Better Prevent than React: Deep Stratified Learning to Predict Hate Intensity of Twitter Reply Chains

Dhruv Sahnan*, Snehil Dahiya*, Vasu Goel*, Anil Bandhakavi† and Tanmoy Chakraborty*
*Dept. of CSE, IIIT-Delhi, India; †Logically, UK
Email: {*dhruv18230, snehil19046, vasu18322, tanmoy}@iiitd.ac.in, †anil@logically.co.uk

2021 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-2398-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICDM51629.2021.00066

***Abstract*—Given a tweet, predicting the discussions that unfold around it is convoluted, to say the least. Most if not all of the discernibly benign tweets which seem innocuous may very well attract inflammatory posts (hate speech) from people who find them non-congenial. Therefore, building upon the aforementioned task and predicting if a tweet will incite hate speech is of critical importance. To stifle the dissemination of online hate speech is the need of the hour. Thus, there have been a handful of models for the detection of hate speech. Classical models work *retrospectively* by leveraging a *reactive* strategy – detection after the postage of hate speech, i.e., a backward trace after detection. Therefore, a benign post that may act as a surrogate to invoke toxicity in the near future, may not be flagged by the existing hate speech detection models.**

In this paper, we address this problem through a *proactive* strategy initiated to *avert* hate crime. We propose `DRAGNET`, a deep stratified learning framework which predicts the *intensity of hatred* that a root tweet can fetch through its subsequent replies. We extend the collection of social media discourse from our earlier work [1], comprising the entire reply chains up to $\sim 5k$ root tweets catalogued into four controversial topics . Similar to [1], we notice a handful of cases where despite the root tweets being non-hateful, the succeeding replies inject an enormous amount of toxicity into the discussions. `DRAGNET` turns out to be highly effective, significantly outperforming six state-of-the-art baselines. It beats the best baseline with an increase of 9.4% in the Pearson correlation coefficient and a decrease of 19% in Root Mean Square Error. Further, `DRAGNET`'s deployment in Logically's advanced AI platform designed to monitor real-world problematic and hateful narratives has improved the aggregated insights extracted for understanding their spread, influence and thereby offering actionable intelligence to counter them

***Index Terms*—Hate intensity prediction, reply chain, social media, hate speech, stratified learning.**

I. Introduction

In the era of the Internet, undoubtedly the biggest crisis is how to curb the proliferation of polarising content and hateful material on online social media. Hate crimes around the world have intensified owing to the rampant onslaught of provocative content; there have been violent incidents ranging from lynchings [2] to even ethnic cleansing. While a handful of corporate firms are continuously attempting to reconfigure the terms and conduction of social media usage, they are also constrained by domestic laws on censorship.

Prior art and limitations. The urgent need to combat hate speech has been recognized, resulting in a handful of methods to *detect* online hate speech [3]. Attempts have also been made to study how hate speech *breeds* on social media [4] – how to mathematically model hate speech propagation [5], who is likely to engage in hate speech [6], and so on. However, as Liu et al. [7] highlighted, the aforementioned methods consider a *reactive* strategy – given an online post, these methods attempt to detect whether it is hate speech or not; sometimes, they also classify a hateful post into a fine-grained category (offensive, provocative, aggressive, etc.). One may think of developing methods for the *early detection* of hate speech; however, it is unknown how expeditiously its adverse effects manoeuvre and manipulate online users upon exposition. The promptness and the extent of damage that an online post can cause is hard to quantify and predict. Our recent work [1] captures the early detection through forecasting future trend of hate speech propagation on the fly. However, it does not help in detecting hate propagation in distant future, since it only captures hate propagation for just the near future in real time.

Motivation. The retrospective nature of the state-of-the-art hate speech detection models has been a major bottleneck for the content moderators to intervene before the online hate crime takes place. A better strategy would be to *proactively prevent* the hate crime from happening, in place of letting it materialize and then detecting it. Moreover, within social media discourse, we often notice that the semantics of a benign post undergoes a series of transpositions, resulting in the manifestation of hate speech [1]. A conversation around a benign post often bypasses the lens of the content moderators, and therefore receives ample opportunity for inviting hate speech perpetrators to adversely act upon it. Figure 1 shows the *hate intensity* (defined in Section III) of the reply chain of tweets related to 'Donald Trump's COVID crisis' and 'Joe Biden's campaign'. Two perceptible observations are as follows: (i) The root tweets and their initial replies are not very hate intensive; however, as the discussions progress, the hate intensity score rises. (ii) There is no consistent pattern across the hate intensity profiles of reply chains from two topics. The former observation confirms why a proactive (prevention) strategy is a requisite, while the latter shows the non-triviality of the hate intensity prediction problem. Furthermore, we believe that combating online hate speech is not as simple as a binary classification problem (hate vs not-hate); rather, it is about the *intensity of hatred* that a post can exhibit. Predicting the intensity of hatred that stems from the root tweet and its reply chain is crucial for a content moderator to prioritize which reply chain needs greater manual inspection and further intervention.

978-1-6654-2398-4/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDM51629.2021.00066

Proposed problem and solution. Our problem definition is as follows [1] – *given a root tweet and a few of its initial replies, can we predict the hate intensity of the upcoming replies of the tweet*?

We begin by suitably quantifying the hate intensity of a tweet (and a reply chain) and then converting a reply chain into a sequence of hate intensities [1]. We then propose a novel model, called DRAGNET. It works underneath a stratified learning framework which aims to capture the heterogeneity amongst the hate intensity profiles of reply chains and to categorize them into homogeneous clusters/strata. We train DRAGNET to predict the weights representing the cluster representative probabilities. The prior knowledge vector formulated by unifying the cluster information and the predicted weights, is used in cohesion with the new reply chain to predict the hate intensity of upcoming replies.

Experiments. We curate a novel dataset of $\sim 5k$ root tweets and their complete reply chains, consisting of ~ 1.1 million tweets. A thorough analysis on the curated dataset reveals the characteristics of the heterogeneous hate intensity profiles. We then present a detailed comparative analysis of DRAGNET against six baselines (adopted to the current setting). DRAGNET proves to be the best model, outperforming the strongest baseline with a 9.4% higher Pearson correlation, a 19% lower Root Mean Square Error (RMSE) and a 6.6% lower Mean Forecast Error (MFE) (for latter two metrics, lower value is better). Further analysis on the implication in the change of model parameters and various ablation studies help us diagnose DRAGNET.

Real-world deployment. DRAGNET has been deployed in an advanced AI platform by Logically, a technology startup designed to monitor real-world problematic and harmful narratives in social media data streams. We observe that DRAGNET offers complementary insights for ranking and profiling narratives (semantically aggregated content clusters) in terms of hate speech. These insights can be potentially combined with other influence analytics about narratives to formulate useful counter measures to minimise the impact and damage caused by their viral spread.

Contributions. Our major contributions are five-fold:

- We attempt to solve a novel problem – predicting hate intensity of reply chains on Twitter [1].
- We propose a new model, DRAGNET that leverages a stratified learning paradigm for hate intensity prediction.
- We extend the large-scale dataset [1] containing complete Twitter discussion threads.
- DRAGNET turns out to be highly effective, outperforming six baselines significantly.
- DRAGNET has been deployed by a tech startup in its advanced AI platform to combat mis-informative, problematic and harmful narratives.

Reproducibility. The source codes of DRAGNET and all the baselines, and the dataset are available at the following link: https://github.com/LCS2-IIITD/DRAGNET_ICDM21.

