Journal of Data Intelligence, Vol. 2, No. 1 (2021) 064–094

EVENT PREDICTION IN ONLINE SOCIAL NETWORKS

LEONARD TAN
School Of Biological Sciences, Nanyang Technological University, 60 Nanyang Drive, Singapore
Singapore, 637511, Singapore
Leonard.wl.tan@gmail.com

THUAN PHAM
School Of Engineering And Sciences, The University of Southern Queensland, 45 West Street, Darling Heights
Toowoomba, Queensland 4305, Australia
Thuan.Pham@usq.edu.au

HANG KEI HO
Faculty Of Social Sciences, The University Of Helsinki, Yliopistonkatu 4
Helsinki, 00100, Finland
hang.kei.ho@helsinki.fi

TAN SENG KOK
Construction Intelligence, Construction AI Research Labs, Applipro Services, 209 Henderson Road
Singapore, 159551, Singapore
aps_rudi@yahoo.com

Event prediction is a very important task in numerous applications of interest like fintech, medical, security, etc. However, event prediction is a highly complex task because it is challenging to classify, contains temporally changing themes of discussion and heavy topic drifts. In this research, we present a novel approach which leverages on the RFT framework developed in [1]. This study addresses the challenge of accurately representing relational features in observed complex social communication behavior for the event prediction task; which recent graph learning methodologies are struggling with. The concept here, is to firstly learn the turbulent patterns of relational state transitions between actors preceeding an event and then secondly, to evolve these profiles temporally, in the event prediction process. The event prediction model which leverages on the RFT framework discovers, identifies and adaptively ranks relational turbulence as likelihood predictions of event occurrences. Extensive experiments on large-scale social datasets across important indicator tests for validation, show that the RFT framework performs comparably better by more than 10% to HPM [2] and other state-of-the-art baselines in event prediction.

Keywords: Event Prediction, Artificial Intelligence, Topic Modeling, Wavelet Transformation, Fractal Neural Networks

1. Introduction

Event prediction is a complex topic which encompasses a mix of multiple disciplines across wide ranging applications [3]. Some of them include recommender systems, marketing and advertising, governance and rule, news and propaganda, etc. [4]. Some examples of emerging event prediction applications include preemptive disease and medical condition prevention, patient-drug matching pair diagnosis and administration, cyber security, data privacy and utility, etc. The social pre-cursors of a large majority of real life events are often staged through popular online social media like facebook, twitter, google, etc. These pre-cursors are often identified as activity through online social medium as information transactions [5]. Although it may be intuitive to think of a similarity based approach on how an actor incites other members within a community through matching attributes, such a culmination of affective sentiments are often times a lot less direct [6].

1.1. *Challenges*

Although numerous approaches [7], [8], [9], [10], [11], [12], [13] have been developed to address certain areas of semantic / spatio-temporal event detection and prediction, their methods have been limited in applications to specific events [5], [9], [13] in question. Furthermore, techniques to date focuses on the application of batch learning methods which can only be used at static instances in time [14]. Such approaches are known to be unscalable to continuous (social KG) data streams and changing environment contexts [15].

In the same vein, many relational learning approaches used in trending studies, also lack depth and representative power [1]. For example, [7], [8], [9], [10], [11], [12], [13] are methods developed to date which rely primarily on learning how spatio-temporal dependencies of shallow word-feature changes. These semantic patterns are then used as precursors - which are modeled to predict eventful occurances in a future timeframe. Furthermore, the key critical key questions uncovered in [1] still remain unanswered.

1.2. *Data Model*

To address these problems, this paper adopts the Fractal Neural Network (FNN) model which was developed in [1] and extends its efficacy by adapting the dynamic growth of the fractal network into a robust adversarial framework. FNNs, encode ground truths of the Relational Turbulence Theory (RTT) [16] framework into the lowest principle decompositions of our model. It is able to self-evolve from a meta-learning perspective - in response to random "anytime-sequenced" data streams of fluctuating information sophistication [17]. Next, we define what relational turbulence is and explain our motivation.

In this approach, the main motivation is to firstly, characterize Relational Turbulence by probabilistic measures of Relational Intensity $P(\gamma_{rl})$, Relational Interference $P(\vartheta_{rl})$ and Relational Uncertainty $P(\varphi_{rl})$ [18]. Secondly, extensions on [1] uses the principle of Relational Turbulence Theory (RTT) [19] to establish a framework of theoretical processes linking evolving relational features learned over past event occurrences (causals). The main novelty of our model focuses firstly, on discovering relational intelligence through identifying relational profiles on three popular social Knowledge Graphs (KGs): Twitter, Google and Enron email datasets. Then secondly, leveraging on this discovery to generalize event occurrences for three major social streaming platforms: Twitter, Google Feed and Live Journal.

1.2.1. *Research Objectives*

This paper addresses the following important research objectives for event prediction tasks:

- Accurate representations and correlations of complex online social behavior in OSNs for generalized event prediction.

- Classification of relational profiles to dynamic social communication patterns in predicting events.
- Quantification of dynamic errors arising from social disruptions (outliers) in event-predictive representations.

1.3. *Technical Model*

The technical model of this paper extends the RFT architecture developed in [1] to effectively represent the dynamism of popular key relational dimensions uncovered from previous approaches and techniques conducted on online social structures. It is developed from the principles of self evolving fractals and artificial neural networks in a real-time machine learning model for active data streams [20], [21]. This current extension of RFT is capable of representing social structures as a time evolving flow of relational attributes (time-realistic relationships) between node entities of a network in question with the constant inception of social shocks. Its objective function describes the turbulence profiles of social graph constructs and their resulting communication behavioral patterns across apriori relational state altering events - to predict likelihood occurrences of tracked topics as events of interest.

Similar to [1], the extended RFT model accepts as inputs, the concurrent key relational feature states f_i between actors E_ϵ from past and present social transactions to predict the likelihood of an event occurance E_φ in an evolving state of relational turbulence τ_{ij} from an identified social flux F_ϵ within a continuous stream of social transactions [22], [23].

1.4. *Contributions*

In this study, we examine the dynamic structure of such an shallow ANN known as fractals. We design a fractal layered structure that maintains a key property of self-similarity across different varying scales [20]. Fractals generally maintain structural affinity [24] and can grow to become complex enough to represent high levels of sophistication that are yet trivially efficient enough to re-create by repeating similar simple shallow architectures in a loop - ad infinitum. The main scientific contributions of our work are presented as follows:

(i) We extend the RFT method in [1] to continuously adapt to real-time streaming social transactions to predict occurance of events from tracked topics of interest;
(ii) The RFT model in [1] was extended into an adversarial FNN framework architecture which is used to significantly reduce event prediction errors and improve accuracy.
(iii) The method adaptively learns from a Fractal Neural Network (FNN) which builds on key relational fractal structures discovered in a given Online Social Network (OSN) from tracked topics.
(iv) Our experimental design and detailed results on Twitter, Google and Enron email datasets at different instances show that RFT is able to offer a good modeling of relational ground truths. While FNN is able to efficiently and accurately represent evolving relational turbulence and flux profiles within a given OSN.
(v) The extended RFT model improves the efficacy and efficiency of existing approaches

towards event prediction through studies and comparisons of experimental results conducted with real-life social networks on Twitter, Googlefeed and LiveJournal datasets.

The remaining part of the paper is organized as follows: Section II presents a brief overview of related works drawn from social theories and relational structures. Section III discusses the methodology we have developed for predicting events in OSNs. Section IV introduces our experimental data, model baselines and design. Section V presents the results and discussion of this paper that leads to a conclusion and potential future directions in section VI.

2. Related Literature

2.1. *Relational Turbulence*

Relational Turbulence was first studied in [25]. It was typically characterized as a resultant state in conflict of interests from competing goals between two or more actors in question [26]. Although conflict does provide the basis of stimulation for communication within a relationship that is centered in a flux, it also correlates to negative consequences in the form of detrimental event occurrences if left undetected and unchecked [26]. An important discriminator of detecting conflict and hence the resulting turbulence in any relationship model between networks of actors is the observation and management of relational altering events. As reciprocated negative expectancy violations grow larger over time, instability in a cumulative relational flux of an OSN increases [1].

Excluding relational expectation management, some detrimental relational altering events include: geographic displacements (or low proximity measures), conflict escalation (high frequencies of friction), environmental changes (expectation disparities), etc. [27], [28], [29]. Relational Turbulence is briefly defined as changes which occur within a relationship that may cause friction ([**?**]) between actors and their local online community. These changes are mostly studied as a series of transitions (often abrupt) between actor-environmental states that inadvertently influences relational characteristics by altering communication flux patterns [30] of a given relationship in an Online Social Network (OSN). These shifts in relational characteristics during difficult state transitions (altering events) may lead to volatile consequences.

The Relational Turbulence Model (RTM) [18] defines an artificial construct which enables intelligent predictions of communication behaviors during relationship transitions, in an environment of continuous online social disruptions. Turbulent relationship development shifts between continuous and affective communicative states of flux which are affected by the polarization of sentiments. The extent of such polarizations are characterized by actor interferences and relational uncertainty as state transition probabilities that can cause conflict [1]. These two prime relational features in OSNs enable the effective detection and prediction of conflict and event occurrences in sentimental and affective computing. While RTM explains and predicts relational conflict through communicative behaviors between actors, Relational Turbulence Theory (RTT) [19] correlates uncertainty and interference to specific behaviors, actions and sentiments (either hidden or expressed). A summary of research work done on relational turbulence is shown in Table 1.

Table 1. Table of related research on relational turbulence.

Study	Subject	Features
[16]	Relational Turbulence Theory	Relational Interference, Intensity and Uncertainty, Expectancy Violation, Engagement, Valance.
[31]	Expectancy Violation	Sentiment polarity, triadic closure, Emotion Gradient, Relational State Aberrations.
[32]	Relational Turbulence Model	Relational disruptions, Relational state altering events, Uncertainty, Associative irritations, Longitudinal analysis.
[33]	Theory of Engagement	Multi-agent perspective, Engagement theory.
[26]	Relational characteristics	Reciprocity, Directed information transfer, latent semantics.
[30]	Relational state transitions	Relational disruptions, Gradient turning points.
[1]	Relational Intelligence	Relational Turbulence profile, State transition.
[34]	Relational Fractals	Sentiment, Confidence, Mentions.

2.2. *Event Prediction*

As a broad overview, there are two categories of mainstream methods used for predicting events. The first category is the markovian sequenced model (also known as association rule based prediction) [35]. In this category, future event occurances are predicted based on past event association patterns. While this approch is able to capture temporal features relative to key (anchor) events, it assumes that events are correlated to each other in a fixed sequence. The second category is the stochastic word distribution model (also known as narrative generation) [36]. In this category, future event occurances are predicted based on the topic-context word (semantic) distributions surrounding key actors in question. For example, when the name "Donald Trump" and the topic-context "President of the United States" is mentioned, there will be major events which are stochastically related (e.g. trade wars, tax tarrifs, mexico border, etc.). While this approach is able to draw a coreference resolution between word-topic to events, it overlooks the temporal aspects of such occurances [36].

