ABSTRACT A time-series of numerical data and a sequence of time-ordered documents are often correlated. This paper aims at modeling the impact that the underlying themes discussed in the text data have on the time series. To do so, we introduce an original topic model, Time Series Impact Through Topic Modeling (TSITM), that includes contextual data by coupling Latent Dirichlet Allocation (LDA) with linear regression, using an elastic net prior to set to zero the impact of uncorrelated topics. The resulting topics act as explanatory variables for the regression of the numerical time series, which allows us to understand the time series movements based on the events described on the text data. We have tested our model on two datasets: first, we used political news to explain the US president’s disapproval ratings; then, we considered a corpus of economic news to explain the financial returns of 4 different multinational corporations. Our experiments show that an appropriate selection of hyperparameters (via repeated random subsampling validation and Bayesian optimization) leads to significant correlations: both an intrinsic baseline and state of the art methods were significantly outperformed by TSITM in MSE, MAE and out-of-sample $R^2$, according to our hypothesis tests. We believe that this framework can be useful in the context of reputational risk management.

INDEX TERMS Expectation-maximization algorithms, natural language processing, regression analysis, text mining, text analysis, time series analysis.

I. INTRODUCTION

In many circumstances, a time-series of numerical data and a sequence of time-ordered texts may exhibit noticeable correlations. One of the most paradigmatic examples is the field of natural language based financial forecasting (NLFF), which makes use of NLP techniques to predict financial markets from news and social media [1]. Another example is the use of social media comments about a certain product to predict the number of sales of that product as a function of time [2]. Other authors have studied correlations between macroeconomic variables, such as inflation, and news [3]. In recent years, there has been a growing interest in quantifying elusive concepts such as reputational risk [4], that may be seen as the potential economic impact on a company or entity caused by reputational news that appear in press and social media. The appearance of investor based social networks such as StockTwits\(^1\) and the Dongfang Wealth Network\(^2\) have also spurred interest in making investment recommendations from investor’s comments about stock trends [5], [6]. To us, it is clear that the joint analysis of text and numerical time series data is a growing field of research of considerable practical importance.

Typically, this analysis has been approached from the domain of text regression [7]. In its simplest form, text regression may take as input a TF or TF-IDF representation of the documents to predict a time series [8]. In this framework, a weight will be obtained for each word of the vocabulary, thus determining the contribution of each document to the forecasting of the dependent variable. However, we identify two main problems in this approach: first, there is a high risk of overfitting, given that there are as many regressors as words in the vocabulary; second, the interpretability of results
is poor. Both problems would be solved if topics [9], rather
than words, were used as explanatory variables. Thanks to
the dimensionality reduction, the number of features needed
to represent a document is much smaller than the number
of words, so we should expect a better behavior in terms
of overfitting. Furthermore, topics contain a higher semantic
meaning than words, which yields a better interpretability of
results.

Topic modeling has become a standard technique for
uncovering the hidden structure of texts (see [10] for a recent
review). A topic is defined as a probability distribution over
words in a vocabulary. Consequently, a document made of
a large number of words from that vocabulary can be modeled
as a probability distribution over a reduced number of
topics, each of them representing an underlying semantic
theme. Following the pioneering work of Probabilistic Latent
Semantic Analysis (PLSA) [11], Blei et al. proposed the now
predominant model, Latent Dirichlet Allocation (LDA) [9].
As we said before, one of the main advantages of topic mod-
eling is that it allows for the decomposition of a very high-
dimensional and sparse matrix (D documents times V words)
into a smaller number of features (a matrix of D documents
times K topics and another of K topics times V words).

In this work, we introduce an original topic model, Time
Series Impact Through Topic Modeling (TSITM), that com-
bines LDA with linear regression in order to determine the
impact that a corpus of news has on a particular time series.
In our model, a regression coefficient is determined for
each topic that show a significant correlation with the time
series. As we will later demonstrate, in TSITM there exists a
coupling between LDA and linear regression, and therefore
the extraction of topics will be customized to fit each time
series specifically. This coupling implies that TSITM is not a
straightforward combination of existing methods, but a new
model that requires its own optimization algorithm.