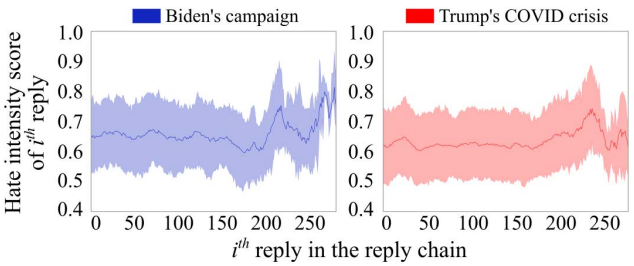

Fig. 1: Temporal change of hate intensity score in the reply chains related to two topics – Joe Biden's campaign and Donald Trump's COVID crisis. Solid lines (shaded regions) signify the average (confidence) over related reply chains.

II. Related Work

Studies on hate speech. Plethora of techniques have been proposed for the detection of online hate speech in various languages [8]–[10]. These methods range from lexicon-based logistic regression [11]–[13] to multi-modal systems [14]. Overcoming the use of static hate-lexicon by prototypical learning has also been explored [15]. Note that the objective of our research is not to propose a new hate detection model; however, this work depends on a hate detection model (Section VIII-B). Readers are encouraged to go through [16] for a detailed survey on hate detection models. Mathew et al. [4] performed an exploratory analysis of Gab [17], and established the presence of strong ties (cohesiveness) among users participating in hateful content. Masud et al. [5] further proposed a predictive model to determine the followers who are more likely to retweet a hateful post. Another aspect of user interaction is the reply chain. Our analysis of reply threads on Twitter uncovered a nuanced and fluctuating sentiment of hate and non-hate, which we explore in this work.

Time-series models. Since the hate intensity of a reply chain can be mapped as a time series, we consider several time-series forecasting (TSF) models as baselines. The conventional methods for TSF such as ARMA [18], including exponential smoothing and linear space models, achieved superlative performance. Deep learning based models such as CNNs, RNNs and LSTMs have become popular for TSF because they do not assume prior structure of data. RNNs [19] are capable of retaining information of preceding results; however, their performance degrades with the increasing length of the sequence. LSTM [20] overcomes the long-term dependency problem to a certain extent. Some of the recent studies used sequence-to-sequence (Seq2Seq) models which align with our interest of multi-step forecasting [21], [22]. New architectures were proposed for the same in [23], [24] to remit the error buildup in multi-step forecasting. Transformer-based models like N-Beats [25] have also been successful for time-series data.

To handle uncertainty in real-world time-series data, many probabilistic forecasting models were proposed, one of them being DeepAR [26]. Some studies leverage Generative Adversarial Network (GAN) to estimate prediction distributions using cVAEs [24], cGANs [27] and ForGAN [28]. However,

Symbol	Description
φ	A tweet
c_i^{φ}	i^{th} reply to the tweet
$\mathcal{R}_{p,q}$	p^{th} datapoint where q denotes the length of its hate intensity profile
$\mathcal{R}^*{}_{p,q}$	The recreated p^{th} datapoint of the dataset from the autoencoder with q denoting the length of its hate intensity profile
$\mathcal{S}_{s_{(1,n)}}$	Sequence of cosine similarity values
$\mathcal{R}_{s_{(1,n)}}$	Set of all hate intensity profiles
j	Number of clusters
δ	Window size
t_h	Number of replies in history
t_f	Index of the last reply in the reply chain
n	Maximum length of the reply chain
s	Total number of reply chains
N_{X_h}	Size of encoded history latent vector
N_{X_f}	Size of encoded future latent vector
$\mathcal{X}_h$	Encoded history latent vector
$\mathcal{X}_f$	Encoded future latent vector
$\mathcal{X}^*{}_f$	Predicted future latent vector
$\mathcal{X}$	Latent vector of $\mathcal{R}_{p,q}$
$\mathcal{X}^*$	Latent vector of $\mathcal{R}^*{}_{p,q}$
C_c	List of cluster centres
$\mathcal{P}(C_{ci})$	Likelihood of belongingness to i^{th} cluster identified with C_{ci}
$\mathcal{X}^c$	Prior knowledge
$\mathcal{P}^*(C_{ci})$	Predicted weight for i^{th} cluster centre C_{ci} to calculate $\mathcal{X}^c$
$\mathcal{X}_d$	Pre-processed prior vector
$\mathcal{X}_{hc}$	Intermediate prediction vector
$\mathcal{GM}(.)$	Clustering model
$\mathcal{PR}(.)$	Classification model
$\mathcal{FP}(.)$	Prediction model
$\mathcal{FP}_d(.)$	1^{st} segment of prediction model
$\mathcal{FP}_p(.)$	2^{nd} Prediction model
$\mathcal{D}(.)$	Decoder model

TABLE I: Notations and denotations.

they fail to encode the multiplicity in time series. Recently, we proposed DESSERT that captures uncertainty as well as has functional approximation ability to predict hate intensity trend in near future on the fly but, it is limited to trend prediction only at a given point of time based on recent replies of an ongoing reply thread on Twitter [1].

III. Preliminaries

Table I summarizes the denotations of the important notations. Let an ordered sequence of first t replies to a root tweet φ be $\mathcal{T}_{1,t}^{\varphi} = \langle c_1^{\varphi}, c_2^{\varphi}, \ldots, c_t^{\varphi} \rangle$, where c_i^{φ} denotes the i^{th} reply to the tweet. (Note that t denotes an integer index associated to t^{th} reply in the sequence, not the actual/continuous time.) As suggested in [1], for each reply c, we quantify its hate intensity – $\mathcal{H}(.)$ which is a weighted sum of two measures,

$$\mathcal{H}(c) = w\mathcal{H}_c(c) + (1-w)\mathcal{H}_l(c) \tag{1}$$

where w ($0 \leq w \leq 1$) is a hyper-parameter. $\mathcal{H}_c$ refers to the probability that the reply is hateful as indexed by a state-of-the-art hate speech detection model[1] (Section VII). $\mathcal{H}_l$ is defined as the average score for all words in a reply from a model-independent hate lexicon that comprises $2,895$ words (scores are normalised using min-max scaling) as proposed in [29]. Since, $0 \leq \mathcal{H}_c(c) \leq 1$ and $0 \leq \mathcal{H}_l(c) \leq 1$, therefore, $0 \leq \mathcal{H}(c) \leq 1$. Each reply chain $\mathcal{T}_{1,t}^{\varphi}$ can be mapped to a sequence of hate intensities, $\mathcal{H}(\mathcal{T}_{1,t}^{\varphi}) = \{\mathcal{H}(c_1^{\varphi}), \cdots, \mathcal{H}(c_t^{\varphi})\}$. While this mapping can be incorporated via other more complex methods as well, previous studies in hate intensity prediction found this method very effective [1]. We further smooth each such sequence $\mathcal{H}(\mathcal{T}_{1,t}^{\varphi})$ using a *rolling average operation* with window size δ. A **window** is a set of δ consecutive replies to a root tweet φ. The hate intensity of a window consisting of a sequence of replies $\mathcal{T}_{k,k+\delta}^{\varphi}$ for a tweet φ is measured as,

$$\begin{aligned}\mathcal{H}(\mathcal{T}_{k,k+\delta}^{\varphi}) &= \sum_{c \in \mathcal{T}_{k,k+\delta}^{\tau}} \mathcal{H}(c) \\ &= w \sum_{c \in \mathcal{T}_{k,k+\delta}^{\varphi}} \mathcal{H}_c(c) + (1-w) \sum_{c \in \mathcal{T}_{k,k+\delta}^{\varphi}} \mathcal{H}_l(c).\end{aligned} \tag{2}$$

Note that $0 \leq \mathcal{H}(\mathcal{T}_{k,k+\delta}^{\varphi}) \leq \delta$.

Sentiment features: To capture sentimental context flow in a new reply chain w.r.t. the initial tweet, we calculate the cosine similarity between the sentiment embedding of the root tweet φ and its corresponding replies, $c_1^{\varphi}, c_2^{\varphi}, \ldots$, $\mathcal{CS}(c_i^{\varphi}) = CosineSim(Embed(c_i^{\varphi}), Embed(\varphi))$. The sentiment embedding is the second last fully-connected layer from the pre-trained XLNet model [30] for sentiment classification. We apply the same *rolling average operation* to $\mathcal{CS}(\mathcal{T}_{1,t}^{\varphi})$ with the same window size δ as performed on $\mathcal{H}(\mathcal{T}_{1,t}^{\varphi})$. Figure 2 illustrates the complete data preprocessing pipeline.

