correlation by a margin of 13% and 5%, respectively. This also holds true across different splits we tested on – namely, low vs. high popularity items (less or more than 1000 views) and early vs. late predictions with respect to content age at prediction time (less or more than 24 hours since content creation). The performance gap is especially striking when comparing predictions by using the RMSE metric, where for low popularity items and early predictions, our model is orders of magnitudes more accurate than SEISMIC-CF.

We have also conducted experiments to compare against RPP, which we introduced in Section 4. As discussed in Section 4, the computation complexity of fitting RPP model for each content item is proportional to the product of the number of steps of the MLE optimization algorithm and the number points in the observed history. In out settings, this was in the order of minutes for high popularity content items in comparison to less than a second for our proposed model. We managed to evaluate RPP on a small subset of content items in our dataset and achieved a MAPE of 4.1, which is significantly worse than for our model.

5.3 Predictions over arbitrary horizons

In this section we compare prediction performance of our model against two different baseline models, including models that are separately trained for specific prediction time horizons and models

[1]http://snap.stanford.edu/seismic/

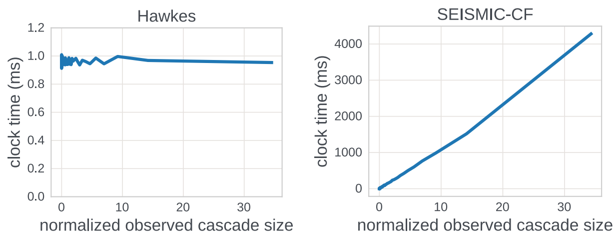

Figure 2: Computation cost of Hawkes and SEISMIC-CF models as a function of the normalised observed cascade size.

that use the prediction time horizon as the input feature. More specifically, we consider: (a) *Point-based (PB)* models that are trained separately for every given prediction time horizon. Although in practice it might not be feasible to maintain a family of models for potentially infinite horizons of interest, this approach provides a good estimate for upper bound performance when a dedicated model is trained for each horizon. (b) *Horizon-as-feature (HF)* models for popularity prediction that use the prediction time horizon as the input feature. This requires training examples sampled at a multitude of horizons δ, i.e. $Y(\delta; s) = h(\delta, x, \tau(\mathcal{F}_s); \theta)$, which has an additional independent variable δ.

The PB models may be regarded as a strong baseline for comparison of prediction performance for specific prediction time horizons, as they are trained for these specific prediction time horizons. The HF models may be regarded as a natural class of prediction models.

For training HF models, we sample prediction time horizons in the range between 1h and 7d for each content item, hence synthetically increasing the size of the training set by the number of considered horizons, i.e., eight-fold for a model variant trained on all considered horizons in the range (HF (1h-7d)) and four-fold for a model variant trained only on a subset of them (HF (1h,6h,1d,4d)).

We compare the performance of our model against the aforementioned baselines for different reference time horizons of our model. We denote our model as HWK($\delta_1^*, \ldots, \delta_m^*$) for given reference prediction time horizons $\delta_1^*, \ldots, \delta_m^*$.

As seen in Figure 1, all considered Hawkes models outperform the HF baselines on longer horizons (δ > 24h) with the best one (HWK (6h,1d,4d)) having an average decrease of 7% in Median APE and an average increase of 2% in Rank Correlation. Evidently, the HF model struggles to generalize beyond the horizons it has been trained on, as seen from the sharp drops of the HF (1h,6h,1d,4d)'s performance for δ = 3h, 12h, 2d in comparison to the HF (1h-7d) variant trained on all horizons in the range. Last but not least, our model also reaches a parity in performance with PB models for δ > 24h, suggesting its good generalization capability for long prediction horizons.

We further discuss tuning of the reference horizon parameters δ and performance of the models on cascades of different sizes in Appendix A.17 and Appendix A.18 [23], respectively.

5.4 Computation cost of different methods

We evaluate computation cost of different methods by measuring the clock time for the computation required to predict the final cascade size on the testing set. We ran these experiments on a sever with 24 Intel Core Processor (Broadwell) CPUs and 114GB of RAM. In Figure 2 we report the mean clock time in milliseconds for generating predictions on cascades of different observed sizes (normalized by the average value) in SEISMIC-CF and Hawkes models.