Recently, there is an emergence of a third category of multi-task learning models [10], [11], [12], [13] which leverage on correlations of entities in knowledge graphs as pre-cursors to an event. Although this approach is able to account for temporal and spatial dependences for event prediction tasks, data sparsity and underlying relational dynamics still remains an unsolved problem. Essentially, many such approaches turn to learning shallow attributes in their attempts to describe complex communication behaviors at static "snapshot" instances of information knowledge graphs [7], [8], [9]. As a result, their predictions generally suffer from inaccuracies arising from coarse temporal resolutions. A concise list of recent event prediction approaches are shown in Table 2. In summary, this research differs from existing mainstream event prediction approaches in the following areas:

(i) Firstly, this research analyses tweets for complex topic detection, classification and tracking. Existing methods such as [7], [8], [9] bias events towards certain topics in question (i.e. crisis: such as riots, terrorist attacks, floods, etc.).
(ii) Secondly, this research introduces a new RTT framework as a structured theoretical process that quantifies the evolution of learned relational features over past causals in the event prediction task. To the best of our knowledge, there are no existing approaches which have used a socially relational approach to address the problem of event prediction. Existing techniques such as [37], [38], [39], [9] reduce complex relational features and states into shallow attributes that result in inaccurate predictions of events.
(iii) Thirdly, this research introduces a new practical extended architecture that is capa-

ble of leveraging on the design developed in [1] for unsupervised event prediction in a continuous stream of social transactions. To the best of our knowledge, there is no one similar architecture developed in mainstream approaches like [40], [41], [37] which leverages on the efficient adaptive effects of fractal structures toward representing dynamic complexities of observed communicative relational behaviors in OSNs for event prediction tasks.

Table 2. Table of related research on event prediction.

Study	Research	Modality
[38]	Bayesian Method for Event Prediction	emperical bayesian, belief based prediction, regressive binomial event distribution, markovian sequenced model
[42]	Shallow and Deep Learning for Event Relatedness Classification	LSTM approach, event relatedness classification, representation learning, coreference resolution model
[43]	An Empirical Investigation of Different Classifiers, Encoding, and Ensemble Schemes for Next Event Prediction Using Business Process Event Logs	Ensemble based classification, decision trees, rules based classifiers, neural network classifiers, bagging, boosting, random subspace, nested dichotomies, dagging, multi-task learning model
[44]	Lazy event prediction using defining trees and schedule bypass for out-of-order PDES	Lazy strategy on defining trees with scheduled bypass, hedging, lazy event prediction, multi-task learning model
[39]	Multi-scale temporal memory for clinical event time-series prediction	Dynamic patient-state model, predictive EHRS event representation, markovian sequenced model
[37]	Predictive Business Process Monitoring via Generative Adversarial Nets: The Case of Next Event Prediction	Generative Adverserian Networks (GANs), recursive learning, discriminative-generative neural networks, multi-taks learning model
[41]	Events, Event Prediction, and Predictive Processing	Active learning models, event maps, temporal structure, multi-task learning, markovian sequenced, coreference resolution model
[40]	Computer System and Method for Creating an Event Prediction Model	multi-task learning model
[45]	Complex event recognition in the big data era: a survey	Event classification, complexity analysis, multi-task learning model
[9]	Learning Dynamic Context Graphs for Predicting Social Events	Knowledge graph, representation learning, graph convolutional networks, multi-task learning model

3. Preliminaries And Problem Formulation

We begin our problem formulation as follows: given a relational flux F_ϵ, we wish to predict the probability of an event E_φ occuring in the topic context L_ϵ of the social transaction. An important assumption that this study makes is that the order of events are randomly distributed over the sentiments expressed in any given OSN/s. This key assumption derives from the fact that most real-life events are weakly dependent on each other from a sequential occurrence standpoint [23], [35]. Instead, they are highly correlated through key reciprocated relational sentiments to their common topic supersets of interest [5]. To address this complex problem in our study, the extended RFT model leverages on important logical and structural key concepts developed in [1]. A detailed architecture of the FNN designed is given in Figure 3.

Event detection is done from learning relational turbulence profiles over multiple streams of social transaction data. The main approach is two fold. Firstly, we break down initial social streams for topic detection. Then, the detected topics are tracked to determine social interest over time. Secondly, events are identified to be intense topics over a “burstiness” index. Then, the social stream is re-queried for the identified events of interest. From the newly acquired social stream, the key relational feature states between individual actor/entity pairs of the social transaction flux are fed into the FNN as inputs which are then learned to produce the correlating relational turbulence profile. The turbulence profile - τ_{ij} from each actor pair is then passed through a gated recurrent unit (GRU) and both generator and discriminator FNN in an adversarial framework to accurately predict event occurances as key relational turbulence scores.

3.1. *Events*

We begin our approach with the definition of real-life events as bursty time-scaled periods of highly intense, dense and volatile social transaction/s within a given Online Social Platform [46]. In our study, we define events as actual occurances of topic-context related real-world disruptions e within a time period T_ϵ where the number of social transactions N and relational turbulence τ_{rl} scores exceed peak thresholds Th_N and Th_τ respectively. An event peak signal is identified by our model, as large transient spectral energy densities of expectancy violations (EVs) $\frac{\partial E_{rl}}{\partial \tau_{rl}}$ in T_ϵ which leads to the aberration of relational states within the given OSN community. These multi-dimensional peaks (e.g. frequency, sentiment, polarity, reciprocity, etc.) arising from relational turbulence in an online social scene due to the occurrence of an event carries a unique signature pattern defining the stretch and length to the profile of the observed burstiness in information exchange within a given social network [47].

3.2. *Relational Turbulence Model*

The extended RFT model developed in this paper adopts the RTM approach derived in [1]. In the RTM model, relational turbulence $P(\tau_{rl})$ is defined by likelihoods of relational intensity $P(\gamma_{rl})$, relational interference $P(\vartheta_{rl})$ and relational uncertainty $P(\varphi_{rl})$ of the model outputs. Key contributing relational features at the input of the model are annotated confidence ρ_{ij}, salience ξ_{ij} and sentiment λ_{ij} scores. Expressions of relational turbulence in the extesions of the RFT model are derived from relational intensity, uncertainty and interference. They are given by the defining equations (6) - (10) in [1].

3.2.1. *Relational Turbulence Theory*

While the RTM is adequate in offering a cause and effect framework used to model relational turbulence, it is lacking in three substantial areas. The RTM does not offer distinctive processes through which key relational reciprocal features arising from actor uncertainty and interference affect the evolution of relational communication behaviors [19]. Secondly, RTT establishes correlations between a subset of causal relational features to observed sentimental and affective social transaction behaviors, which are missing in the RTM framework [19]. Thirdly, RTT establishes a Markovian construct where specific signature evolution patterns of graphical social transactions are correlated to their corresponding detected event occurrences. The RTM however, only models the time specific relational turbulence profile within an identified relational flux [16].

The RTT framework [19] constrains the three key dyadic relational turbulence parameters $P[\gamma_{rl}, \vartheta_{rl}, \varphi_{rl}]$ in a mixed contribution model of a relationship. These framework constraints define that relationship parameters of dyadic actor uncertainty and interference contribute to episodic a-priors of relational intensity $P(\gamma_{rl})$, communication polarity sgn_{rl}, communication engagement ε_{rl} and reciprocal bias χ_{rl}. This graphical representation is given in Figure 2.

Communication engagement is defined mathematically as:

$$\varepsilon_{rl} \in Hom(\amalg_{\tau \in T} \chi_{ij}, \gamma_{ji}) \tag{1}$$

Such that the communication engagement ε_{rl}, represents the unique isomorphism $\varepsilon_f : \chi \to \gamma$.

3.2.2. ***Expectancy Violation***

We determine Expectancy Violations (EVs) as polar mismatches between expected and actual reciprocates of the Confidence ρ_{ij}, Salience ξ_{ij} and Sentiment λ_{ij} feature scores in actor-actor relationships of social transactions during event occurances. These violations (however small), are a contributing factor to temporal representations of relational turbulence - γ_{rl}, ϑ_{rl} and φ_{rl} [1]. Aberration of relational states occur when some EV critical threshold is breached. This critical threshold differes across specific relationships. The extended RFT model builds on conflict escalation minimization from equations (1) and (2) in [1] to determine the EV threshold of interest.

3.3. ***Topic Detection***

The RFT uses topic detection techniques which are derived from two main branches of main stream topic modeling methods. They are the stochastic and generative models [48]. Generative models use word distribution kernels over topic mixtures generated by the document. It defines word co-occurrences to be mutually inclusive over the generative topics of a given corpora [48]. Mathematically, it is expressed as:

$$P(W_i \cup V_j) = \sum_{i,j \in D} P(W_i) + P(V_j) - P(W_i \cap V_j) \tag{2}$$

Where W_i and V_j represent word co-occurances for all word pairs i, j articulated within a document corpora D. The main weakness of using this approach is that syntactic word choice information is not well handled [49]. Stochastic models on the other hand, statistically infers topics from words in a given corpora. This is achieved from observations of word distributions over a set of key documents to determine posterior likelihood estimations [48]. Mathematically, it is expressed as:

$$P(k|W_{d,u,n}) = \frac{P(W_{d,u,n} \vee k)P(k)}{\sum_{s \in k} P(W_{d,u,n} \vee k)P(k_s)} \tag{3}$$

Where k is the topic assignment of the word $W_{d,u,n}$ in document d over the word passage u with a co-occurance word count n and s is the topic segment in question. A main drawback to this approach is that it makes no real assumptions about how distributions over topics and / or their word frequency correlations are masked. Thus, making exact word-topic inferences intractable and inaccurate [50].

3.3.1. ***Wavelet Transformation***

RFT uses wavelet transforms to identify events from a topic mixture signal. Wavelet transformations enable analysis of signal profiles at specific time scales from full time windows of observations [51]. The method first starts by defining a mother wavelet $\psi(t_{ji})$. Then, it geometrically reconstructs affinine child wavelets from the $\psi(t_{ji})$ signal form. This is achieved by adaptively determining scaling and translation factors ν and ϕ respectively [52]. The process can be mathematically described by:

$$\psi_{\nu,\phi}(t) = \frac{1}{|\sqrt{\nu}|}\psi(\frac{t-\phi}{\nu}) \tag{4}$$

Where $\nu, \phi \in \Re$ and $\nu \neq 0$. Wavelet transformations fall into two broad categories. They are the discrete (DWT) and continuous (CWT) transforms [5]. While the CWT enables smooth detection of slow and continuous varying features, DWT provides a more efficient mechanism of detecting discontinuous signals [47].