Problem definition: Given a new tweet φ and the last t_h replies in its reply chain $\mathcal{T}_{1,t_h}^{\varphi}$ (used as a training set or history), we aim to predict the hate intensity of the upcoming replies $c_{t'}^{\varphi}$ (where $t' > t_h$). However, instead of predicting the hate intensity per reply, we consider each window of δ future replies and predict the hate intensity per window, $\mathcal{H}(\mathcal{T}_{t',t'+\delta}^{\varphi})$.

IV. Our Proposed Model

In this section, we explain our proposed method, DRAGNET[2] for hate intensity prediction. DRAGNET is a deep stratified learning [31] approach, which first divides the heterogeneous data points (reply chains in our case) into homogeneous clusters/strata and then trains a deep regressor on each strata to predict the hate intensity. The schematic diagram of DRAGNET is shown in Figure 3.

The training set is formed by the two dimensional vector of window-wise hate intensity profile and sentiment context value sequences,

$$\begin{aligned}\mathcal{R}_{s_{(1,n)}} &= \{\mathcal{R}_{p,q} : 1 \leq p \leq s, 1 \leq q \leq n\} \\ \mathcal{R}_{p,q} &= \{\mathcal{H}(\mathcal{T}_{k,k+\delta}^{\varphi_p}) : 1 \leq k \leq q-\delta\} \\ \mathcal{S}_{s_{(1,n)}} &= \{\mathcal{S}_{p,q} : 1 \leq p \leq s, 1 \leq q \leq n\} \\ \mathcal{S}_{p,q} &= \{\mathcal{CS}(\mathcal{T}_{k,k+\delta}^{\varphi_p}) : 1 \leq k \leq q-\delta\}\end{aligned} \tag{3}$$

where s is the total number of reply chains, and n is the maximum length of the reply chain. The elements $\mathcal{R}_{p,q} \in \mathcal{R}_{s_{(1,n)}}$ and $\mathcal{S}_{p,q} \in \mathcal{S}_{s_{(1,n)}}$ are the p^{th} data point, whose reply chain is of length q (φ_p represents the p^{th} root tweet). DRAGNET starts by first learning low-dimensional latent representations for the hate intensity profile of reply chains (of irregular

[1]We use the Davidson model [12] as the default hate speech classifier. However, we also show the results with other hate speech classifiers in Section VIII.

[2]**D**eep st**R**atified le**A**rnin**G** for hate i**N**tensity of r**E**ply chains on **T**witter

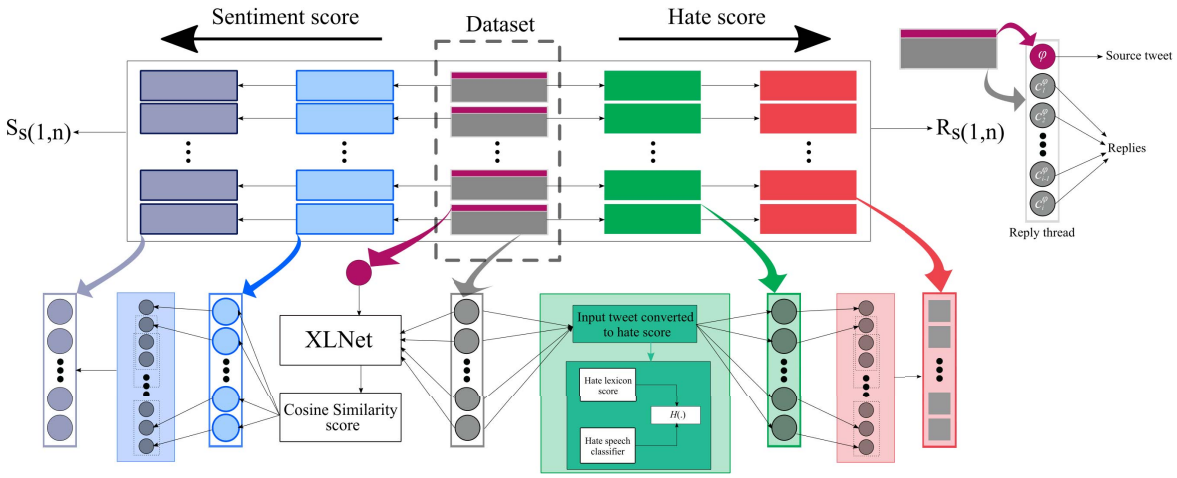

Fig. 2: Schematic diagram of the data transformation module. $\mathcal{S}_{s_{(1,n)}}$ is the sequence of cosine similarity values (calculated between the sentiment embedding of root tweet φ and its corresponding sequence of replies) for all reply chains in the dataset, and $\mathcal{R}_{s_{(1,n)}}$ is the set of all hate intensity profiles.

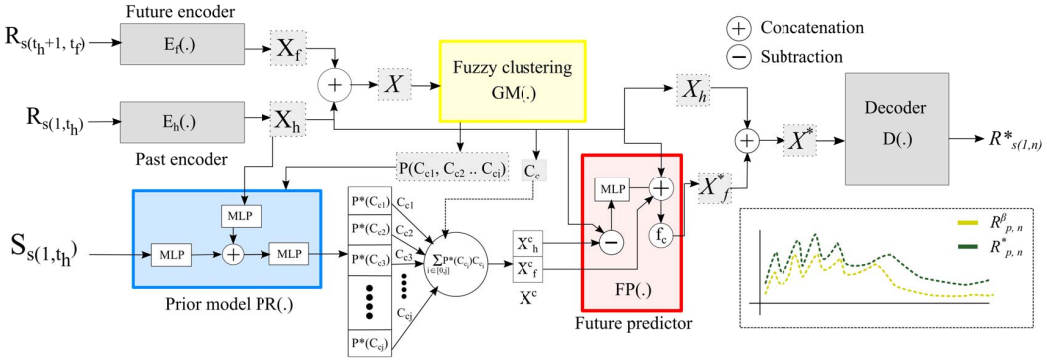

Fig. 3: Overall architecture of DRAGNET. After training the autoencoder, the concatenated history and future latent representations are clustered using a fuzzy clustering algorithm. For a new reply chain, the future hate intensity profile is predicted using (i) the history latent representation, (ii) the sentiment similarity sequence of the history, and (iii) the prior knowledge vector extracted from $\mathcal{PR}(.)$.

lengths) using an autoencoder. In this setting, we learn two separate latent representations – $\mathcal{X}_h$, the initial few replies which are treated as the history, and $\mathcal{X}_f$, the future hate trend for the rest of the replies. We then employ a fuzzy clustering approach in an unsupervised setting to assign cluster membership probabilities $\mathcal{P}(C_{c1}, C_{c2}, .., C_{cj})$ and cluster centres $(C_{c1}, C_{c2}, .., C_{cj})$ to each reply chain, where j is a hyper-parameter indicating the number of clusters. Following this, we train a novel deep neural network unit that predicts the cluster membership probabilities, given $\mathcal{X}_h$ and $\mathcal{S}_{s_{(1,t_h)}}$, which assigns cluster centres for a new reply chain. Finally, a novel deep regressor predicts the latent representation of the future hate trend $\mathcal{X}_f$ using $\mathcal{X}_h$ and $\mathcal{P}(C_{c1}, C_{c2}, .., C_{cj})$, which, when combined with $\mathcal{X}_h$, is converted to the complete hate trend by the decoder trained during the autoencoder phase.