As anticipated in Section 4, the computational cost for SEISMIC-CF scales linearly with the observed cascade size $N(s)$. Indeed, it can vary 4000x between predictions on cascades with a handful of observed events and cascades with millions of observed events. This is because SEISMIC-CF model requires a pass through all events in the history of the cascade to yield a prediction. As discussed in Sec 4, other considered models require multiple passes through the observed history of a cascade to produce a prediction for each content item and hence their computation complexity will increase even faster.

In contrast, our proposed Hawkes model has a constant computation time for making predictions of any observed cascade size. This is because it only requires an inference from few gradient boosted decision tree models. The static and temporal features we use in the model (discussed in detailes in Appendix A.16 [23]) can be computed efficiently at prediction time. For instance, the temporal features in our model constitute simple counters of events in the observed history of a cascade. These counters can be tracked efficiently with a dedicated data structure and fetched in constant time with respect to the cascade history size [18].

This result confirms our theoretical findings and suggests that our proposed model can effectively operate at Facebook scale.

6 CONCLUSION

We proposed a model for popularity prediction of social media items that satisfies a set of design considerations that arise in large-scale online platforms. These considerations include providing accurate predictions for any given prediction time and horizon, having a constant-time computation complexity at prediction time, and leveraging both static and temporal features to ensure accurate predictions. The model requires combining only a few point predictors, including prediction of the view count acquired up to one or more fixed reference time horizons and a predictor of the effective popularity growth rate. The prediction accuracy is shown to be competitive to separately trained models for specific prediction time horizons, using a large collection of post sharing on Facebook.

Future work may further explore the space of scalable popularity prediction methods, and study the trade-off between computation complexity and prediction accuracy.

REFERENCES

[1] Mohamed Ahmed, Stella Spagna, Felipe Huici, and Saverio Niccolini. 2013. A Peek into the Future: Predicting the Evolution of Popularity in User Generated Content. In *Proc. of ACM WSDM '13*.
[2] Sitaram Asur and Bernardo A. Huberman. 2010. Predicting the Future with Social Media. In *Proc. of IEEE/WIC/ACM WI-IAT '10*.
[3] Lars Backstrom, Jon Kleinberg, Lillian Lee, and Cristian Danescu-Niculescu-Mizil. 2013. Characterizing and Curating Conversation Threads: Expansion, Focus, Volume, Re-entry. In *Proc. of ACM WSDM '13*.
[4] Eytan Bakshy, Brian Karrer, and Lada A. Adamic. 2009. Social Influence and the Diffusion of User-created Content. In *Proc. of ACM EC '09*. 325–334.
[5] Eytan Bakshy, Itamar Rosenn, Cameron Marlow, and Lada Adamic. 2012. The Role of Social Networks in Information Diffusion. In *Proc. of WWW '12*. 519–528.

[6] Christian Bauckhage, Fabian Hadiji, and Kristian Kersting. 2015. How Viral Are Viral Videos?. In *Proc. of AAAI ICWSM '15*.

[7] Christian Bauckhage, Kristian Kersting, and Fabian Hadiji. 2013. Mathematical Models of Fads Explain the Temporal Dynamics of Internet Memes. In *Proc. of AAAI ICWSM '13*.

[8] Qi Cao, Huawei Shen, Keting Cen, Wentao Ouyang, and Xueqi Cheng. 2017. DeepHawkes: Bridging the Gap Between Prediction and Understanding of Information Cascades. In *Proc. of ACM CIKM '17*.

[9] Qi Cao, Huawei Shen, Jinhua Gao, Bingzheng Wei, and Xueqi Cheng. 2020. Popularity Prediction on Social Platforms with Coupled Graph Neural Networks. In *Proc. of ACM WSDM '20*.

[10] George H. Chen, Stanislav Nikolov, and Devavrat Shah. 2013. A Latent Source Model for Nonparametric Time Series Classification. In *Proc. of NIPS '13*.

[11] X. Chen, F. Zhou, K. Zhang, G. Trajcevski, T. Zhong, and F. Zhang. 2019. Information Diffusion Prediction via Recurrent Cascades Convolution. In *Proc. of IEEE ICDE '19*. =.

[12] Justin Cheng, Lada Adamic, P. Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. 2014. Can Cascades Be Predicted?. In *Proc. of WWW '14*.