3.4. *Fractal Neural Network*

At the core of the extended RFT structural design, this study adopts the Fractal Neural Network (FNN) in [1]. This FNN is used in both discriminator and generator networks of our architecture to determine accurate likelihoods of event predictions from ranked likelihoods of relational turbulence profiles.

3.4.1. *Adversarial Learning*

The framework of our system model is structured within a robust adversarial learning approach [53]. The principle of adversarial training mechanisms leverage on two important stages of the learning model. The first stage implements a generative neural network (NN) architecture which is used to create false positives (posteriors) from a bag of input training samples. This stage is processed in synchronization with calculations of actual truth values shared from the same bag at the input. The second stage involves the use of a discriminative NN design which will then estimate an actual output based on a risk / reward mechanism [54]. The discriminative model estimates outputs based on concatenations of inputs which are derived from both generative and real-valued model outputs [55].

3.5. *The Model Problem*

From a practical viewpoint, wavelet signal structures of an event can be used to match real-time information exchanges in an active stream efficiently [47]. However, when used to predict events, it is highly inaccurate [56]. Furthermore, a key assumption we make in this paper is that the order of events are randomly distributed over the sentiments expressed in any given OSN/s. This key assumption derives from the fact that most real-life events are weakly dependent on each other from a sequential occurrence standpoint [23], [35]. Instead, they are highly correlated through key reciprocated relational sentiments to their common topic supersets of interest [5]. Furthermore, the high costs of mainstream turing learning designs used to generate strong adversarial examples makes training impractical on large scale problems like Twitter, GoogleFeed or LiveJournal [53].

3.6. *Our Model Solution*

Our model effectively tackles the problem of topic drifts and establish soft event footprint evolutions over time. To achieve this, we use the non-parametric mixture model [57] to detect topic models over a continuous time stream of social exchanges on Twitter, GoogleFeed and LiveJournal. Additionally, we have adopted the Discrete Wavelet Transform (DWT) to overcome the problem of solving for an infinite number of coefficients - which is computationally intensive [5]. In our architecture, an adapted hybrid FNN is used to drive predictable, accurate and strong estimations from continuous input data streams [55]. Our approach achieves this objective without the associated heavy computational costs by a dynamic true depth scaling technique. This mechanism is used to build and collaspe affinine structures in the FNN design [21].

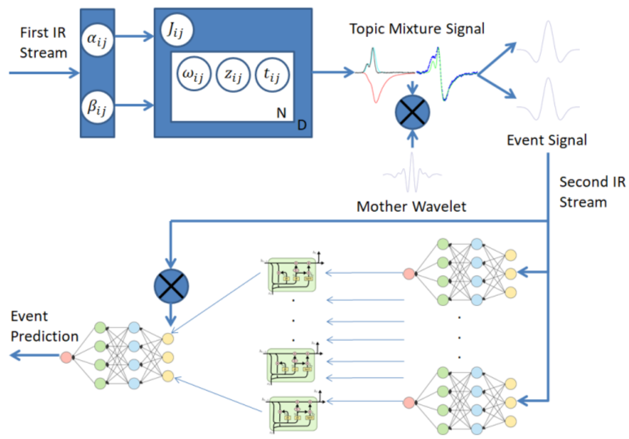

Fig. 1. Event Prediction System Architecture. The diagram shows The different stages of the generalized event prediction process. The first stage identifies topic mixtures, the second stage extracts events from trending topics and the third stage predicts future event occurances.

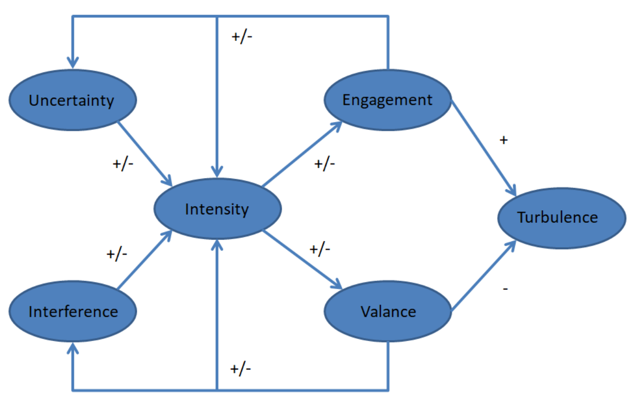

Fig. 2. The simple RTT framework. The diagram shows the construct of how the simple relational turbulene theory is designed and their positive and negative valance affecting interdependent feature entities in relational turbulence.

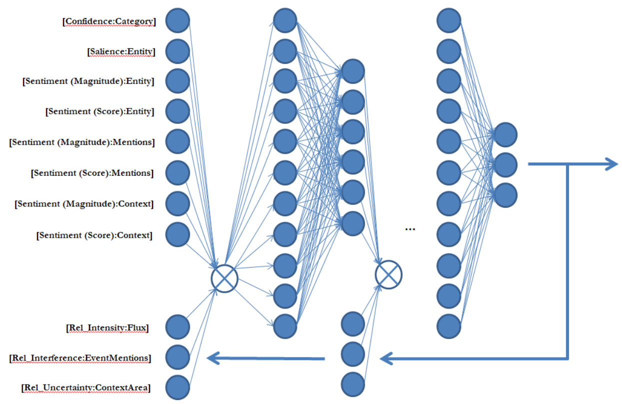

Fig. 3. The RFT Baseline Design. The diagram shows the baseline architecture of a simple convolutional perceptron which was used in the study to build an efficient Fractal Neural Network.

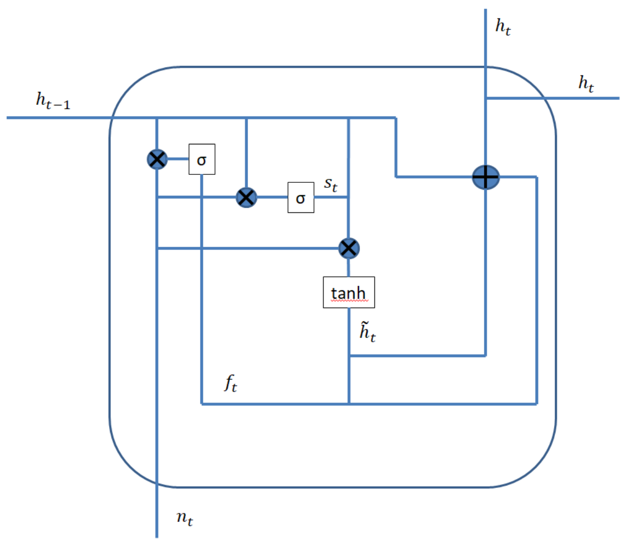

Fig. 4. A GRU Baseline Implementation. The diagram shows the basic design of a Gated Recurrent Unit used in stage three of the RFT architecture.

4. Model and Methods

A high level system architecture of our model is given in Figure 1. Specifically, in our design, data is streamed from online social platform sources (Twitter, GoogleFeed and LiveJournal). In active streaming, the data is continuously pulled from multiple server sources through a data hash pipe using their respective APIs (for GoogleFeed[a], Twitter[b] and LiveJournal

[a]https://developer.feedly.com/v3/search/
[b]https://developer.twitter.com/en/docs.html

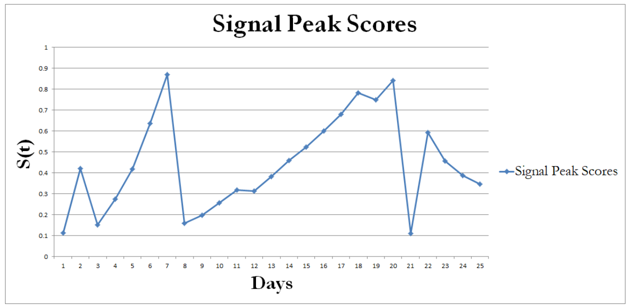

Fig. 5. Peak Event Signal Scores. The diagram shows a typical event topic signal profile.

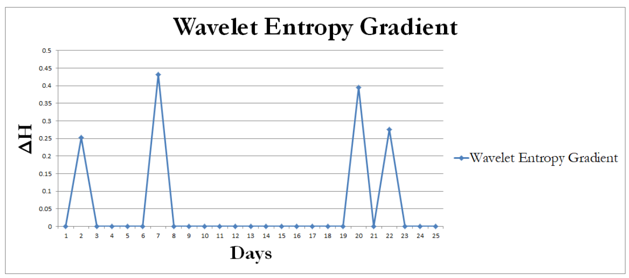

Fig. 6. Change of Wavelet Entropy. The diagram shows the wavelet entropy gradient profile of event topics of interest.

syndicated accounts[c,d]). Incoming streams are filtered according to queries of interest and decoded at the pre-processing stage of our model. This procedure extracts key confidence ρ_{ij}, salience ξ_{ij} and sentiment λ_{ij} scores in a social transaction using Googles NLP API[e]. Firstly, all Information Retrieval (IR) in the first active stream before pre-processing are fed into the first phase of our model. This phase handles topic detection from a contextual corpus of words. The topic model developed in our implementation is derived from a large text corpora by learning the thematic structure of key vocabulary word embeddings [49]. Our approach uses the non-parametric mixture model as a statistical inference mechanism to deduce likelihoods of the underlying topic-word distributions and reject anomalous syntactic word-topic combinations [50]. It is noteworthy of mention that various other word distribution models like Latent Dirichlet Allocation (LDA), correlated topic models, Pachinko allocation, etc. may be assumed and used as drop-in replacements [48], [58], [59]. The second phase of our model performs continuous wavelet transformation on detected topics from the first phase. This technique is used to decompose the topic mixture signal to uncover unique localized predicate signals for key anchor words in a time-scaled domain [5]. After signal decomposition, peak detection is then performed on these topic mention frequencies [47]. Following peak

[c]https://fetchrss.com/api
[d]https://www.livejournal.com/syn
[e]https://cloud.google.com/natural-language/

detection, an LDA inference mechanism based on the Gibbs sampling approach is then used to detect and identify events from the extracted topics in the first IR text stream [59]. Once the events have been identified and their unique contextual key word signatures learnt, these events are then re-queried in a second IR data stream. The continuous information flow received from this query allows us to focus our application of the RTT framework onto specific detected events of interest. The second IR stream is then pre-processed from the query. From the pre-processing phases, the key relational feature vectors ρ_{ij}, ξ_{ij} and λ_{ij} are fed into a three stage FNN to predict the occurrence of the queried event. In the first stage, relational turbulence features like Intensity $P(\gamma_{rl})$, Interference $P(\vartheta_{rl})$ and Uncertainty $P(\varphi_{rl})$ are learnt over the occurrences of past events. Then, in the second stage, output from each first stage FNN are fed into a Gated Recurrent Unit (GRU) structure where cell states of the Long Short Term Memory (LSTM) are constantly updated with merged forget and input gates [60]. Finally, in the third stage of the FNN architecture, the GRU cell states containing long term memory structure of peak turbulence features at the hidden layers are retrieved. They act as concatenations to turbulence inputs of streaming social transactions about an interesting event. The discriminator FNN at this stage is then used to predict the likelihood of when an event will occur.