Notice that in this work we will focus on fitting an external
time series. While autoregressive models are certainly possi-
bile for capturing linear interdependencies among the different
topics and the influence between each other [12], we will
exclude that scenario in our research.

The structure of this paper is as follows: in section II, we
describe previous works in the field of text regression that
used topics as regressors for a time series; in section III,
we specify the research goals and contributions of our work;
in section IV, we describe our model and propose an algo-
rithm for parameter estimation; in section V, the datasets used
for the experimental validation are described; the experi-
mental set-up is discussed in section VI and results are presented
in section VII; the implications and limitations of our work
are discussed in section VIII. We conclude with some final
remarks in section IX.

II. LITERATURE REVIEW

Different approaches have already been tried to correlate
topics with a time series. We will briefly review some of the
most relevant precedents in this field.

A. SEPARATE MODELS FOR TOPICS AND TIME SERIES

1) ITERATIVE TOPIC MODELING WITH TIME SERIES
FEEDBACK (ITMTF)

One of the first attempts to correlate topics with time series
that appeared in the scientific literature was the Iterative Topic
Modeling with Time Series Feedback (ITMTF) model [13],
an iterative framework for discovering causal topics. The
strategy is to progressively increase the correlation of topics
with the time series data through the introduction of prior
distributions in a feedback mechanism. The appropriate prior
distributions are obtained as follows: first, a collection of
topics is extracted from a corpus of texts using any stan-
dard topic model; then a causality measure is computed to
determine correlations between the topics and the time series,
and a selection of candidate causal topics is obtained; for
each of these candidate topics, the most significant causal
words are obtained by applying the causality measure at
a word level; finally, the prior distribution is defined by
using the previously identified significant words and their
impact values, separating positive impact terms and negative
impact terms, and assigning prior proportions according to
the significance levels. By applying the topic model on the
collection of documents again, but using the new priors,
the new topics will be more correlated with the time series.
This process is repeated until a stopping criterion is reached.
Although this framework is successful at introducing external
information into the topics, it must be noted that it does not
make use of a unified model for text and numerical data.
This implies that there will be two different probabilities that
are not simultaneously optimized, so the efforts to create
semantically coherent topics might not be the right direction
to increase the correlation with the time series. We also
note that this framework does not provide a mechanism to
forecast the time-series values given the corresponding texts
either.

B. UNIFIED MODELS FOR TOPICS AND TIME SERIES

1) SUPERVISED LDA (sLDA)

The standard example of a unified model that jointly gen-
erates topics and a discrete or continuous response variable
is supervised Latent Dirichlet Allocation (sLDA) [14]. Its
generative process mirrors that of LDA, but it also pairs each
document of the corpus with a response variable modeled by a
normal distribution whose mean is a function of the empirical
frequencies of the topics in that document. However, sLDA
is mainly used in document classification tasks and it was not
originally intended to be applied for time series prediction.
Recent attempts to employ sLDA for this purpose showed a
severe overfitting of the time series by distorting the topic
parameters [15]. Additionally, notice that in sLDA each docu-
ment will be used to predict a value of the time series, instead
of aggregating all the documents published on a given time
period to make a single prediction. This problem could be
addressed by repeating the value of each numerical data as
many times as the total number of documents published on

J. Cendrero et al.: Time Series Impact Through Topic Modeling

VOLUME 10, 2022

97328
that time period, but then the model would not be sensible to the absolute number of news. Furthermore, this would amplify the statistical value of a given numerical data point with respect to the others depending on the total number of associated documents, which does not seem appropriate for our goals.

2) TOPIC FACTOR MODEL (TFM)
In 2015, two groups concurrently claimed to have developed for the first time a unified model for both topic modeling and time series prediction: Joe Staines and David Barber proposed the Topic Factor Model (TFM) [16] and Sungrae Park, Wonsung Lee and Il-Chul Moon proposed the Associative Topic Model (ATM) [17].