[13] Justin Cheng, Lada A. Adamic, Jon M. Kleinberg, and Jure Leskovec. 2016. Do Cascades Recur?. In *Proc. of WWW '16*.

[14] Justin Cheng, Jon Kleinberg, Jure Leskovec, David Liben-Nowell, Bogdan State, Karthik Subbian, and Lada Adamic. 2018. Do Diffusion Protocols Govern Cascade Growth?. In *Proc. of AAAI ICWSM '18*.

[15] Hyunyoung Choi and Hal Varian. 2012. Predicting the Present with Google Trends. *Economic Record* 88, s1 (2012), 2–9.

[16] Peng Cui, Shifei Jin, Linyun Yu, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2013. Cascading Outbreak Prediction in Networks: A Data-driven Approach. In *Proc. of ACM KDD '13*.

[17] Angelos Dassios and Hongbiao Zhao. 2011. A dynamic contagion process. *Advances in Applied Probability* 43, 3 (2011), 814–846.

[18] Mayur Datar, Aristides Gionis, Piotr Indyk, and Rajeev Motwani. 2002. Maintaining stream statistics over sliding windows. *SIAM J. Comput.* 31, 6 (2002), 1794–1813.

[19] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent Marked Temporal Point Processes: Embedding Event History to Vector. In *Proc. of ACM KDD '16*.

[20] Jerome H Friedman. 2002. Stochastic gradient boosting. *Computational statistics & data analysis* 38, 4 (2002), 367–378.

[21] Daniel Gruhl, R. Guha, David Liben-Nowell, and Andrew Tomkins. 2004. Information Diffusion Through Blogspace. In *Proc. of WWW '04*.

[22] Adrien Guille and Hakim Hacid. 2012. A Predictive Model for the Temporal Dynamics of Information Diffusion in Online Social Netw.. In *Proc. of WWW '12*.

[23] Daniel Haimovich, Dima Karamshuk, Thomas J Leeper, Evgeniy Riabenko, and Milan Vojnovic. 2021. Scalable Prediction of Information Cascades over Arbitrary Time Horizons. *Technical report, https://arxiv.org/abs/2009.02092* (2021).

[24] Liangjie Hong, Ovidiu Dan, and Brian D. Davison. 2011. Predicting Popular Messages in Twitter. In *Proc. of WWW '11*.

[25] Maximilian Jenders, Gjergji Kasneci, and Felix Naumann. 2013. Analyzing and Predicting Viral Tweets. In *Proc. of WWW '13*.

[26] Ryota Kobayashi and Renaud Lambiotte. 2016. TiDeH: Time-Dependent Hawkes Process for Predicting Retweet Dynamics. In *Proc. of AAAI ICWSM '16*.

[27] Quyu Kong, Marian-Andrei Rizoiu, and Lexing Xie. 2020. Modeling Information Cascades with Self-Exciting Processes via Generalized Epidemic Models. In *Proc. of ACM WSDM '20*.

[28] Andrey Kupavskii, Liudmila Ostroumova, Alexey Umnov, Svyatoslav Usachev, Pavel Serdyukov, Gleb Gusev, and Andrey Kustarev. 2012. Prediction of Retweet Cascade Size over Time. In *Proc. of CIKM '12*.

[29] Sylvain Lamprier. 2019. A Recurrent Neural Cascade-based Model for Continuous-Time Diffusion. In *Proc. of ICML '19*.

[30] Cheng Li, Jiaqi Ma, Xiaoxiao Guo, and Qiaozhu Mei. 2017. DeepCas: An End-to-end Predictor of Information Cascades. In *Proc. of WWW '17*.

[31] Hui Li, Hui Li, and Sourav S. Bhowmick. 2020. CHASSIS: Conformity Meets Online Information Diffusion. In *Proc. of SIGMOD '20*.

[32] Dongliang Liao, Jin Xu, Gongfu Li, Weijie Huang, Weiqing Liu, and Jing Li. 2019. Popularity Prediction on Online Articles with Deep Fusion of Temporal Process and Content Features. In *Proc. of the AAAI '19*.

[33] Zongyang Ma, Aixin Sun, and Gao Cong. 2013. On predicting the popularity of newly emerging hashtags in Twitter. *Journal of the American Society for Information Science and Technology* 64, 7 (2013), 1399–1410.