A. Time-series Representative Learning

The reply chain vector $\mathcal{R}_{s_{(1,n)}}$ can be treated as a collection of time series (window-wise hate intensity profiles) with irregular lengths. State-of-the-art methods on clustering irregular time series involve the use of the Dynamic Time Warping (DTW) distance metric and its variants to group similar trends together [32]. Even with favorable outcomes in terms of precision by DTW to map time series similarity, the noisy and volatile nature of the data points in the current study does not allow it to show high efficiency in clustering similar hate trends into a single stratum. To capture a more suitable representation of the time series, we propose an autoencoder to map each reply chain $\mathcal{R}_{p,q}$ in $\mathcal{R}_{s_{1,n}}$ to a low-dimensional latent representation. Additionally, instead of a single encoder-decoder architecture, we propose a multi-encoder approach as proposed in [33].

B. Proposed Autoencoder

We propose an autoencoder using the Inception-Time module [34] for the initial transformation stage. The autoencoder consists of dual encoders to make up the latent space and a single decoder to recreate the complete hate intensity profile from the latent representation.

Encoder. Many studies proposed models for both univariate and multivariate time series [34], [35]. Time-series modeling involves a transformation stage. Instead of manually engineering it, we use the Inception-Time module, denoted as $\mathcal{E}_t(.)$ [34]. It provides a multivariate transformation, which is the concatenation of various uni-variate time series created by using different kernel sizes along the original time series. We represent it as,

$$\mathcal{X}_m = \mathcal{E}_t(\mathcal{R}_{s_{(1,n)}}) \tag{4}$$

where $\mathcal{X}_m \in \mathbb{R}^{s \times n \times 4}$ is the intermediate multivariate representation of $\mathcal{R}_{s_{(1,n)}}$. The intermediate representation $\mathcal{X}_m$ is subjected to the $\mathcal{F}latten(.)$ operator that converts $\mathcal{X}_m$ to a one-dimensional vector $\mathcal{X}_{flat}$. It is followed by layers of linear transformations which are trained to learn the best representative transformation from $\mathcal{X}_{flat}$ and correspondingly produce the final representation of $\mathcal{R}_{s_{(1,n)}}$ in the latent space. If $\mathcal{E}_{lt}$ denotes the layers of linear transformations, we can write,

$$\begin{aligned} \mathcal{X}_{flat} &= \mathcal{F}latten(\mathcal{X}_m) \\ \mathcal{X}_o &= \mathcal{E}_{lt}(\mathcal{X}_{flat}) \end{aligned} \tag{5}$$

Here, $\mathcal{X}_o$ is the final latent space representation of the original time series $\mathcal{R}_{s_{1,n}}$.

Using Equations 4 and 5, we define two encoders $\mathcal{E}_h(.)$ and $\mathcal{E}_f(.)$. We encode two segments of each reply chain, $\mathcal{R}_{s_{(1,t_h)}}$ and $\mathcal{R}_{s_{(t_h+1,t_f)}}$ that are termed as history and future representations respectively, as follows:

$$\begin{aligned} \mathcal{X}_h &= \mathcal{E}_h(\mathcal{R}_{s_{(1,t_h)}}) \\ \mathcal{X}_f &= \mathcal{E}_f(\mathcal{R}_{s_{(t_h+1,t_f)}}) \end{aligned} \tag{6}$$

Here, $\mathcal{X}_h$ and $\mathcal{X}_f$ form the history and future latent representations, which we will use finally to decode to the complete hate trend. Note $\mathcal{X}_h \in \mathbb{R}^{s \times N_{X_h}}$ and $\mathcal{X}_h \in \mathbb{R}^{s \times N_{X_f}}$, where N_{X_h} and N_{X_f} represent the length of low dimensional vectors X_h and X_f respectively.

Decoder. Although we use dual encoders to convert the hate intensity of each reply chain into two latent representations, $\mathcal{X}_h$ and $\mathcal{X}_f$, we use a single decoder $\mathcal{D}(.)$ to convert the latent representation back to the original input. For this, we concatenate the two latent representations and train the decoder to recreate the original hate intensity profile per reply chain. The functioning of the decoder can be denoted as follows:

$$\mathcal{R}^*{}_{s_{(1,n)}} = \mathcal{D}(\mathcal{X}^*) \tag{7}$$

We measure the precision of predictions from the decoder by comparing $\mathcal{R}^*{}_{s_{(1,n)}}$ to $\mathcal{R}_{s_{(1,n)}}$.

C. Fuzzy Associations

As stated in Section IV-A, the hate intensity profiles of reply chains in our dataset are noisy and volatile in nature, and have no evident pattern. Our objective is to use low dimensional latent representations of the data procured via autoencoder (as discussed in Section IV-B) to group similar profiles together. Recent research that align with this approach are an attestation to the validity of performance of the deep learning based models in learning hidden features from time-series data for different tasks [36], [37]. We use a clustering approach over the latent representations to group heterogeneous hate intensity profiles into (near-)homogeneous clusters. In this unsupervised setting, the task of finding meaningful associations in data is unfairly dependent on the number of clusters j and the cluster centres. Therefore, we adopt a fuzzy clustering approach and use the membership probabilities as a feature embedding, rather than confining each profile to a single cluster.

We define the combined latent space $\mathcal{X}$ as,

$$\mathcal{X} = \mathcal{X}_h \oplus \mathcal{X}_f \tag{8}$$

Now, in place of employing a hard clustering approach, i.e., fixing the association of each profile to the closest cluster, we make use of cluster membership probabilities from a fuzzy clustering. The membership probability vector represents the associative probabilities of each cluster with the given chain, denoted by $\mathcal{P}(C_{c1}, C_{c2}, .., C_{cj})$, where C_{ci} denotes the cluster centre of the i^{th} cluster.

Fuzzy Clustering. Since our objective is to find associations of each hate intensity profile to the homogeneous clusters, we perform clustering on the combined latent representation $\mathcal{X}$. We adopt a state-of-the-art fuzzy clustering model [38], denoted by $\mathcal{GM}(.)$ to detect the clusters C_c which is the set of cluster centres as follows:

$$C_c = \mathcal{GM}(\mathcal{X}) = (C_{c1}, C_{c2}, .., C_{cj}) \tag{9}$$

where j is the pre-defined number of clusters, and C_{ci} is the cluster centre of the i^{th} cluster.

D. Boosting Prediction with Prior Knowledge

The task of predicting the hate intensity of upcoming replies, provided limited history $\mathcal{R}_{s_{(1,t_h)}}$, is strenuous even for state-of-the-art deep learning models due to the noisy, volatile and heterogeneous nature of the time-series hate intensity profiles. To address this, we introduce the notion of prior knowledge to the prediction component of our pipeline as the weighted sum of the cluster centres, where the weights correspond to the cluster membership probabilities for the new chain, denoted by $\mathcal{P}^*(C_{c1}, C_{c2}, .., C_{cj})$. We define prior knowledge as follows:

$$\mathcal{X}^c = \sum_{i \in (0 \le i \le j)} C_{ci} \cdot \mathcal{P}^*(C_{ci}) \tag{10}$$

Note that C_c is calculated over $\mathcal{X}$, i.e., the combined latent representation. Therefore, to calculate the complete membership probability vector for a new chain, we cannot use the fuzzy clustering model $\mathcal{GM}(.)$ directly. Rather, we construct a prior model $\mathcal{PR}(.)$ with the aim of predicting the membership probabilities for new chains using only the latent representation of the history $\mathcal{X}_h$ and sentiment feature $\mathcal{S}_{s_{(1,t_h)}}$.

$$\mathcal{PR}\big(\mathcal{E}_h(\mathcal{R}_{s_{(1,t_h)}}), \mathcal{S}_{s_{(1,t_h)}}\big) = \mathcal{P}^*(C_{c1}, C_{c2}, .., C_{cj}) \tag{11}$$

The precision of the predictions by the prior regression model is measured by comparing $\mathcal{P}^*(C_{c1}, C_{c2}, .., C_{cj})$ against $\mathcal{P}(C_{c1}, C_{c2}, .., C_{cj})$.