TFM [16] is an original topic model which resembles sLDA. The goal here is to predict a time series from a single document. TFM achieves this by associating a latent time series to each topic, \( q_{dt} \), and defining the document time series, \( r_{dt} \), as a linear combination of these latent series (plus a document specific time series \( \epsilon_{dt} \) to ensure that it is always possible to fit the data). The key point that unifies topic modeling and time series prediction in this framework is the fact that the document topic matrix, \( \theta_{dk} \), acts both as the parameters of a multinomial probability distribution over each topic \( k \) for each document \( d \) (as in LDA) and, at the same time, as the coefficients for the factor model used to construct \( r_{dt} \) from \( q_{dt} \). Within this model, a single document is expected to predict an entire time series of numerical data, and this qualitatively differs from the applications we have in mind in our work, mainly because we would expect that the impact of a document published at a time \( t \) on the value of the time series at a sufficiently distant timestamp \( t' \) will be negligible. Furthermore, this model does not allow us to see how the publication of new documents, with the corresponding change in the presence of each topic over time, dynamically impacts the evolution of the numerical time series.

3) ASSOCIATIVE TOPIC MODEL (ATM)
The ATM model [17], on the other hand, assumes that documents published at a given time \( t \) and the numerical value of the time series at that same \( t \) are generated from a common latent random variable, which indicates the popularity of each topic \( k \) at that time. Taking the Dynamic Topic Model (DTM) [18] as a starting point, they introduce the novelty of considering that the parameters of the prior distribution for \( \theta_{tk} \) at time \( t \), \( \alpha_{tk} \), will also be used to sample a numerical value, \( y_t \). This way, the same prior information that is used to obtain the topic proportions of documents at a given time is also used to predict the value of the numerical time series at that time. Intuitively, this is equivalent to considering that the same series of underlying events is manifesting itself through two types of data: numerical and textual. In order to generate the numerical values \( y_t \), one has to introduce a set of K-dimensional linear coefficient parameters \( b_k \) and take the scalar product with \( \alpha_{tk} \) (which has been previously normalized using a softmax function). The \( b_k \) coefficients, which can be positive or negative, would quantify the impact of each topic on the time series. This model can forecast future values of the series if the matching texts are provided. There is no regularization on the \( b_k \) parameters, hence they are prone to overfit when the number of topics expected is sufficiently large (or this could lead to model complexity issues). The number of topics is set by human choice, and that determines how many correlations will be discovered by the model. Also, the coupling of \( b_k \) with the softmax of the \( \alpha_{tk} \) introduces multi-collinearity.

C. DOMAIN SPECIFIC MODELS

1) FINANCIAL LDA
Recently, domain-specific models have also been proposed. This is the case for Financial Latent Dirichlet Allocation (FinLDA) [19], a variant of LDA specifically designed for fitting financial time series. In this framework, the publication of a document \( d \) generates a movement of prices in the financial time series, \( f_d \). The finite time-lag used to determine a change in prices can range from minutes to days, depending on the nature of the dataset. To model the change induced by each document, the authors introduce a hidden distribution of changes per topic, \( \delta_k \), so that the probability to observe \( f_d \) can be determined by calculating the weighted average of the probabilities of movement of each topic by the number of word tokens coming from each topic \( k \) in document \( d \), \( \left( \frac{\sum_{n}^{N_d} \delta_{tk}}{N_d} \right) \). Two versions of the model are proposed, one for discrete changes (Discrete FinLDA, or D-FinLDA) and the other for continuous changes (Continuous FinLDA, or C-FinLDA). Once the parameter inference is finished, the outputs obtained from FinLDA then serve as input features for other machine learning algorithms that fit the financial time series (such as Support Vector Regression Machines or Multi-Layer Perceptron). As compared to the aim of our work, the main shortcoming of this model is that only one document is allowed to be published for each time-lag. We have in mind situations where several documents, often dealing with very different topics, are published at the same time-lag. Another issue is the lack of a lasso-type regularization in this model, which implies that topics that show no correlation with the time series will be used to construct the FinLDA features. It is also important to notice that standardization of the inputs is not performed in this model, which poses a problem under different scaling of the data.