4.1. *The RFT Design*

We begin the design of our model with the definition of a soft kernel used to discover a markovian structure which we then encode into confabulations of fractal sub-structures. For a given set of data observables as inputs: $\chi \in X$ and outputs: $\Xi \in \Im$ we wish to loosely define a mapping such that the source space (X, α) maps onto a target space $(\Im, \omega)$. The conditional $P(\chi \vee \omega)$ assigns a probability from each source input χ to the final output space in ω. Each posterior state-space from in between input to output is generated and sampled through a random walk process. It is worth noting that markovian random walks are used to build a more generalized stochastic discovery process in our experiment. However for larger datasets, any one of the more sophisticated markovian sampling methods (e.g. Gibbs, Monte carlo, Metropolis-Hastings, Hamiltonian, etc.) can be used as drop-in replacements. An indicator function which we have chosen to describe the state transition rule is:

$$\Theta_{t+1} = min \begin{cases} 0 \\ \amalg_{c=1}^{n} \frac{\delta E_{t+1}^{c}}{\delta \chi_{t}^{c}} \end{cases} \tag{5}$$

Where δE_{t+1}^{c} is the error change from one hidden feature activity state $h_t \in H$ onto higher posterior confabulations. The objective function at each transition seeks to minimize error gradients to eliminate problems associated with exploding and vanishing gradients during backpropagation. This can be caused by an excessive generation of layered confabulations which leads to unnecessary increments in depth from the markovian ANN discovery mechanism. For a general finite state space markovian process, the markov kernel is thus defined as:

$$Kern(M) = \begin{cases} p : X \times \omega \to [0, 1] \\ p(\chi|\omega) = \oint_{\omega} q(\chi, \Im)\nu(\delta\Im) \end{cases} \tag{6}$$

Once a unique markovian neural network has been discovered, a Single Layer Convolutional Perceptrion (SLCP) is proposed as a baseline structure to learn the fractal sub-network from pre-existing posterior confabulations. The SLCP baseline structure (Figure 3) changes as discovered knowledge is progressively encoded during the learning process. Although for simplicity we have used the novel SLCP architecture as our baseline schema; in reality however, any one baseline model can be used to learn a morphing transposition into a fractal signature structure. In essence, methods like Progressive Neural Networks (PNNs) where activation links of neighboring DNN stacks are learned laterally across hidden layers [61] or the wide use of summarizing information from ensemble methods like distillation [62] are relevant alternatives.

From the viewpoint of a DNN architecture, we can define an input to output transition broadly as:

$$Y_n = KX_n + B \tag{7}$$

Where n corresponds to the number of layers in the stack, Y_n is the expected output, X_n is the data at the input and B is the network bias. K is a unique signature of transpositions of hidden activities from source to destination data spaces which we wish to capture and encode into RFT. For an N-deep neural network, K can be expressed as a chained state of lower to upper activation weights:

$$\prod_{n=1}^{N} U_n^T \sigma W_n^T \tag{8}$$

Where T is the target tensor, U^T are the upper layer target vector weights and W^T are the lower layer weights. In our model, σ (sigmoid logistic function) was arbitrarily chosen as the default activation for each neuron. Since posterior hidden activities $h_n \in H$ are known, then an error derivative which converges to zero gives:

$$U = (HH^T)^{-1}HT^T \tag{9}$$

Which essentially states that upper layered weights maintain a canonical property of identical symmetry and reversibility about each hidden layer activity H and the target T. The hidden layer encoding technique was built using the following optimal maximum entropy constraint:

$$min_{P(\psi|\phi)} - H(P) = \sum_{\psi_e,\psi_z} \bar{P}(\phi_z)P(\psi_e|\phi_z)\log P(\psi_e|\phi_z) \tag{10}$$

Where ψ_e are the feature transposed structure states and ϕ_z are the given posterior hidden feature weights. The transposed structure state ψ is given simply by the mathematical relation as:

$$\sum_{e=1}^{n} \psi_e = \sum_{z=1}^{i} \Theta_z \phi_z \oplus k_z \psi_z + \rho_z \tag{11}$$

Where k_z is the morphing constraint on the trans-positioned structural states ψ_z in z posteriors and ρ_z is the normally distributed transpositioned errors as: $\rho_z\ N(0, Q_z)$. Such that Q_z is the error covariance matrix. The estimation of $P(\psi|\phi)$ that minimizes $H(P)$ is done using the Lagrange parameters ς. The convex solution is then estimated as:

$$P_\varsigma(\psi|\phi) = \frac{1}{D_\varsigma(z)} \exp(\sum_{n=1}^{F} \varsigma_n f_n(z,e)) \tag{12}$$

Where n denotes the number of features measured in a dataset. Once feature entropies have been encoded into the fractal subnetwork, this structure is then used to generate depths for highly sophisticated model representations. This is done through de-quantization of the entropy decoded (expanded) model as:

$$G^{k'}(\psi|\phi) = G^{k}(\psi|\phi) \times k(\psi, \phi) \tag{13}$$

Where $G^{k'}(\psi, \phi)$ is the de-quantized graphical representation of the confabulations to the sophistication levels of a feature tensor from the previously encoded fractal structure $G^{k}(\psi|\phi)$. A safe stopping condition is triggered when error gradients approaches zero for expectations to converge. The theoretically desired condition is given as:

$$E(Y) = P(Y|X) \forall \frac{\delta E_n^c}{\delta \chi_n^c} \rightarrow 0 \tag{14}$$

Where $\frac{\delta E_n^c}{\delta \chi_n^c}$ is the error gradient of each confabulation at every epoch.

4.2. *The First Phase*

In the first phase, RFT uses topic detection techniques which are derived from two main branches of main stream topic modeling methods. They are the stochastic and generative models [48]. Our approach uses the non-parametric mixture model [57] as a statistical inference mechanism to deduce likelihoods of the underlying topic-word distributions and reject anomalous syntactic word-topic combinations [50], [63]. Given a continuous stream of contextual information exchanges forming the corpora, we assume that each time-batched social transaction $d_j \in D$, contains a unique set of tokens $\zeta_i \in Z$ over a Hierarchical Dirichlet Process (HDP). Each token atom is defined to be constructed from an ordered pair of word-time primitives. We further assume that the temporal variation of each word-topic pair follows a multi-modal distribution given mathematically as:

$$P(k_{n,t}|\zeta_{n,t}) = (\sum_{s=1}^{S} P(s \vee \zeta_n) P(k_{nt} \vee \zeta_{nt}, s)) \tag{15}$$

Where $P(k_{n,t}|\zeta_{n,t})$ is the conditional probability that topic k (an unknown prior) is chosen for the token ζ. Our topic inference mechanism is based on a Markov Chain Monte Carlo (MCMC) estimation process built around HDP mixtures [50] to efficiently label topics according to a Dirichlet distribution process over a shared relational hierarchy of parent-child topic segments. We establish that each token atom ζ, is associated to an atom specific decision $\delta(\zeta)$ - which maximizes likelihoods of a topic assignment to a token and minimizes errors of the relationship made with the label. In turn, each $\delta(\zeta)$ is correlated to the word observed from the document-token indexed pair. Additionally, each $\delta(\zeta)$ is also correlated to the time window of the document-token indexed observation pair. The sampling τ_{ji} over the time dirichlet distribution process $G(t_{ji})$ is described mathematically by our model as:

$$P(\tau_{ji} = t_{ji}|t^{-ji}, k) \propto \begin{cases} n_{W_{ji}}^{-jt} & \text{In a previous time step} \\ \alpha_0 & \text{Otherwise} \end{cases} \tag{16}$$

Where $n_{W_{ji}}^{-jt}$ is the token word W_{ji} frequency , over an observation time window t and α_0 is the scaling factor of the word dirichlet distribution.

4.3. *The Second Phase*

In the second phase, RFT uses wavelet transforms to identify events from a topic mixture signal [47]. Wavelet transformations enable analysis of signal profiles at specific time scales from full time windows of observations [51]. The method first starts by defining a mother wavelet $\psi(t_{ji})$. Then, it geometrically reconstructs affinine child wavelets from the $\psi(t_{ji})$ signal form. This is achieved by adaptively determining scaling and translation factors ν and ϕ respectively [52]. In this phase, wavelet transformation is used to identify events after a topic mixture (TM) has been successfully sampled from the previous step. Since each topic category already contains words with high likelihoods of association with it, we are only interested in separating specific events from general themes of discussion in our mixture. We analyse TM signal profiles at specific time scales from the full time window of the document-token indexed observation pair t_{ji}. This is done over all tokens ζ_i of the social transaction batch d_i of interest. The resolution of this transform is adaptively adjusted according to approximations of sharp discontinuities from forest (wide window) to trees (small window). These approximations allow us to separate specific events from general themes of discussion. From equation 5, we then classify token-topic signals in a given time scale as:

$$S_k(t) = \frac{N_k(t)}{N(t)} \times \log \frac{\sum_{x=1}^{T} N(x)}{\sum_{x=1}^{T} N_k(x)} \tag{17}$$

Where $N_k(t)$ is the number of transactions which contain the token ζ referenced by the topic k and appears within the time window $(t-1 \leq t \leq t+1)$. $N(t)$ measures the total number of transactions within that same time window. The fraction $\frac{N_k(t)}{N(t)}$ denotes the salience of a topic k over the sampling time scale of interest. While the second term on the right measures the inverse log-linear relationship between the number of times a topic has been referenced to from a token taken as the target observation. Essentially, the second term acts as a linear scaling factor to filter out false peaks. This means that topics which have less mentions within a sample window of transactions are disregarded as general discussion themes (with low signal - $S_k(t)$ peak scores) while topics with more mentions within that same sample time window are identified as potential occurring events (with high signal - $S_k(t)$ peak scores) within the same time window in question. Since events are characterized by short "bursts" of token-topic mentions over a time window of tweets, we can easily identify if a topic belongs to an event or general discussion theme by analyzing both signal peak scores $S_k(t)$ and entropy changes of a wavelet [46], [64]. We use the normalized form (H-measure) of entropy measure to identify changes in token-topic wavelet entropy across longer observation time frames. This is given mathematically as:

$$\triangle H(S_k(t)) = \begin{cases} \frac{H_{t+\triangle} - H_{t-1}}{H_{t-1}} & \text{For } H_{t+\triangle} > H_{t-1} \\ 0 & \text{Otherwise} \end{cases} \tag{18}$$

Where H is the H-measure of the topic signal $S_k(t)$ and $\triangle H$ is the H-measure entropy.