E. Estimating Latent Representation of Upcoming Reply Chains

Since our ultimate objective is to predict the complete hate intensity profile trend given the history $\mathcal{R}_{s_{(1,t_h)}}$, we need to estimate the latent representation of upcoming future hate intensity profile $\mathcal{X}^*{}_f$. This will finally be fed into the decoder along with the latent representation of the history. For this, we utilize the prior knowledge (as explained in Section IV-D), and the latent representation of the history $\mathcal{X}^c$.

To avoid the estimation task from being overly governed by the prior knowledge, we design the predictor in two stages. The first stage involves the creation of vector i.e., $\mathcal{X}_h^c$ that constitutes the information required to predict $\mathcal{X}^*{}_f$. Since we are only given $\mathcal{X}_h$, we measure the deviation of the prior from the expected only in the latent space where we encode the initial history of hate diffusion, i.e., $\mathcal{R}_{s_(1,t_h)}$. We calculate deviation by first applying the difference operator on the expected ($\mathcal{X}_h$) and the estimated ($\mathcal{X}_h^c$) priors of the reply chain history. We employ a single-layer perceptron, $\mathcal{FP}_d(.)$ to formulate a vector that encodes the dissimilarity between the prior knowledge from history $\mathcal{X}_h^c$ and $\mathcal{X}_h$, which is denoted as $\mathcal{X}_d$.

$$\mathcal{X}_s = \mathcal{X}_h \ominus \mathcal{X}_h^c; \qquad \mathcal{X}_d = \mathcal{FP}_d(\mathcal{X}_s) \tag{12}$$

We finally obtain the $\mathcal{X}_h c$ vector by concatenating the provided input $\mathcal{X}_h$, the pre-processed prior $\mathcal{X}_d$ and $\mathcal{X}_f^c$ as, $\mathcal{X}_{hc} = \mathcal{X}_h \oplus \mathcal{X}_d \oplus \mathcal{X}_f^c$.

The second stage is the deep linear transformation model $\mathcal{FP}_p(.)$ that predicts the upcoming hate intensity in the latent space $\mathcal{X}_f^*$ as follows:

$$\mathcal{X}_f^* = \mathcal{FP}_p(\mathcal{X}_{hc}) \tag{13}$$

F. Decoding Latent Representation

As mentioned in Section IV-E, $\mathcal{X}_f^*$ is the upcoming hate intensity in the latent space. We regress this low-dimensional representation to hate intensity profiles of length n, i.e., $\mathcal{R}^*{}_{s_{(1,n)}}$. To serve this purpose, we concatenate the primal versions of the history $\mathcal{X}_h$ and the predicted future $\mathcal{X}_f^*$ of reply chains.

$$\mathcal{X}^* = \mathcal{X}_h \oplus \mathcal{X}_f^* \tag{14}$$

where $\mathcal{X}^*$ is the predicted hate intensity profile of the upcoming reply chain in the latent space.

We contrive the decoder as a mirror of the encoder. We define the decoder as,

$$\mathcal{R}^*{}_{s_{(1,n)}} = \mathcal{D}(\mathcal{X}^*) \tag{15}$$

where $\mathcal{R}^*{}_{s_{(1,n)}}$ is the final predicted hate intensity profile of the reply chain. We measure the accuracy of the prediction with respect to $\mathcal{R}^*{}_{s_{(1,n)}}$.

Geolocation	Hashtag / Keyword
United States of America	#TrumpVirus, #CreepyJoe, #MAGA, MAGA terrorist, biden not my president
United Kingdom	brexit, #BrexitShambles, tory, #RejoinEU, boris, #Tories
India	#NRC, #CAA, Sushant Singh Rajput
Other	china virus, chinese virus, covid crisis, #COVID19

TABLE II: List of hashtags and keywords used to curate the chains by location.

G. Implementation Details

We train our proposed autoencoder by backpropagating L2 loss to learn low-dimensional representations for the hate intensity profiles. Note that the decoder $\mathcal{D}(.)$ has a transpose architecture w.r.t. the encoder $\mathcal{E}_h(.)$, which is why the decoder also takes special indices as input for the decoding process. These special indices returned by the Inception module of the encoder are used for the *unmax-pooling operation*. We apply the Gaussian Mixture model for fuzzy clustering on the concatenated latent representations.

$\mathcal{PR}(.)$ is a 3-layered deep neural network unit, where the last layer employs Sigmoid activation. We consider the output of $\mathcal{PR}(.)$ to be the estimated weights $\mathcal{P}^*(C_{c1}, C_{c2}, .., C_{cj})$, as explained in detail in Section IV-D. $\mathcal{FP}_p(.)$ is another 3-layered feed forward neural network that performs linear transformation and estimates the future latent space $\mathcal{X}_f^*$.

V. Dataset Curation and Analysis

To the best of our knowledge, barring [1], there is no other dataset that contains *complete* reply chains on Twitter. Since the Twitter API does not furnish to fetch the entire reply chain for a particular tweet, we opt for an alternative method. We manually identify tweets corresponding to various real-world events; we essentially perform a hashtag-based search related to the 2020 US presidential election, the Brexit referendum in the UK and various other political issues particularly in the US, the UK and India. We extend our existing dataset [1], by focusing on the recent racial comments and controversies about the second wave spread of the COVID pandemic in various countries. The final dataset comprises $\sim$ 5000 root tweets (with their reply chains), a total of $\sim$ 1.1 million tweets and $\sim$ 950k unique users across all chains. Table II shows that our curated dataset consists of tweets from various geographical locations spanning multiple topics over which mass discussions took place during and before our collection process.

We further map the extracted reply chains to the following major topics: (i) *Donald Trump's COVID crisis*: root tweets posted as official statements from the Trump administration, public reactions of Donald Trump's handling of COVID-19 and controversial claims of Twitter users about the situation in the USA; (ii) *Joe Biden's campaign*: root tweets posted during the 2020 US presidential campaign, particularly with Joe Biden as the person of interest, consisting of controversial claims about Biden's history, rumours about his campaign's claims and some official statements from his campaign team; (iii) *Brexit referendum*: root tweets posted by the people of the UK expressing their opinions about the Brexit situation,

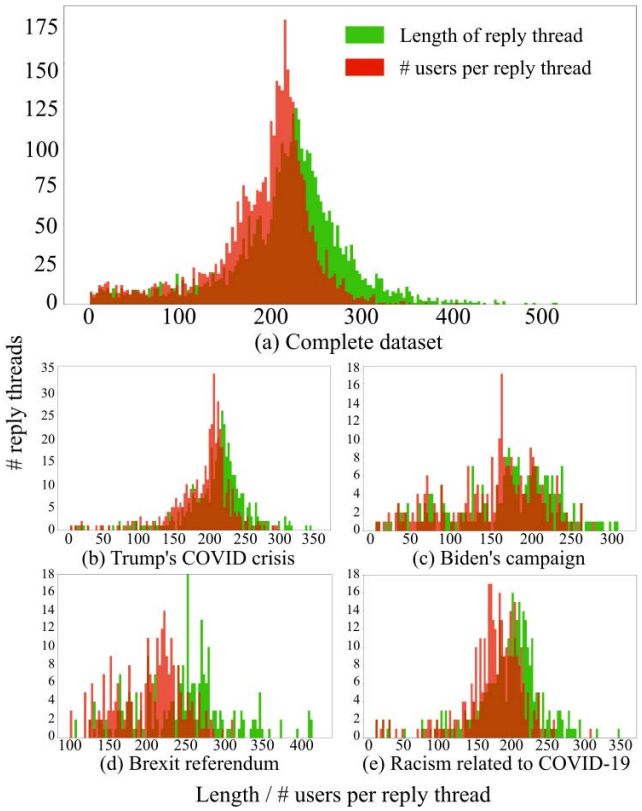

Fig. 4: Distributions of the length of the reply chains and the number of unique users per reply chain (a) in the complete dataset and (b-e) across topics.