2) LDA FOR MACROECONOMIC VARIABLES
In [20] the output of a topic model (LDA) trained in a Norwegian financial news dataset is used as predictor variables in a regression (Latent Threshold Model, to enforce sparsity), in order to predict several macroeconomic variables, including asset prices as summarized in the Oslo Exchange index OSEBX. There is no coupling between the time series and the topics, in the sense that topics are trained alone. Surprisingly,
the daily topic distribution is equally normalized (to one) every day, so that the presence of a topic in a given day depends on the overall distribution of other topics that same day.

III. OBJECTIVES AND CONTRIBUTIONS

In this article, we propose a joint model that takes as inputs a corpus of documents and a time-series, and outputs a collection of topics that are designed to act as regressors for the given time series together with the values of their regression coefficients. We pretend to accomplish three research goals with this model:

(RG1) Automatically select only the topics of the corpus that show significant correlation with the time series and discard the ones that are uncorrelated. If this is achieved via regularization of the regression coefficients, then the penalization should be applied equally to all topic presence series, irrespective of their relative scale.

(RG2) Quantify the impact (which can be positive or negative) of each selected topic on the observation of unseen documents.

(RG3) Predict the value of the time-series given the observation of unseen documents.

None of the related works described in section II satisfactorily addresses all the three research goals proposed above at the same time. In particular, RG1 is not satisfied by most of the models, since the number of correlations is typically chosen arbitrarily as an input parameter (this is the case for [16], [17], [20], to name a few). Additionally, as we will later see, in TSITM we are going to introduce a rescaling of the topic presence, a crucial feature in regularized linear regression which is not present in any of the models previously cited. RG2 is more commonly satisfied, although some models lack a topic dependent response (this is the case for [13]). RG3 is not a feature present in many of these models, whether because a single document is used to predict several values of the time-series at the same time [16], because the publication of many documents at the same time-lag is not considered [14], [19] or because there is no forecasting mechanism at all [13]).

The precise meaning of the third goal requires some clarification: we aim at modeling the relationship between the time series and the topics during a specified period of time in order to explain the moves of the series, but we do not try to forecast its future values outside that period, since the signals discovered by the model may have vanished. Further discussion of our evaluation strategy regarding this point will be found in section VI.

The contributions of this paper can be summarized as follows:

(C1) We introduce a joint topic model, TSITM, that satisfactorily addresses research goals RG1 - RG3.

(C2) We derive an efficient method for obtaining a maximum likelihood estimate of the model parameters by making use of the ECM algorithm.

(C3) We test TSITM on two different scenarios of practical importance: explaining US president’s disapproval ratings from political news and stock returns of multinational corporations from economic news.

IV. TIME SERIES IMPACT THROUGH TOPIC MODELING (TSITM)

In this section, we describe our proposed model, TSITM. In subsection IV-A, we will briefly review the LDA model and the elastic net regularization, which are the two building blocks that will be later used to construct our model, in order to fix notation and terminology. In subsection IV-B, we will introduce the complete log-likelihood for our model and describe its core features. Finally, in subsection IV-C we will propose an optimization strategy for the log-likelihood using the ECM algorithm, paying special attention to the use of proximal operators to find numerical solutions when no analytical methods are available.

A. NOTATION AND TERMINOLOGY

For most of the paper, we will follow the standard notation used in topic modeling, as it appears in [9] and [10]: a word $w$ is an item from a vocabulary of size $V$ represented by a one-hot encoding vector and a document is a sequence of $N$ words denoted by $w = \{w_1, w_2, \ldots, w_N\}$, where $w_n$ is the $n$th word in the sequence. We define a corpus as a collection of $D$ documents. Let us denote $n_{dw}$ the number of appearances of the word $w$ in the document $d$. The first building block that we will employ in the construction of our model is Latent Dirichlet Allocation [9], which aims at optimizing the following log-likelihood:

$$
\mathcal{L}_{\text{LDA}} = \sum_{d=1}^{D} \sum_{w=1}^{V} n_{dw} \log \left( \sum_{k=1}^{K} \theta_{dk} \beta_{kw} \right)
+ \sum_{d=1}^{D} \sum_{k=1}^{K} \alpha_{k} \log \theta_{dk} + \sum_{w=1}^{V} \sum_{k=1}^{K} \eta_{w} \log \beta_{kw},
$$