4.4. *The Third Phase*

In this phase of the model, we apply the three stage FNN architecture to predict the likelihood of event occurances.

4.4.1. *GA-FNN*

The macro-framework of our system model is structured within a robust adversarial learning approach [53]. The principle of adversarial training mechanisms leverages on both generative and discriminative stages of the learning model. The generative stage uses a generative Neural Network (NN) design to create false positives (posteriors) from a bag of input training samples and truth values. The discriminative stage involves the use of a discriminative NN design which will then estimate an actual output based on a risk / reward mechanism [54] from both generative and real-valued model outputs [55].

The Generative FNN. For the first stage generative DNN architecture, a Convolutional Recurrent Network (CRN) [65] fractal is used to learn from pre-existing posterior confabulations. We define an input to output transition broadly as:

$$Y_n = KX_n + B \tag{19}$$

where n corresponds to the number of layers in the stack, Y_n is the expected output, X_n is the data at the input and B is the network bias. K is a unique signature of transpositions of hidden activities from source to destination data spaces which we wish to capture and encode into RFT. We determine a safe stopping condition for the growth of confabulation depth layers. This is triggered when error gradients approaches zero for expectations $E(Y)$ to converge. The theoretically desired condition is given as:

$$E(Y) = P(Y|X) \qquad \forall \qquad \frac{\delta E_n^c}{\delta \chi_n^c} \to 0 \tag{20}$$

Where $\frac{\delta E_n^c}{\delta \chi_n^c}$ is the error gradient of each confabulation at every epoch and $P(Y|X)$ is the conditional probability of estimation between input X and output Y.

The LSTM Structure. The second stage RFT architecture involves taking outputs from the first stage FNN framework and remembering them as cells to a larger LSTM structure. Given a hidden layer h_{t-1} from a learned FNN architecture, we wish to remember the output activations and weights of the confabulations at the peak of the episodic social turbulence attributed to the occurrence of a past event. The design of the LSTM structure is built with three gated functions that allow pass-through or blocking of convolutions from both episodic confabulations and current inputs which act as updates to the cell committing these confabulations to long term memory. This first sigmoid gate resets old, in favor of new information that is learned from the eventful social transaction stream. The new information here corresponds to higher peaks in turbulence features learned from the first stage RFT. Mathematically, this can be written as:

$$s_t = \sigma(W_s.[h_{t-1}, n_t]) \tag{21}$$

Where r_t is the output of the reset gate (volatile memory), which is driven by the single layer neural network weights W_s on the convolutions of both LSTM cell inputs (h_{t-1}) and external confabulations (n_t). The next gate is the update gate which acts as both input and forget gates of traditional LSTM models. The update gate is driven mathematically as:

$$f_t = \sigma(W_f.[h_{t-1}, n_t]) \tag{22}$$

Where f_t is the output of the update gate (persistent memory), which is driven by the single layer neural weights W_f on convolutions of past LSTM cell memories and current inputs. This gate decides which new information is important to add and what old information is unimportant and gets thrown away. Finally, the last gate is the output gate. This gate decides what the next hidden state should be from memories stored in the LSTM cell. It is given mathematically as:

$$h_t = (1 - f_t) * h_{t-1} + f_t * \tilde{h_t} \tag{23}$$

where,

$$\tilde{h_t} = tanh(W.[S_t * h_{t-1}, n_t]) \tag{24}$$

Here, $\tilde{h_t}$ represents the updated cell memory. A detailed structure of our GRU implementation is given in Figure 6.

The Discriminative FNN. Finally, in the third stage of the RFT architecture, hidden confabulation states remembered by the GRU are then passed onto the last discriminative FNN architecture which is built on a single layer convolutional perceptron fractal. The confabulation prediction outputs of the social transactions during episodic events from the GRU are concatenated to the confabulations from inputs of the model in a current social stream of transaction data calculated from equations 3, 4 and 5. Mathematically, this is represented as:

$$H_t = A(W_t.(h_t \oplus h_{t-1})) + B \tag{25}$$

Where H_t is the next hidden layer activity, A is the activation function, W_t are the hidden layer weights, $h_t \oplus h_{t-1}$ is the concatanate of both (event) episodic and current hidden layer activities and B is the prediction bias. This is given mathematically as:

$$B = \begin{cases} \frac{h_t - h_{t-1}}{h_{t-1}} \\ 0 \end{cases} \tag{26}$$

This bias is translated as gradient changes across the hidden confabulation layers.

5. Experimental Implementation

5.1. *Experimental Data*

The experiments were conducted on three datasets using RFT and five different baseline algorithms. The datasets are: Twitter, GoogleFeed and LiveJournal. We have chosen these three datasets because they are widely benchmarked throughout the academic circle for studies in sentimental computing and can be easily understood by the audience of this paper.

The first is the LiveJournal dataset streamed from syndicated accounts. We use the top 20 syndicated community sites to retrieve real time feeds from the RSS API. We use snopes[f],

[f]https://www.snopes.com

slashdot[g], google blog[h], reuters[i] and BBC news[j] accounts to capture realtime RSS news feeds and comment posts on international developing topics of debate. The dataset collectively contains approximately 1 billion feeds generated from about 300 external sources from the five chosen syndicated accounts streamed from 03 September 2018 to 31 August 2019. The data feed stream was broken down using googles NLP API to provide the inputs we require of our training model.

the second is the Google[k]news feed dataset obtained from the repositories of feedly cloud. The GoogleFeed dataset allows us to demonstrate the capability of RFT in generalizing relational turbulence profiles across a hetrogeneous graphical structure. The dataset was streamed with the feedly API and broken down from the extracted feed contents. Specifically, in this instance the feedly cloud - which processes over 50 million feeds per hour was used to extract GoogleFeed metadata and contains approximately 668522179 entities and 9923167883 dyads accumulated from 03 September 2018 to 31 August 2019.

The last dataset is streamed from twitter in real-time using the twitter streaming API[l]. In our experiment, our twitter dataset was streamed for 756 million tweets across 253 million nodes and over 1 billion links - cumulatively from 03 September 2018 to 31 August 2019. The Twitter4J contains APIs[m]for classifying raw tweets that allows us to integrate their classifiers into our deep learning model. Their plug-in module allows us to stream tweets continuously over a span of time and their filters allowed us to query tweets by geo-locality so that we were able to detect interesting evolving events. In addition to the sentiment results obtained from their model, we cross validated the output against googles NLP API to replicate the most accurate sentiment scores and magnitudes of context spaces and mentions.

5.2. *Experimental Model Baselines*

We consider several state-of-the-art methods for comparison with our proposed RFT model. Since our model is the first in line for this type of adaptive online event predictive approach, we use modified versions of similar methods along with the baselines we developed earlier for comparison. Another notable point is although many prediction models exist, not all methods have the same goal or data features as ours. Therefore we consider only the models which use similar data to ours for comparison. It should be mentioned that not all the methods can both predict events and profile communication patterns together. Therefore we compare only the event prediction outputs between each other. The short description of the competing methods are given below:

Association Rules Prediction (ARP) [35] is a rule based approach which predicts future events from past observations of event sequences. This baseline enables us to directly

[g]https://m.slashdot.org
[h]https://www.blog.google.com
[i]https://reuters.com
[j]https://www.bbc.com/news
[k]https://news.google.com
[l]https://developer.twitter.com/en/docs.html
[m]http://help.sentiment140.com/api

model underlying conditional probabilities. For our experiments, we use the Bayesian shrinkage estimator to adjust the confidence of rule probabilities which are conditionally dependent on previous event occurances in a sequence. This "adjustable" confidence technique allows highly accurate, frequently apprearing rules of sequential events to be chosen for the prediction probabilities of an event.

Compositional Neural Network (CoNN) [36] is a model which extracts knowledge of event sequences from text. This method is built around neural composition representations of word arguments and predicates to events. In our experimental baseline, word vectors from news feeds, posts or tweets about an event is fed into the neural network model. Argument compositions are then generated as hidden layer activities using the tanh function. The higher event composition layers are then used to produce a coherence score which measures the likelihood that two events belong to the same sequence. This measure is then used to rank the predictions of future events of interest from currently developing events.

Hidden Markov Model (HMM) [66] is a method used to predict future event occurances based on current latent markovian relationships of observed event development stages. This baseline method leverages on characteristic patterns of event occurance sequences prior to a future time frame $t + \triangle t$. For this baseline model, we extract the sequence of event development stages from past news feeds and posts. Using the HMM model, we then characterize the event development stages using normalized event mentions. We define the burstiness of an event as the ratio of event mentions to the total number of mentions within a time window t_w of interest. From a 7-day moving average, we then deteremine a state transition probability matrix to predict the likelihood of a future event occurance.

Hybrid Probabilistic Markovian (HPM) [2] is a mixed probabilistic time-series model that captures trends and periodicity of associated events. This baseline approach leverages on an autocorrelation function to classify event time sequences as periodic, partially-periodic, trend-based or random. In our experiments, we capture historic time series y of detected peaks Q of an event in question. We then use the time series periodicity determined by lags in the autocorrelation function to compute the probability of future event peak occurances P in a temporal frame t of interest. The prediction of an event at a future instance y_t is then calculated as the average of the peak profiles which were historically detected for all $t \in P$.

Generalized Logit Model (GLM) [67] is a logistic regression technique which is built from semantic knowledge of text sentiment representations. This baseline method directly models topic distributions of events which are extracted using the Latent Dirichlet Allocation (LDA) technique. As a three step process, this method first extracts events using Semantic Role Labeling (SRL). Topic distributions of the extracted event in the text are then discovered using LDA. Finally, events are predicted using linear regression. In our experiments, we use the topic distributions about past event occurances from the first phase of our model (after the pre-processing stage) to predict the likelihood of future occurances. We achieve this using the baseline logistic regression model with parameters $\beta_0 ... \beta_n$ estimated from apriori eventful datasets.

GDELT (True Value) The GDELT repository[n] is used to choose significant events from April to August 2019. The dates at which the events took place are used as truth values against the models prediction. The event topics we have selected from this dataset include: "One Belt One Road", "Terrorist Attack", "Trade Tariff Cuts", "Mexico Border" and "Pacific Hurricane". Due to space constraints, only three of the chosen five events will be discussed in the section of experimental results.