[34] Yasuko Matsubara, Yasushi Sakurai, B. Aditya Prakash, Lei Li, and Christos Faloutsos. 2012. Rise and Fall Patterns of Information Diffusion: Model and Implications. In *Proc. of ACM KDD '12*.

[35] Hongyuan Mei and Jason Eisner. 2017. The Neural Hawkes Process: A Neurally Self-modulating Multivariate Point Process. In *Proc. of NIPS '17*.

[36] Swapnil Mishra, Marian-Andrei Rizoiu, and Lexing Xie. 2016. Feature Driven and Point Process Approaches for Popularity Pred.. In *Proc. of ACM CIKM '16*.

[37] Nuno Moniz and Luís Torgo. 2019. A review on web content popularity prediction: Issues and open challenges. *Online Social Networks and Media* 12 (2019), 1–20.

[38] Marian-Andrei Rizoiu, Swapnil Mishra, Quyu Kong, Mark Carman, and Lexing Xie. 2018. SIR-Hawkes: Linking Epidemic Models and Hawkes Processes to Model Diffusions in Finite Populations. In *Proc. of WWW '18*.

[39] Marian-Andrei Rizoiu, Lexing Xie, Scott Sanner, Manuel Cebrian, Honglin Yu, and Pascal Van Hentenryck. 2017. Expecting to Be HIP: Hawkes Intensity Processes for Social Media Popularity. In *Proc. of WWW '17*.

[40] Huawei Shen, Dashun Wang, Chaoming Song, and Albert-László Barabási. 2014. Modeling and Predicting Popularity Dynamics via Reinforced Poisson Processes. In *Proc. of AAAI '14*.

[41] Aleksandr Simma and Michael I. Jordan. 2010. Modeling Events with Cascades of Poisson Processes. In *Proc. of UAI '10*.

[42] Karthik Subbian, B. Aditya Prakash, and Lada Adamic. 2017. Detecting Large Reshare Cascades in Social Networks. In *Proc. of WWW '17*.

[43] Gabor Szabo and Bernardo A. Huberman. 2010. Predicting the Popularity of Online Content. *Commun. ACM* 53, 8 (2010), 80–88.

[44] Linpeng Tang, Qi Huang, Amit Puntambekar, Ymir Vigfusson, Wyatt Lloyd, and Kai Li. 2017. Popularity Prediction of Facebook Videos for Higher Quality Streaming. In *Proc. of USENIX ATC '17*.

[45] Oren Tsur and Ari Rappoport. 2012. What's in a Hashtag?: Content Based Prediction of the Spread of Ideas in Microblogging Communities. In *Proc. of ACM WSDM '12*.

[46] James Vincent. 2020. Facebook is now using AI to sort content for quicker moderation. The Verge - https://www.theverge.com/2020/11/13/21562596/facebook-ai-moderation.

[47] Senzhang Wang, Zhao Yan, Xia Hu, Philip S. Yu, and Zhoujun Li. 2015. Burst Time Prediction in Cascades. In *Proc. of AAAI '15*.

[48] Jaewon Yang and Jure Leskovec. 2011. Patterns of Temporal Variation in Online Media. In *Proc. of ACM WSDM '11*.

[49] Shuang-Hong Yang and Hongyuan Zha. 2013. Mixture of Mutually Exciting Processes for Viral Diffusion. In *Proc. of ICML '13*.

[50] Bowen Zhang and Wing Cheong Lau. 2018. Temporal Modeling of Information Diffusion Using MASEP: Multi-Actor Self-Exciting Proc.. In *Proc. of WWW '18*.

[51] Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. 2015. SEISMIC: A Self-Exciting Point Process Model for Predicting Tweet Popularity. In *Proc. of ACM KDD '15*.

[52] Fan Zhou, Xovee Xu, Goce Trajcevski, and Kunpeng Zhang. 2021. A Survey of Information Cascade Analysis: Models, Predictions, and Recent Advances. *Comput. Surveys* 54, 2 (March 2021).

[53] Ke Zhou, Hongyuan Zha, and Le Song. 2013. Learning Social Infectivity in Sparse Low-rank Networks Using Multi-dimensional Hawkes Processes. In *Proc. of AISTATS '13*.

[54] Ke Zhou, Hongyuan Zha, and Le Song. 2013. Learning Triggering Kernels for Multi-Dimensional Hawkes Processes. In *Proc. of ICML '13*.