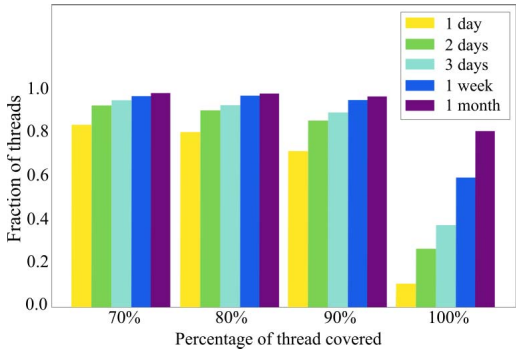

Fig. 5: Lifetime of reply chains. We show how much time a chain takes to grow upon the posting of the root tweet.

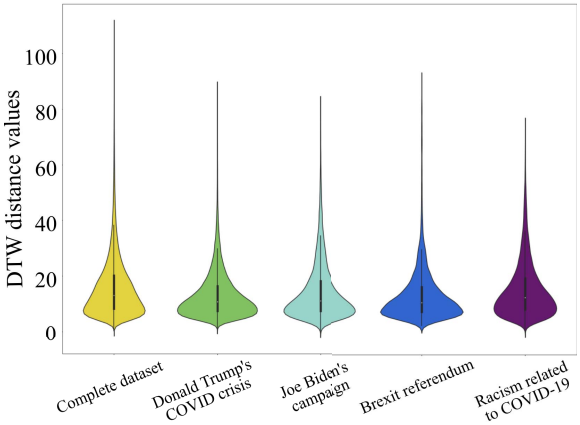

Fig. 6: Violin distribution for the pairwise DTW distances of reply chains.

some about the mistrust in Boris Johnson's administration and the Conservative (Tory) party; (iv) *COVID-19 impelled xenophobia - specifically Sinophobia*: this subset contains the Twitter mentions of COVID-19, from across the world, as the "China virus", wherein the reply chains hold the Chinese community responsible for the global pandemic.

Length of the reply chain. Figure 4 shows that the reply chains are of different length, with an average (maximum) length close to 200 (566) replies in the entire dataset. These numbers are quite similar for the four topics.

Number of unique users per reply chain. Figure 4 also illustrates that the distribution of the number of unique users in the reply chains appears to be very similar to the distribution of lengths of the reply chains for the overall dataset and across topics. This indicates that there is a high percentage of unique users in a reply chain (on an average, 88% of replies in a reply chain originate from unique users), i.e., a single user avoids taking part in the same reply chain multiple times.

Distribution of the lifetime of chains. Figure 5 highlights the correlation between the percentage of tweets per chain and the amount of time it takes to cover them. We see that 72% of the chains attract 90% of their total replies within the first 24 hours (1 day). It jumps up to 90% when we account for 72 hours (3 days). Interestingly, only 60% of the chains terminate within the first week, while more than 20% of the chains do not reach full length even after the first month of the root tweet being posted. Therefore, a large fraction of the activity in reply chains is observed within just the first 3 days.

Dissimilarity in the hate intensity profiles. As explained in Section III, we introduce the concept of "windowing" to smoothen the hate intensity profile by introducing a rolling average on the hate intensity profile of the reply chain. Figure 6 illustrates the distribution of the pairwise Dynamic Time Warping (DTW)[3] distances over the dataset. We observe that the pairwise DTW distance distribution has a huge range and a high variance; the median value lies close to 5 which is quite high given that the range of values in the hate intensity profiles is $[0-1]$. This further indicates that there is no coherent pattern across hate intensity profiles, neither in the complete dataset nor within the topic-specific datasets, indicating the non-triviality of the problem under study.

VI. Baseline Methods

To our knowledge, there is no prior work on forecasting hate intensity of reply chains as it evolves given initial few replies of a tweet on Twitter. We, therefore, adopt both classical and deep learning based time-series forecasting and temporal pattern modeling approaches.

[3]See [32] for the formulation of DTW.

Model	r	RMSE ↓	MFE ↓
LSTM	0.145	0.611	0.500
CNN	0.105	0.644	0.509
DeepAR	0.310	0.484	0.065
TFT	*0.469*	*0.437*	*0.076*
N-Beats	0.380	0.544	0.085
ForGAN	0.240	0.603	0.360
DRAGNET w/o Sentiment	0.515	0.286	0.018
DRAGNET	**0.563**	**0.247**	**0.010**

TABLE III: Overall performance (↓: lower value is better). Second last row indicates the performace of DRAGNET without the sentiment feature. The best baseline is italicized.

- **LSTM:** We use stacked LSTM with hyper-parameters setup defined in a recent study [20]. For hate intensity prediction, we incorporate a dense layer with ReLU activation.
- **CNN:** We use 1-D CNN architecture followed by a fully-connected layer with ReLU activation. For the convolutional layer we use 64 filters and a kernel size of 2 [39].
- **N-Beats:** It is a deep learning based model used for univariate time-series forecasting. It relies on forward and backward residual links. [25].
- **DeepAR:** It is an auto-regressive recurrent network primarily used for time-series forecasting [26].
- **TFT:** It is deep neural network architecture for multi-horizon forecasting using self-attention [40].
- **ForGAN:** It is a conditional GAN based model designed to learn data distribution with modules for feature selection. It is used for probabilistic time-series forecasting. [28].

We do not compare DRAGNET with DESSERT [1] because DESSERT makes predictions at any time point on the fly, whereas DRAGNET predicts for the entire reply chain by accounting for only a few initial replies in the reply chain.

VII. Experimental Setup

We consider the following hyper-parameters as default: w = 0.6, δ = 10, t_h = 35, t_f = 284, n = 300, j = 15, N_{X_h} = 32, N_{X_f} = 128 and DRAGNET with Inception-Time module [34] in the autoencoder for transformation step with kernel sizes $[5, 7, 9]$. Davidson's model [12] is considered as the default hate speech detection model (see Section III). A Gaussian Mixture model with covariance type **full** is used for fuzzy clustering. We use 80-20 split for training and testing. In Section VIII, we will show how the model responds to the change in the major hyper-parameters.

For the purpose of evaluation, we use three metrics - **Pearson Correlation coefficient** (r), **Root Mean Square Error** (RMSE) and **Mean Forecast Error** (MFE)[4]. For the former metric, higher value is better, whereas for the latter two, lower value is better.

VIII. Experimental Results and Analysis

In this section, we perform a detailed comparative analysis of the performance of the competing models. We also show how DRAGNET and the best baseline respond with respect to changes in the – hate detection model used to measure the hate intensity, demography and topic of the root tweet, types of users, etc. In addition to this, we also show how DRAGNET reacts to changes in the model parameters.

A. Comparative Analysis

Table III shows the overall performance. In general, DRAGNET outperforms all the baselines by a significant margin across all three comparison metrics. TFT ends up as the best performing baseline. However, DRAGNET beats TFT by 0.094 points in r, 0.190 points in RMSE and 0.066 points in MFE. Among other baselines, LSTM and CNN turn out to be the worst baselines, while DeepAR and N-Beats have performances comparable to TFT. Surprisingly, ForGAN also seems to be one of the worst performers. This may be due to the extreme dissimilarity in the hate intensity profiles present in our dataset. We also show that DRAGNET abetted with the sentiment feature is more effective than the other baselines. However, with sentiment as an additional feature, the performance of DRAGNET improves further.

B. Detailed Introspection

We further dig deeper into the performance of DRAGNET to better understand its superiority and limitations. Throughout this study, we compare DRAGNET with TFT, the best baseline.