(1)

where $k$ indexes each of the $K$ topics; $\beta_{kw}$ are the parameters of a multinomial distribution over the vocabulary for each topic (i.e. the probability that topic $k$ contains word $w$); $\theta_{dk}$ are the parameters of a multinomial distribution over the topics for each document (i.e. the probability that document $d$ covers topic $k$); and $\alpha_{k}$ and $\eta_{w}$ are the hyperparameters of the Dirichlet distributions used as priors for the topic distributions (i.e. to “smooth” the multinomial parameters). When optimizing this log-likelihood via the EM method, a latent variable $z_{dhw}$ is introduced to indicate which topic generated each word of a document.

The second building block is regularized linear regression. We will consider a time-series of target values $y_t$, with $t = 1, \ldots, T$, and $J$ series of regressors $X_{j,t}$. We will make use of linear regression with elastic net regularization [21], which linearly combines the $L_1$ and $L_2$ penalties of the Lasso and Ridge methods respectively, to fit the data. The log-likelihood
for this problem is:

\[
\mathcal{L}_{\text{ENET}} = -\frac{1}{2T} \sum_{t=1}^{T} \left( y_t - \mu_0 - \sum_{j=1}^{J} X_{ij} \mu_j \right)^2 - \lambda_{\mu} \left\{ \frac{1 - \rho}{2} \sum_{j=1}^{J} \mu_j^2 + \rho \sum_{j=1}^{J} |\mu_j| \right\},
\]

where \( \mu_0, \ldots, j \) are the regression coefficients; and \( \lambda_{\mu} \) and \( \rho \) are the regularization weights (\( \lambda_{\mu} \) determines how strong the regularization is, while \( \rho \) weights the relative importance of the Ridge and Lasso terms). The \( L_2 \) constraint results in a shrinkage of the coefficients, while the \( L_1 \) term can set to zero some of them, thus mitigating multi-collinearity and model complexity. This type of regularization is standard in situations where there is a large number of features for a very small set of observations. From a Bayesian point of view, (2) can be seen as the product of a normal distribution with mean \( \mu_0 + \sum_{j=1}^{J} X_{ij} \mu_j \) and an exponential prior distribution for the \( \mu_j \) coefficients of the type

\[
p(\mu_j) \propto \exp\left[ -\lambda_{\mu} \left( \rho |\mu_j|^2 + (1 - \rho) |\mu_j| \right) \right].
\]

where, in order to arrive at (2), we take the variance of the normal distribution, \( \sigma^2 \), to be a known constant with value \( \sigma^2 = T \).

### B. DESCRIPTION OF THE MODEL

The key idea behind the TSITM model is to couple the two building blocks that we have just presented in subsection IV-A: our goal is to fit the target variable \( y_t \) by making use of the topic breakdown of each document, \( \theta_{dk} \), as regressors. In order to do this, we first have to find a way to convert the time-independent parameters \( \theta_{dk} \) into a time series. Assuming that each document \( d \) has a timestamp that indicates its publication date, \( t_d \), we can define the time series of a topic \( k \) as

\[
\theta_{dk} = \sum_{d=1}^{D} \delta_{t_d, t} \theta_{dk},
\]

where \( \delta_{t_d, t} \) is a Kronecker delta that selects only the documents that have been published at a time \( t \). Therefore, \( \theta_{dk} \) can be seen as the presence or popularity of each topic over time. Notice that using \( \theta_{dk} \) and \( \theta_{k} \) to denote two different concepts is an abuse of notation; however, no conflict will occur because time indexes will be labeled with \{ \( t, t', \ldots \) \} and document indexes with \{ \( d, d', \ldots \) \}.