Relational Flux Turbulence (RFT) is the model which we have developed in this paper that builds on the FNN learning framework. Using our model, we deploy the key relational features of our interest and dynamically adjust depth of the architecture according to the complexity of learning at the inputs. RFT uses both the RTM inspired framework and FNN to recursively build a fractal during active online learning, which is then used to predict the likelihoods of event occurances at the output.

5.3. *Experimental Design*

Figure 3 describes the inputs into our model. Specifically, the RFT model accepts as inputs, the confidence of the detected category in every social transaction, the Salience of all detected entities in the transaction, the sentiment scores and magnitudes of entities, mentions and drifting contexts. These eight relational features form the key independent input into our RFT fractal neural network (FNN) model. Additionally, the outputs (Relational Intensity γ_{rl}, Relational Interference ϑ_{rl} and Relational Uncertainty φ_{rl}) which represent turbulence are fed back into the model as recurrent inputs into the neural network to act as memory retention for the relational turbulence profiles of previous transaction/s, and as good influential initialization points for new training sequences of extracted sentiments in later social transactions.

Relational Turbulence was calculated from conditional posteriors of γ_{rl}, ϑ_{rl} and φ_{rl} as the mathematical relation of:

$$P(\tau_{rl}) = \sum_{i=1}^{n} \frac{P(\gamma_i|\theta_i)P(\vartheta_i|\varphi_i)P(\varphi_i|\gamma_i)}{N_i P(\gamma_i)P(\vartheta_i)P(\varphi_i)} \tag{27}$$

The inputs were tested across the RFT dynamically stacked Fractal Neural Network (FNN) and our chosen baseline models. The learning results were compared using both F1 score and K-fold cross validation to measure both accuracy and performance of the prediction. We define positive predictions as likelihood scores for the occurance of an event in question, which have been estimated by RFT and our baseline models to be at 0.7 or higher. The "truth" at the timeframe (+/-1 day) of this prediction is cross validated against archived events extracted from GDELT. The actual training data was from 03 September 2018, to 01 April 2019 and the test data from 02 April, 2019, to 31 August, 2019.

[n]https://www.gdeltproject.org/data.html#rawdatafiles

5.4. *Parameter Settings*

In our experiments, we choose our system model parameters based on the combined effect of several factors - including errors in observational data, choices of calibration methods and Design Of Experiment (DOE) criterias [68]. We use a hybrid of both global and local Sensitivity Analysis (SA) approaches to determine and specify the best performing parameters for experimentation based on a predefined behavior threshold for our model. Firstly, we generate a sample set of parameters at 0.1% intervals within realistic operating ranges of each parameter setting. For example, for learning rate, we generate a set of parameters $0.0, 0.1, 0.2, 0.3, ..., 10.0$ across a feasible space. Then, we use the PSUADE [69] open source software to analyze input-output relationships of statistical and sensitivity metrics. These metrics are obtained from the varying attribution of output distributions due to changing model parameters with inputs being kept constant [70].

As determined by our approach, our experiments were conducted on our training model with a learning rate set to 1.1, a sliding window set to 3, an error tolerance set to 0.1 (10%), a data outlier threshold set to 1.0, with scaling set to 10, a vanishing gradient error threshold at 0 and an exploding gradient error threshold set to 100. Finally, our trust region radius parameter was set to 5 and our softmax temperature regularization parameter at 1.2.

6. Analysis and Discussions

6.1. *Experimental Findings*

The tests were run across the baselines and our RFT model. For clarity and simplicity of explainations, approximately every 10000th running data from a chosen testing output sample set is plotted on a graph and displayed for discussion purposes. Additionally because of space constraints, only the F1 score results of experiments conducted on the chosen events are shown. The results are displayed in Figure 7 - 12:

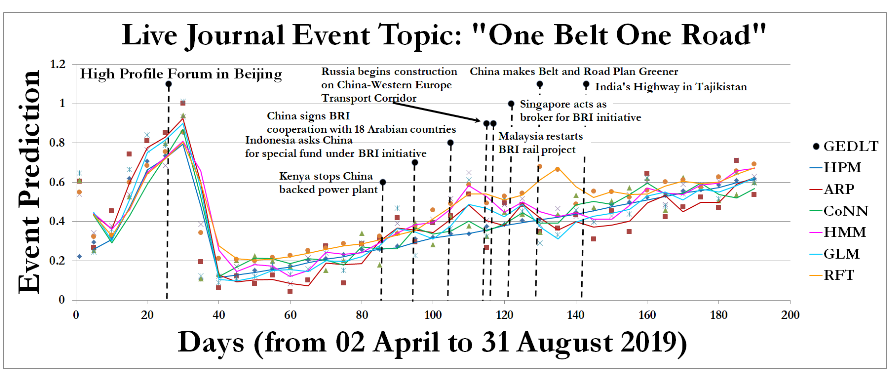

Fig. 7. Figure of Live Journal Event Prediction.

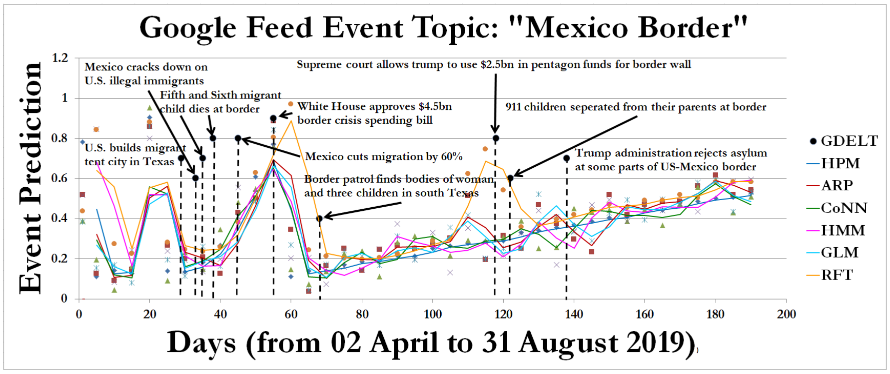

Fig. 8. Figure of Google Feed Event Prediction.

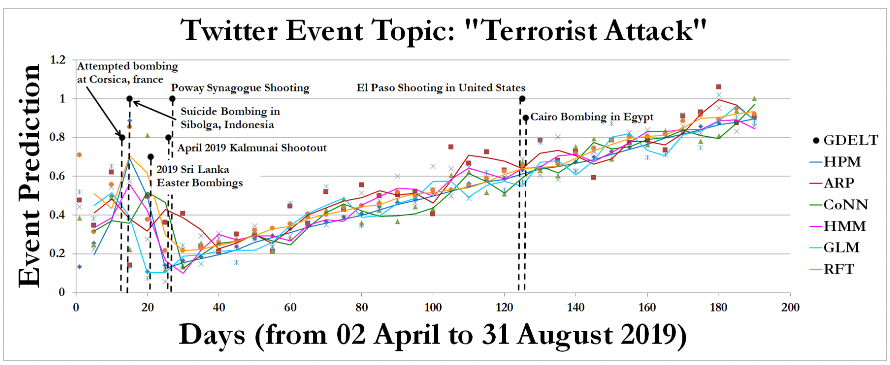

Fig. 9. Figure of Twitter Event Prediction.

6.2. *Performance Measurements*

The F1-score test was conducted on the results obtained from the experiments. We chose this benchmark because importantly, the F1 score measures both contributions of precision and recall as important metrics to access the performance of the RFT prediction to the baseline models prediction of future events. Specifically, the F1 score is given as:

$$F1 = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \tag{28}$$

Where

$$Precision = \frac{(TruePositives)}{(TruePositives + FalsePositives)} \tag{29}$$

And

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{30}$$

Where precision in our experimental design corresponds to the accuracy of predicting actual true positives and recall measures against the dependability of the predictions.

6.3. *Testing Results*

Live Journal "One-Belt-One-Road" Events F1-Score				
	Confusion Matrix		Precision / Recall	F1 - Score
HPM	TP = 50	TN = 492	P = 0.641	0.629
	FP = 28	FN = 31	R = 0.617	
ARP	TP = 95	TN = 397	P = 0.621	0.662
	FP = 58	FN = 39	R = 0.709	
CoNN	TP = 321	TN = 118	P = 0.772	0.801
	FP = 95	FN = 64	R = 0.834	
HMM	TP = 284	TN = 130	P = 0.721	0.760
	FP = 110	FN = 69	R = 0.805	
GLM	TP = 241	TN = 155	P = 0.655	0.709
	FP = 127	FN = 71	R = 0.772	
RFT	TP = 56	TN = 20	P = 0.835	**0.848**
	FP = 11	FN = 9	R = 0.862	

Fig. 10. Table of F1 score between baseline models and RFT event prediction for the Twitter validation dataset averaged over k cross validations.

Google Feed "Mexico Border" Events F1-Score				
	Confusion Matrix		Precision / Recall	F1 - Score
HPM	TP = 292	TN = 448	P = 0.689	0.640
	FP = 132	FN = 197	R = 0.597	
ARP	TP = 205	TN = 359	P = 0.406	0.445
	FP = 300	FN = 211	R = 0.493	
CoNN	TP = 398	TN = 402	P = 0.755	0.757
	FP = 129	FN = 127	R = 0.758	
HMM	TP = 329	TN = 489	P = 0.770	0.731
	FP = 98	FN = 144	R = 0.696	
GLM	TP = 386	TN = 106	P = 0.596	0.561
	FP = 262	FN = 342	R = 0.530	
RFT	TP = 79	TN = 32	P = 0.859	**0.819**
	FP = 13	FN = 22	R = 0.782	

Fig. 11. Table of F1 score between baseline models and RFT event prediction for the LiveJournal validation dataset averaged over k cross validations.

Twitter "Terrorist Attack" Events F1-Score				
	Confusion Matrix		Precision / Recall	F1 - Score
HPM	TP = 443	TN = 1342	P = 0.580	0.676
	FP = 321	FN = 104	R = 0.810	
ARP	TP = 404	TN = 542	P = 0.542	0.574
	FP = 342	FN = 257	R = 0.611	
CoNN	TP = 577	TN = 872	P = 0.640	0.682
	FP = 324	FN = 213	R = 0.730	
HMM	TP = 500	TN = 1325	P = 0.626	0.699
	FP = 299	FN = 131	R = 0.792	
GLM	TP = 262	TN = 1179	P = 0.547	0.515
	FP = 217	FN = 227	R = 0.486	
RFT	TP = 101	TN = 69	P = 0.673	**0.737**
	FP = 49	FN = 23	R = 0.815	

Fig. 12. Table of F1 score between baseline models and RFT event prediction for the GoogleFeed validation dataset averaged over k cross validations.