- **Hate speech classifier.** As explained in Section III, we use a hate speech detection model to measure the hate intensity score. We are curious to know how the model performs if we change the hate speech classifier. To this end, we consider the following hate speech detection models – Davidson et al. [12] (default), Founta et al. [13] and Waseem et al. [11]. Figure 7(a) shows a consistent precedence of DRAGNET over TFT across all three hate detection models.
- **Weight w in the hate intensity score.** In Section III, our hate intensity score uses w to facilitate the trade-off between hate detection model and the hate lexicon component. As shown in Figure 7(b), DRAGNET clearly outperforms TFT for all three values of w (i.e., 0.45, 0.6, 0.75).
- **Demography of root tweet.** Our curated dataset consists of reply chains with root tweets originating in various countries – the US, the UK and India & others (Brazil, Australia, Argentina). DRAGNET shows consistency across all these geographical locations as illustrated in Figure 7(c).
- **Topic-wise data.** Figure 7(d) illustrates the performance of both DRAGNET and TFT across four topics (detailed in Section V). DRAGNET beats TFT across all topics.
- **Type of root tweet.** We randomly select a set of 1830 root tweets and manually label them as 'controversial', 'fake', 'hate' and 'others'. Figure 7(e) again illustrates that DRAGNET shows consistency across these labels.
- **Type of root users.** We also analyse whether the popularity of the user who posted the root tweet affects the performance of the models. To check this, we consider the follower count of a user as a proxy to popularity. We then divide the root tweets equally into various bins based on the follower count of root users and measure the performance across bins as shown in Figure 7(f) – Bin 1 (Bin 4) represents the set of the least (most) popular root users. DRAGNET once

[4]Given two sequences a and b of length n, $MFE(a,b) = \frac{\sum_{i=0}^{n}(a[i]-b[i])}{n}$.

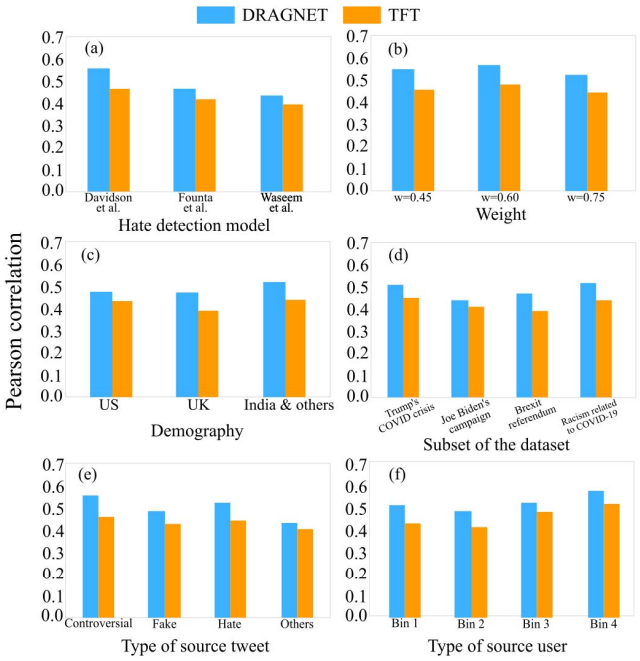

Fig. 7: Comparison in performance of DRAGNET and TFT using the Pearson correlation w.r.t (a) hate speech detection model used, (b) weight w in the hate intensity function, (c) demography of the root tweet, (d) topics, (e) type of root tweet, and (f) type of source user.

again remains consistent across bins, indicating its stable performance irrespective of the popularity of the root users.

C. Ablation Study

Furthermore, we study how various model parameters affect the performance of DRAGNET and TFT.

- **Change in δ.** The hyper-parameter δ represents the window size in the hate intensity profiles. Figure 8(a) shows that both DRAGNET and TFT improve in performance with an increase in the value of δ.
- **Change in t_h.** t_h represents the size of the initial history that it requires to predict the complete hate intensity profile for a new reply chain. As expected, Figure 8(b) illustrates that the prediction accuracy of DRAGNET and TFT increases as t_h increases.
- **Varying the number of clusters.** One of the crucial hyper-parameters of DRAGNET is the number of clusters, j in the fuzzy clustering step. Figure 8(c) illustrates that DRAGNET's performance is the best for $j = 15$ (which is the default value).
- **Varying the clustering algorithm.** We use Gaussian Mixture (GM) model as the default clustering method. We further check the performance of DRAGNET with other clustering methods – C-Means (a hard clustering method) and Bayesian Gaussian Mixture (BGM, a variant GM). Figure 8(d) confirms that GM performs the best out of the three, with BGM showing comparable results, This also supports our decision of choosing a fuzzy clustering approach.

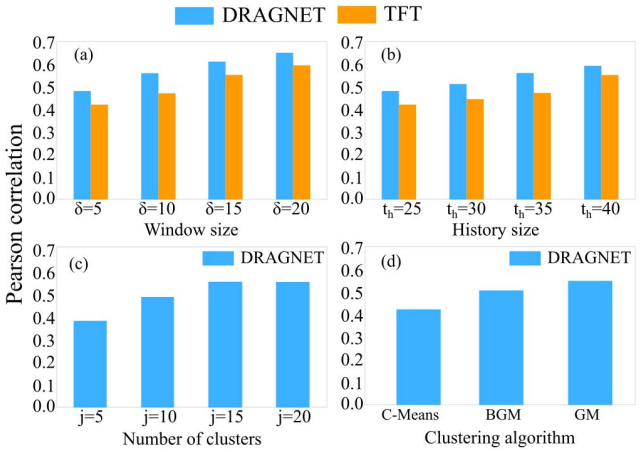

Fig. 8: Performance of DRAGNET and TFT w.r.t the change in (a) window size, and (b) history. We also show the performance of DRAGNET with the change in (c) the number of clusters, and (d) the clustering method.

IX. Real-world Deployment

DRAGNET was deployed by a fake news technology scale-up in its advanced AI platform to monitor real-world harmful content ingested typically from heterogeneous social content streams. It was tested by integrating it into a proprietary pipeline that ranks and scores individual content pieces as well as aggregated semantic content clusters (narratives) for insights about misinformation and deleterious content. In particular, DRAGNET was used as part of a hybrid workflow implemented by the company to harness the expertise of in house fact checkers, content moderators and OSINT analysts along with advanced AI models to track and monitor disinformation and problematic harmful narratives. The aim here was to use DRAGNET to model the hate profile of narratives and also to link different narratives with similar profiles. The workflow of the application of DRAGNET in the AI Platform is illustrated in Figure 9.

We observed that DRAGNET offers unique insights for the OSINT analysts to pay close attention to the characteristics of harmful narratives and formulate actions and countermeasures to proactively detect them when they are in pre-viral stages. DRAGNET has significant potential to improve the overall accuracy of the anti-harm pipelines in profiling and ranking content for hate speech and real-world online harmfulness. In particular, the hate intensity scores from DRAGNET offer additional knowledge to the proprietary pipelines for ranking and prioritization of high-risk online malevolence for enforcement of countermeasures to minimize their impact and damage.

X. Conclusion

In this paper, we studied a novel problem - hate intensity prediction of Twitter reply chains. We started off with curating

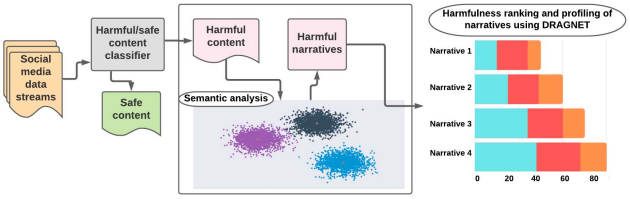

Fig. 9: Visual workflow of the application of DRAGNET in an advanced AI Platform for hate intensity profiling and ranking of narratives (aggregated semantic content analysis).

a large-scale dataset of $\sim 5k$ complete reply chains related to four controversial topics from Twitter. We then proposed DRAGNET, a novel deep stratified learning model to address the problem. DRAGNET is highly efficient, outperforming six baselines. Further, DRAGNET has been deployed in an advanced AI platform for monitoring detrimental content on the web to profile and rank content clusters for high risk threats, thereby to recommend countermeasures to minimise their reach and reduce damage.