#### 1) GENERATIVE PROCESS

Now that we have constructed a set of time series from the topic parameters, we can use them as the regressors to predict \( y_t \). The resulting regression coefficients will indicate the impact of each topic on the target variable. We start from the generative process of LDA, depicted in Fig. (1a), to which we add the sampling of a numeric value, \( y_t \), from a normal distribution that depends on the topic presence parameters \( \theta_{dk} \) at each timestamp \( t \) and the regression coefficients \( \mu_k \), which in turn are derived from an elastic net prior. The result is depicted in Fig. (1b) and can be explicitly described with the following generative process:

1) For each topic \( k \in \{1, \ldots, K\} \):
   a) Generate \( \beta_{kw} \sim \text{Dir}(\beta_{kw} | \eta_w) \).
   b) Generate \( \mu_k \sim \exp\left[ -\lambda_{\mu} \left( \rho |\mu_k|^2 + (1 - \rho) |\mu_k| \right) \right] \).

2) For each timestamp \( t \in \{1, \ldots, T\} \):
   a) For each document \( d \in \{1, \ldots, D\} \) | \( \delta_{t_d, t} = 1 \):
      i) Generate \( \theta_{dk} \sim \text{Dir}(\theta_{dk} | \alpha_k) \).
      ii) For each word token \( n \in \{1, \ldots, N_d\} \):
         A) Generate \( z_{dnw,k} \sim \text{Mult}(z_{dnw,k} | \theta_{dk}) \).
         B) Generate \( w_{dn} \sim \text{Mult}(w_{dn} | \beta_{z_{dnw,k}w_{dn}}) \).
   b) Generate \( y_t \sim N(y_t | \mu_0 + \sum_k \mu_k \theta_{dk}, \sqrt{T}) \).

This way, the same model that generates the topics will also generate the time series in a joint manner. Notice that we are not normalizing the mean of the Gaussian by the number of documents at each time \( t \), \( N_t \) (as done in [22]), because we assume that the absolute number of news should have an impact on the time series. This way, we allow a null impact in a day with little or no news.

The generation of the time series \( y_t \) from the topic presence over time \( \theta_{dk} \) has an additional interpretation in light of the central limit theorem: if we had assumed that each document \( d \) published at time \( t \) had a certain impact on the value of \( y_t \) (which we could model by a normal distribution with mean \( \mu_0 + \sum_k \mu_k \theta_{dk} \)), then the combined impact of all the documents published at that time would result in a Gaussian mixture. However, by applying the central limit theorem, this Gaussian mixture could be reduced to a single Gaussian distribution with mean \( \mu_0 + \sum_k \mu_k \theta_{dk} \), finally arriving at the same result.

#### 2) MODEL LOG-LIKELIHOOD

From the generative process described above, we arrive at the following log-likelihood:

\[
\mathcal{L} = \sum_{d=1}^{D} \sum_{w=1}^{V} n_{dw} \log \left( \frac{K}{k=1} \theta_{dk} \beta_{kw} \right) + \sum_{d=1}^{D} \sum_{k=1}^{K} \alpha_k \log \theta_{dk} + \sum_{w=1}^{V} \eta_w \log \beta_{kw} - \frac{1}{2T} \sum_{t=1}^{T} \left( y_t - \mu_0 - \sum_{k=1}^{K} \theta_{dk} \mu_k \right)^2 + \lambda_{\mu} \left\{ \frac{1 - \rho}{2} \sum_{k=1}^{K} \mu_k^2 + \rho \sum_{k=1}^{K} |\mu_k| \right\}.
\]