K	δ_{MAPE} – (Live Journal)	δ_{MAPE} – (Google Feed)	δ_{MAPE} – (Twitter)
20	0.219	0.385	0.563
30	0.289	0.434	0.554
50	0.223	0.394	0.450
80	0.227	0.352	0.431
100	0.188	0.277	0.392

Fig. 13. Table of K-fold cross validated MAPE for all learning models.

Finally, during the experimentation, the full datasets obtained from the different sources (twitter, google and live-journal) were partitioned into k-subsamples. One of the subsamples was retained as the validation set for each run and the validation set was chosen in a round robin fashion for subsequent experimentation runs. A noteworthy point of mention is that K fold cross validation is used in our experimentation design to obtain a good estimate of the prediction generalization. This testing technique does not scale well to measurements of model precision. How accurately a learning model is able to predict an expected output is based on the F1-scores [71]. K-fold validation was performed over all data streaming sources learnt and predicted by the RFT framework across the Mean Absolute Percentage Error (MAPE) measurement of each run. Mathematically, MAPE can be expressed as:

$$\delta_{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{E_i(x) - Y_i(t)}{E_i(x)}\right| \tag{31}$$

Where $E_i(x)$ is the expectation at the output at data input set i and $Y_i(t)$ is the corresponding prediction over N total subsamples. The tabulation of the K-fold cross validation used in our experimentation is given in Figure 13.

6.4. *Investigation Of Results*

As can be seen from the graphs, our RFT model measures comparably well to the other baseline models for future event occurrences. Additionally, across all sources of information streams, it is capable of measuring high F1 scores over events which have been recognized by our framework from a mixed set of detected topics.

As can be readily observed from Figures 10 to 12, prediction over events like terrorist attack and mexico border do not fare as well as other events like one belt one road. This can intuitively be attributed to the fact that terrorist attacks, mexico border topics and their associated events have either not occurred or that have a high degree of uncertainty in their occurances. As such, positive examples for such past occurrences have been sparse and difficult to acquire and train our model adequately with.

Generally however, it can be observed that from Figure 13, as the number of sub-sample windows increases over the dataset, the MAPE over all data sources decreases considerably. This means that a longer continuous training sample set will produce more accurate results from the total bag size of samples. Thus, it can be intuitively inferred that the longer the time frame spent on learning a continuous stream of social exchanges, the more accurate the prediction of events will be for topics which are currently tracked.

6.5. *Parameter Influence*

RFT is a complex architecture which contains several critical hyperparameters. We study these parameters from two key perspectives of the model. 1) the FNN dynamic depth scaling and 2) the regularization. Our validation and verification (V&V) strategy involves varying one parameter while fixing others in turn - to evaluate their influences. Our results show that RFT is capable of stabilizing performance over a specified range of parameter settings.

The first parameter we evaluated was the dynamic depth scaling threshold of our FNN model. We varied this across the range from 1 to 20. The tabulated Precision, Recall and F1 scores from the predictions of events (measured against GDELT) were plotted on the graph in Figure 14. As can be seen from the results, RFT generally predicts more accurately as scaling thresholds are gradually increased from 1 to 9. The performance of the results stabalizes after a threshold of 10. Specifically, we obtained the best results when the scaling threshold is tuned to 10. A slight dip in performance after this threshold is indicative of overfitting since more depth in the model creates more complexity. Hence, a scaling threshold of 10 offers the best model performing results.

The second parameter we analyzed was the effect of regularization (which controls overfitting by penalizing the model if weights become too large or many) on the models prediction performance. We fix the scaling threshold parameter to 10 and adjusted the regularization parameter from 0 to 3 in increments of 0.1. An illustration of the resulting prediction Precision, Recall and F1 scores are given in Figure 15. We observe that generally, peak performance scores tend to fall within a tight range between 1 to 1.5. The results suffer when regularization is tuned outside this range, which is indicative of both underfitting and overfitting of data. Hence, it proves that a regularization of parameter setting around the region of 1.2 is sufficient to stabilize the performance of our model results.

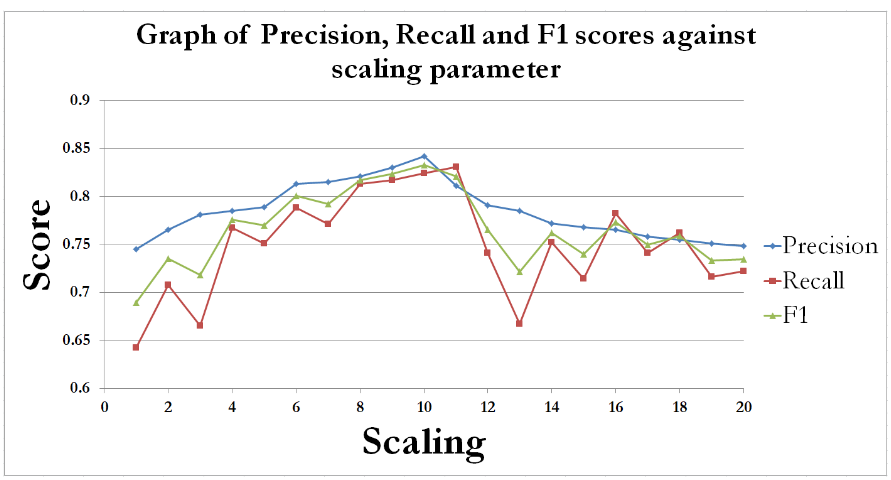

Fig. 14. Performance on different dynamic depth scaling thresholds.

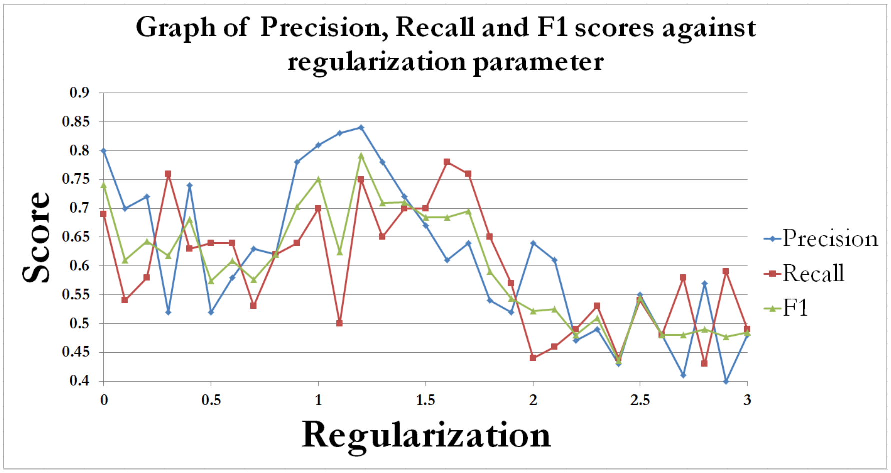

Fig. 15. Performance on different regularization parameters.

6.6. *Ablation Study*

An ablation study was done on the key input parameters to the RFT model. Our approach verifies the influence of these parameters on prediction performance by occluding the relational parameters (confidence ρ_{ij}, salience ξ_{ij} and sentiment λ_{ij} scores) in turn. Figure 16 shows our results. As mentioned in the problem formulation: with the consideration of relational turbulence (which relates to relational reciprocity bias, sentimental and affective communication patterns, state altering events and role-recognition behaviors), RFT is capable of making accurate predictions of event occurances. As evidenced by the significant decreases in model prediction performance in Figure 16, we note that ρ_{ij}, ξ_{ij} and λ_{ij} are all important contributing parameters to measures in Precision, Recall and F1.

Method	Precision	Recall	F1
RFT (w/o ρ_{ij})	0.592	0.709	0.714
RFT (w/o ξ_{ij})	0.746	0.682	0.702
RFT (w/o λ_{ij})	0.727	0.737	0.752
RFT (full model)	**0.897**	**0.796**	**0.838**

Fig. 16. Ablation of our proposed RFT model.

7. Conclusion

In conclusion, our research provides new insights into event prediction from a relational intelligence perspective that results in more accurate predictions over time. Our results show that the FNN model is capable of learning adaptively to the complexity of information received in real-time. Our study uncovers three pivotal long-term objectives from a relational perspective. Firstly, relational features can be used to strengthen medical, cyber security and social applications where the constant challenges between detection, recommendation, prediction, data utility and privacy are being continually addressed. Secondly, in fintech applications, relational predicates (e.g. turbulence) are determinants to market movements - closely modeled after a system of constant shocks. Finally, in artificial intelligence applications like computer cognition, robotics and neuromorphs, learning relational features between social actors enables machines to recognize and evolve.

Acknowledgements

The authors wish to thank the reviewers for their helpful comments. This work was developed in collaboration with Zhejiang Joint Laboratory Ivy institute of technology R&D Program, Applipro Construction AI Research Labs, Shanghai Municipal Science and Technology Major Project and ZJLab.

References

1. L. Tan, T. Pham, H. K. Ho, and T. S. Kok, "Discovering relational intelligence in online social networks," in *International Conference on Database and Expert Systems Applications*. Springer, 2020, pp. 339–353.
2. G. Amodeo, R. Blanco, and U. Brefeld, "Hybrid models for future event prediction," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 1981–1984.
3. G. M. Weiss and H. Hirsh, "Event prediction: learning from ambiguous examples," in *Working Notes of the NIPS9298 Workshop on Learning from Ambiguous and Complex Examples*, 1998.
4. Y. Yang, T. Pierce, and J. Carbonell, "A study of retrospective and on-line event detection," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998, pp. 28–36.
5. M. Cordeiro and J. Gama, "Online social networks event detection: a survey," in *Solving Large Scale Learning Tasks. Challenges and Algorithms*. Springer, 2016, pp. 1–41.
6. X. Dong, D. Mavroeidis, F. Calabrese, and P. Frossard, "Multiscale event detection in social media," *Data Mining and Knowledge Discovery*, vol. 29, no. 5, pp. 1374–1405, 2015.
7. K. Feng, G. Cong, C. S. Jensen, and T. Guo, "Finding attribute-aware similar regions for data analysis," *Proceedings of the VLDB Endowment*, vol. 12, no. 11, pp. 1414–1426, 2019.
8. X. Huang, Q. Song, Y. Li, and X. Hu, "Graph recurrent networks with attributed random walks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 732–740.
9. S. Deng, H. Rangwala, and Y. Ning, "Learning dynamic context graphs for predicting social events," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1007–1016.
10. L. Zhao, J. Wang, and X. Guo, "Distant-supervision of heterogeneous multitask learning for social event forecasting with multilingual indicators." in *AAAI*, 2018, pp. 4498–4505.
11. L. Zhao, Q. Sun, J. Ye, F. Chen, C.-T. Lu, and N. Ramakrishnan, "Multi-task learning for spatio-temporal event forecasting," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1503–1512.
12. Y. Ning, R. Tao, C. K. Reddy, H. Rangwala, J. C. Starz, and N. Ramakrishnan, "Staple: Spatio-temporal precursor learning for event forecasting," in *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 2018, pp. 99–107.
13. X. Zhang, L. Zhao, A. P. Boedihardjo, C.-T. Lu, and N. Ramakrishnan, "Spatiotemporal event forecasting from incomplete hyper-local price data," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 507–516.
14. F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Computational*