REFERENCES

[1] S. Dahiya, S. Sharma, D. Sahnan, V. Goel, E. Chouzenoux, V. Elvira, A. Majumdar, A. Bandhakavi, and T. Chakraborty, "Would your tweet invoke hate on the fly? forecasting hate intensity of reply threads on twitter," in *ACM SIGKDD*, 2021, pp. 2732–2742.

[2] S. R. Manoj Kumar Pathak1, "Mob lynching: A new form of hate crime," *Medico Legal Update*, vol. 20, no. 3, pp. 122–128, Jul. 2020.

[3] F. Poletto, V. Basile, M. Sanguinetti, C. Bosco, and V. Patti, "Resources and benchmark corpora for hate speech detection: a systematic review," *Lang Resources & Evaluation (2020)*, 2020.

[4] B. Mathew, R. Dutt, P. Goyal, and A. Mukherjee, "Spread of hate speech in online social media," in *WebSci*, 2019, p. 173–182.

[5] S. Masud, S. Dutta, S. Makkar, C. Jain, V. Goyal, A. Das, and T. Chakraborty, "Hate is the new infodemic: A topic-aware modeling of hate speech diffusion on twitter," *CoRR*, vol. abs/2010.04377, 2020.

[6] M. Ribeiro, P. Calais, Y. dos Santos, V. Almeida, and W. Meira Jr, ""like sheep among wolves": Characterizing hateful users on twitter," in *MIS2 Workshop at WSDM'2018*, 12 2017.

[7] P. Liu, J. Guberman, L. Hemphill, and A. Culotta, "Forecasting the presence and intensity of hostility on instagram using linguistic and social features," in *ICWSM*, vol. 12, no. 1, 2018.

[8] P. Patwa, M. Bhardwaj, V. Guptha, G. Kumari, S. Sharma, S. PYKL, A. Das, A. Ekbal, M. S. Akhtar, and T. Chakraborty, "Overview of constraint 2021 shared tasks: Detecting english covid-19 fake news and hindi hostile posts," in *CONSTRAINT, AAAI*, 2021, pp. 42–53.

[9] A. Ghosh Chowdhury, A. Didolkar, R. Sawhney, and R. R. Shah, "ARHNet - leveraging community interaction for detection of religious hate speech in Arabic," in *ACL Student Research Workshop*, Florence, Italy, Jul. 2019, pp. 273–280.

[10] S. T. Luu, K. Van Nguyen, and N. L.-T. Nguyen, "A large-scale dataset for hate speech detection on vietnamese social media texts," *arXiv preprint arXiv:2103.11528*, 2021.

[11] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in *NAACL student research workshop*, 2016, pp. 88–93.

[12] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *ICWSM*, vol. 11, no. 1, 2017.

[13] A. M. Founta, D. Chatzakou, N. Kourtellis, J. Blackburn, A. Vakali, and I. Leontiadis, "A unified deep learning architecture for abuse detection," in *WebSci*, 2019, p. 105–114.

[14] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine, "The hateful memes challenge: Detecting hate speech in multimodal memes," in *NeurIPS*, Vancouver, Canada, 2020, pp. 1–14.

[15] J. Qian, H. Wang, M. ElSherief, and X. Yan, "Lifelong learning of hate speech classification on social media," in *NAACL*, 2021, pp. 2304–2314.

[16] P. Fortuna and S. Nunes, "A survey on automatic detection of hate speech in text," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–30, 2018.

[17] S. Zannettou, B. Bradlyn, E. De Cristofaro, H. Kwak, M. Sirivianos, G. Stringini, and J. Blackburn, "What is gab: A bastion of free speech or an alt-right echo chamber," in *WWW*, 2018, p. 1007–1014.

[18] I. Rojas, O. Valenzuela, F. Rojas, A. Guillén, L. J. Herrera, H. Pomares, L. Marquez, and M. Pasadas, "Soft-computing techniques and arma model for time series prediction," *Neurocomputing*, vol. 71, no. 4-6, pp. 519–537, 2008.

[19] X. Wei, L. Zhang, H.-Q. Yang, L. Zhang, and Y.-P. Yao, "Machine learning for pore-water pressure time-series prediction: application of recurrent neural networks," *Geoscience Frontiers*, vol. 12, no. 1, pp. 453–467, 2021.

[20] S. Elsworth and S. Güttel, "Time series forecasting using lstm networks: A symbolic approach," *arXiv preprint arXiv:2003.05672*, 2020.

[21] Z. Mariet and V. Kuznetsov, "Foundations of sequence-to-sequence modeling for time series," in *AISTATS*, 2019, pp. 408–417.

[22] C. Fan, Y. Zhang, Y. Pan, X. Li, C. Zhang, R. Yuan, D. Wu, W. Wang, J. Pei, and H. Huang, "Multi-horizon time series forecasting with temporal attention learning," in *ACM SIGKDD*, 2019, pp. 2527–2535.

[23] R. Sen, H.-F. Yu, and I. Dhillon, "Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting," *arXiv preprint arXiv:1905.03806*, 2019.

[24] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *arXiv preprint arXiv:1907.00235*, 2019.

[25] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," *arXiv preprint arXiv:1905.10437*, 2019.

[26] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[27] Y. Yuan and K. Kitani, "Diverse trajectory forecasting with determinantal point processes," *arXiv preprint arXiv:1907.04967*, 2019.

[28] A. Koochali, P. Schichtel, A. Dengel, and S. Ahmed, "Probabilistic forecasting of sensory data with generative adversarial networks–forgan," *IEEE Access*, vol. 7, pp. 63 868–63 880, 2019.

[29] M. Wiegand, J. Ruppenhofer, A. Schmidt, and C. Greenberg, "Inducing a lexicon of abusive words – a feature-based approach," in *NAACL*, 2018, pp. 1046–1056.

[30] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.

[31] P. Hastings, S. Hughes, D. Blaum, P. Wallace, and M. A. Britt, "Stratified learning for reducing training set size," in *International Conference on Intelligent Tutoring Systems*. Springer, 2016, pp. 341–346.

[32] V. Niennattrakul and C. A. Ratanamahatana, "On clustering multimedia time series data using k-means and dynamic time warping," in *MUE*, 2007, pp. 733–738.

[33] J. Sun, Y. Li, H.-S. Fang, and C. Lu, "Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis," *arXiv preprint arXiv:2103.07854*, 2021.

[34] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, "Inceptiontime: Finding alexnet for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1936–1962, 2020.

[35] C. Yang, J. Qiao, H. Han, and L. Wang, "Design of polynomial echo state networks for time series prediction," *Neurocomputing*, vol. 290, pp. 148–160, 2018.

[36] Z. Cui, W. Chen, and Y. Chen, "Multi-scale convolutional neural networks for time series classification," *arXiv preprint arXiv:1603.06995*, 2016.

[37] A. Ziat, E. Delasalles, L. Denoyer, and P. Gallinari, "Spatio-temporal neural networks for space-time series forecasting and relations discovery," in *ICDE*. IEEE, 2017, pp. 705–714.

[38] D. Reynolds, *Gaussian Mixture Models*. Boston, MA: Springer US, 2009, pp. 659–663.

[39] S. Mehtab, J. Sen, and S. Dasgupta, "Robust analysis of stock price time series using cnn and lstm-based deep learning models," in *ICECA*, 2020, pp. 1481–1486.

[40] B. Lim, S. Ö. Arik, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *CoRR*, vol. abs/1912.09363, 2019.