This log-likelihood can be seen as the joint probability of both (1) and (2). Notice the minus sign that precedes the second term: the optimal likelihood is obtained when the LDA log likelihood is maximized and the squared error of
Apart from the fact that we had to add the ad-hoc term, (5) presents two important problems. First, the order of magnitude of the LDA log-likelihood could be different to that of the sum of squares error of the time series; if that were the case, then the numerical optimization would be driven almost entirely by one term or the other, rather than a balanced mixture of both. Second, in a regularized linear regression problem the solutions are not equi-variant under scaling of the input, so it is typically beneficial to standardize. In order to solve these two problems, we propose the following log-likelihood:

\[
\mathcal{L} = \sum_{d=1}^{D} \sum_{w=1}^{V} n_{dw} \log \left( \sum_{k=1}^{K} \theta_{dk} \beta_{kw} \right) + \sum_{d=1}^{D} \sum_{k=1}^{K} \alpha_k \log \theta_{dk} + \sum_{w=1}^{V} \sum_{k=1}^{K} \eta_w \log \beta_{kw}
\]

\[
- \frac{N_C}{1000} \tau \left[ \frac{1}{2T} \sum_{t=1}^{T} \left( \hat{y}_t - \mu_0 - \sum_{k=1}^{K} \theta_{tk} \hat{\mu}_k \right)^2 \right] + \lambda \sum_{k=1}^{K} \left[ \frac{1}{2} \sum_{k=1}^{K} \hat{\mu}_k^2 + \rho \sum_{k=1}^{K} |\hat{\mu}_k| \right],
\]

where \( \tau \) is a coupling hyperparameter used for weighting the relative importance of the regression term with respect to the LDA log-likelihood (a similar argument is employed in [23]).

This hyperparameter has been multiplied by the total number of tokens \( N_C = \sum_{d=1}^{D} n_{dw} \) with the goal of making it invariant across corpora of different sizes, and divided by 1000 for readability purposes. In order to understand the role of \( \tau \), consider the limit case when \( \tau \to 0 \): in this case, we recover the LDA log-likelihood, thus neglecting the curve fitting part. As we increase the value of \( \tau \), the regression part becomes more relevant and the optimization surface changes towards that of a linear regression. In the opposite limit case, \( \tau \to \infty \), the optimization would be dominated by the linear regression and no effort would be put into the creation of topics.

In order to put all predictors on a common scale, (6) also introduces scaling for the regressors and their parameters, which we denoted by a hat:

\[
\hat{\theta}_{tk} = \frac{\theta_{tk}}{\sigma_k}, \quad \hat{\mu}_k = \frac{\mu_k}{\sigma_k}, \quad \sigma_k = \sqrt{\frac{1}{T} \sum_{t=1}^{T} (\hat{y}_t - \bar{\hat{y}})^2},
\]

and \( \bar{\hat{y}} \) denotes the mean over time

\[
\hat{\theta}_k = \frac{1}{T} \sum_{t=1}^{T} \theta_{tk}.
\]

Rescaling of topics is a novelty of our method which is not present in any other model cited in section II. This rescaling makes all topic time series equivalent for regularized linear regression, thus avoiding excessive penalization of the topics that appear less frequently.

We are also assuming that the input data, \( \hat{y}_t \), has been previously scaled and its mean removed. We refer the reader to Table 1 for a complete summary of all the variables described above, as well as their domains.
C. MAXIMUM LIKELIHOOD ESTIMATE OF THE PARAMETERS

In order to optimize (6), we will make use of the expectation-conditional maximization (ECM) algorithm [24], an adaptation of the expectation-maximization (EM) algorithm in which the M step is decomposed into a series of conditional maximization, or CM, steps where each parameter is maximized individually, assuming that the other parameters remain fixed. The output of each CM-step will act as the input of the next one. It is guaranteed that, for each step, the likelihood will monotonically increase. Thanks to this property, the steps taken to increase the LDA part of the likelihood will not be detrimental for the optimization of the squared error of the elastic net regression, and vice versa.