Intelligence, vol. 31, no. 1, pp. 132–164, 2015.
15. D. Knights, M. C. Mozer, and N. Nicolov, "Detecting topic drift with compound topic models." in *ICWSM*, 2009.
16. L. K. Knobloch and J. A. Theiss, "Relational turbulence theory applied to the transition from deployment to reintegration," *Journal of Family Theory & Review*, vol. 10, no. 3, pp. 535–549, 2018.
17. S. C. Hoi, D. Sahoo, J. Lu, and P. Zhao, "Online learning: A comprehensive survey," *arXiv preprint arXiv:1802.02871*, 2018.
18. D. H. Solomon and L. K. Knobloch, "Relationship uncertainty, partner interference, and intimacy within dating relationships," *Journal of Social and Personal Relationships*, vol. 18, no. 6, pp. 804–820, 2001.
19. D. H. Solomon, L. K. Knobloch, J. A. Theiss, and R. M. McLaren, "Relational turbulence theory: Variation in subjective experiences and communication within romantic relationships," *Human Communication Research*, vol. 42, no. 4, pp. 507–532, 2016.
20. K.-I. Goh, G. Salvi, B. Kahng, and D. Kim, "Skeleton and fractal scaling in complex networks," *Physical review letters*, vol. 96, no. 1, p. 018701, 2006.
21. G. Larsson, M. Maire, and G. Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals," *arXiv preprint arXiv:1605.07648*, 2016.
22. K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2392–2396.
23. S. Petrovic, "Real-time event detection in massive streams," *Data Mining and Knowledge Discovery*, vol. 29, no. 5, pp. 1374–1405, 2013.
24. H.-O. Peitgen, H. Jürgens, and D. Saupe, *Chaos and fractals: new frontiers of science*. Springer Science & Business Media, 2006.
25. D. H. Solomon and J. A. Theiss, "A longitudinal test of the relational turbulence model of romantic relationship development," *Personal Relationships*, vol. 15, no. 3, pp. 339–357, 2008.
26. R. M. McLaren, D. H. Solomon, and J. S. Priem, "The effect of relationship characteristics and relational communication on experiences of hurt from romantic partners," *Journal of Communication*, vol. 62, no. 6, pp. 950–971, 2012.
27. W. Wilmot, J. Hocker, J. Arthur, J. W. Petty, J. J. Martocchio, H. R. Cheeseman, E. Biech, and R. J. Rosania, *Interpersonal Conflict 9th Edition*. McGraw-Hill Higher Education, 2007.
28. D. H. Solomon and J. A. Samp, "Power and problem appraisal: Perceptual foundations of the chilling effect in dating relationships," *Journal of Social and Personal Relationships*, vol. 15, no. 2, pp. 191–209, 1998.
29. C. A. Surra, "Reasons for changes in commitment: Variations by courtship type," *Journal of Social and Personal Relationships*, vol. 4, no. 1, pp. 17–33, 1987.
30. L. A. Baxter and G. Pittman, "Communicatively remembering turning points of relational development in heterosexual romantic relationships," *Communication Reports*, vol. 14, no. 1, pp. 1–17, 2001.
31. L. Simeonova, "Gradient emotional analysis," in *Proceedings of the Student Research Workshop associated with RANLP*, 2017, pp. 41–45.
32. J. A. Theiss and D. H. Solomon, "A relational turbulence model of communication about irritations in romantic relationships," *Communication Research*, vol. 33, no. 5, pp. 391–418, 2006.
33. D. O. Braithwaite and P. Schrodt, *Engaging theories in interpersonal communication: Multiple perspectives*. Sage Publications, 2014.
34. J. Zhang, L. Tan, X. Tao, D. Wang, J. J.-C. Ying, and X. Wang, "Learning relational fractals for deep knowledge graph embedding in online social networks," in *International Conference on Web Information Systems Engineering*. Springer, 2019, pp. 660–674.
35. C. Rudin, B. Letham, A. Salleb-Aouissi, E. Kogan, and D. Madigan, "Sequential event prediction with association rules," in *Proceedings of the 24th annual conference on learning theory*, 2011, pp. 615–634.

36. M. Granroth-Wilding and S. Clark, "What happens next? event prediction using a compositional neural network model," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
37. F. Taymouri, M. La Rosa, S. Erfani, Z. D. Bozorgi, and I. Verenich, "Predictive business process monitoring via generative adversarial nets: The case of next event prediction," *arXiv preprint arXiv:2003.11268*, 2020.
38. J. Tuke, A. Nguyen, M. Nasim, D. Mellor, A. Wickramasinghe, N. Bean, and L. Mitchell, "Pachinko prediction: A bayesian method for event prediction from social media data," *Information Processing & Management*, vol. 57, no. 2, p. 102147, 2020.
39. J. M. Lee and M. Hauskrecht, "Multi-scale temporal memory for clinical event time-series prediction," in *International Conference on Artificial Intelligence in Medicine*. Springer, 2020, pp. 313–324.
40. M. Jermann and J. P. Boueri, "Computer system and method for creating an event prediction model," Jul. 30 2020, uS Patent App. 16/256,992.
41. J. Hohwy, A. Hebblewhite, and T. Drummond, "Events, event prediction, and predictive processing," *Topics in Cognitive Science*, 2020.
42. J. Haneczok and J. Piskorski, "Shallow and deep learning for event relatedness classification," *Information Processing & Management*, p. 102371, 2020.
43. B. A. Tama, M. Comuzzi, and J. Ko, "An empirical investigation of different classifiers, encoding, and ensemble schemes for next event prediction using business process event logs," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 6, pp. 1–34, 2020.
44. D. Mendoza, Z. Cheng, E. Arasteh, and R. Domer, "Lazy event prediction using defining trees and schedule bypass for out-of-order pdes," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 885–890.
45. N. Giatrakos, E. Alevizos, A. Artikis, A. Deligiannakis, and M. Garofalakis, "Complex event recognition in the big data era: a survey," *The VLDB Journal*, vol. 29, no. 1, pp. 313–352, 2020.
46. X. Wang, C. Zhai, X. Hu, and R. Sproat, "Mining correlated bursty topic patterns from coordinated text streams," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 784–793.
47. J. Weng and B.-S. Lee, "Event detection in twitter," in *Fifth international AAAI conference on weblogs and social media*, 2011.
48. D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.
49. L. Du, W. Buntine, and M. Johnson, "Topic segmentation with a structured topic model," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 190–200.
50. A. Dubey, A. Hefny, S. Williamson, and E. P. Xing, "A nonparametric mixture model for topic modeling over time," in *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 2013, pp. 530–538.
51. C. K. Chui, *An introduction to wavelets*. Elsevier, 2016.
52. M. Cordeiro, "Twitter event detection: combining wavelet analysis and topic inference summarization," in *International Workshop on Algorithms and Models for the Web-Graph*, 2011, pp. 113–125.
53. A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *arXiv preprint arXiv:1904.12843*, 2019.
54. A. Kumar, A. Biswas, and S. Sanyal, "ecommercegan: A generative adversarial network for e-commerce," *arXiv preprint arXiv:1801.03244*, 2018.
55. L. Cai and W. Y. Wang, "Kbgan: Adversarial learning for knowledge graph embeddings," *arXiv preprint arXiv:1711.04071*, 2017.
56. F. Jin, W. Wang, P. Chakraborty, N. Self, F. Chen, and N. Ramakrishnan, "Tracking multiple social media for stock market event prediction," in *Industrial Conference on Data Mining*. Springer, 2017, pp. 16–30.
57. E. M. Airoldi, D. M. Blei, E. A. Erosheva, and S. E. Fienberg, "Introduction to mixed membership

models and methods." *Handbook of mixed membership models and their applications*, vol. 100, pp. 3–14, 2014.
58. H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, and L. Zhao, "Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey," *Multimedia Tools and Applications*, pp. 1–43, 2017.
59. M. Hoffman, F. R. Bach, and D. M. Blei, "Online learning for latent dirichlet allocation," in *advances in neural information processing systems*, 2010, pp. 856–864.
60. K. Cho, B. van Merrienboer, aglar Gülehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *EMNLP*, 2014.
61. J. Gideon, S. Khorram, Z. Aldeneh, D. Dimitriadis, and E. M. Provost, "Progressive neural networks for transfer learning in emotion recognition," *arXiv preprint arXiv:1706.03256*, 2017.
62. G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.
63. J. Li, Y. Song, Z. Wei, and K.-F. Wong, "A joint model of conversational discourse and latent topics on microblogs," *Computational Linguistics*, vol. 44, no. 4, pp. 719–754, 2018.
64. Y. Rao, H. Xie, J. Li, F. Jin, F. L. Wang, and Q. Li, "Social emotion classification of short text via topic-level maximum entropy model," *Information & Management*, vol. 53, no. 8, pp. 978–986, 2016.
65. L. Feng, S. Liu, and J. Yao, "Music genre classification with paralleling recurrent convolutional neural network," *arXiv preprint arXiv:1712.08370*, 2017.
66. F. Qiao, P. Li, X. Zhang, Z. Ding, J. Cheng, and H. Wang, "Predicting social unrest events with hidden markov models using gdelt," *Discrete Dynamics in Nature and Society*, vol. 2017, 2017.
67. X. Wang, M. S. Gerber, and D. E. Brown, "Automatic crime prediction using events extracted from twitter posts," in *International conference on social computing, behavioral-cultural modeling, and prediction.* Springer, 2012, pp. 231–238.
68. Y. Gan, Q. Duan, W. Gong, C. Tong, Y. Sun, W. Chu, A. Ye, C. Miao, and Z. Di, "A comprehensive evaluation of various sensitivity anslysis methods: A case study with a hydrological model," *Environmental Modelling & Software*, vol. 51, pp. 269–285, 2014.
69. C. Tong, "Psuade user's manual," *Lawrance Livermore National Laboratory (LLNL), Livermore, CA*, vol. 109, 2005.
70. H. Hsieh, "Application of the psuade tool for sensitivity analysis of an engineering simulation," Lawrance Livermore National Lab.(LLNL), Livermore, CA (United States), Tech. Rep., 2007.
71. P. Flach and M. Kull, "Precision-recall-gain curves: Pr analysis done right," in *Advances in neural information processing systems*, 2015, pp. 838–846.