1) E-STEP
Let us consider the $n^{th}$ iteration of this process (we assume that in the first iteration of the algorithm, $n = 0$, all the parameters were randomly initialized). In the E-step, we introduce a latent variable $z_{dwk}$ that assigns word $w$ in document $d$ to topic $k$. Applying the Bayes rule with fixed parameters at the $n^{th}$ iteration, we can estimate the mean value of $z_{dwk}^{(n)}$,

$$\langle z_{dwk} \rangle^{(n)} = \frac{\theta_{dk}^{(n)} \beta_{kw}^{(n)}}{\sum_{j=1}^{K} \theta_{dj}^{(n)} \beta_{jw}^{(n)}}.$$  \hspace{1cm} (11)

This result, which is the same that one would obtain in standard LDA, allows us to overcome the difficulty arising from the presence of summation over $k$ that appears inside the logarithm in (6). Indeed, the expected value of the complete-data likelihood can then be written as

$$\langle L \rangle_{z} = \sum_{d=1}^{D} \sum_{w=1}^{V} \sum_{k=1}^{K} n_{dw} \langle z_{dwk} \rangle^{(n)} \left( \log \theta_{dk}^{(n)} + \log \beta_{kw}^{(n)} \right)$$

$$+ \sum_{d=1}^{D} \sum_{k=1}^{K} \alpha_k \log \theta_{dk}^{(n)} + \sum_{w=1}^{V} \sum_{k=1}^{K} \eta_w \log \beta_{kw}^{(n)}$$

$$\quad - \frac{N_C}{1000} \tau \left[ \frac{1}{2T} \sum_{t=1}^{T} \left( \hat{y}_t - \mu_{0}^{(n)} - \sum_{k=1}^{K} \hat{\theta}_{tk}^{(n)} \mu_{k}^{(n)} \right)^2 \right]$$

$$\quad + \lambda_{\mu} \left[ \frac{1}{2} \sum_{k=1}^{K} \left( \hat{\sigma}_k^{(n)} \right)^2 + \rho \sum_{k=1}^{K} \left( \hat{\mu}_k^{(n)} \right)^2 \right]$$  \hspace{1cm} (12)

with $\langle z_{dwk} \rangle$ given by (11).

2) FIRST CM-STEP
In the first CM-step, we maximize (12) with respect to $\beta_{kw}^{(n+1)}$ and $\theta_{dk}^{(n+1)}$, taking $\mu_k^{(n)}$ as a fixed parameter. The optimization of $\beta_{kw}^{(n+1)}$ is immediate:

$$\beta_{kw}^{(n+1)} = \frac{\sum_{d=1}^{D} n_{dw} \langle z_{dwk} \rangle^{(n)}}{\sum_{w=1}^{V} \sum_{d=1}^{D} n_{dw} \langle z_{dwk} \rangle^{(n)} + \eta_w}.$$  \hspace{1cm} (13)

This is the well-known solution for $\beta_{kw}^{(n+1)}$ that is obtained when LDA is optimized by using the EM algorithm, and we have recovered it here in its exact same form. On the other hand, due to the appearance of $\theta_{dk}$ in the regression term of (6), which is responsible for the coupling that characterizes TSITM, the computation of $\theta_{dk}^{(n+1)}$ is not simple and will be different from the standard result that one would expect in LDA. The optimization problem in this case is

$$\theta_{dk}^{(n+1)} = \arg\min_{\theta_{dk} \geq 0, \sum_k \theta_{dk} = 1} \left[ \frac{1}{2T} \sum_{t=1}^{T} \left( \hat{y}_t - \mu_{0}^{(n)} - \sum_{k=1}^{K} \hat{\theta}_{tk}^{(n)} \mu_{k}^{(n)} \right)^2 \right]$$

$$\quad + \sum_{d=1}^{D} \sum_{k=1}^{K} \alpha_k \log \theta_{dk}^{(n)}$$

$$\quad - \frac{N_C}{1000} \tau \left[ \frac{1}{2T} \sum_{t=1}^{T} \left( \hat{y}_t - \mu_{0}^{(n)} - \sum_{k=1}^{K} \hat{\theta}_{tk}^{(n)} \mu_{k}^{(n)} \right)^2 \right]$$

$$\quad + \lambda_{\mu} \left[ \frac{1}{2} \sum_{k=1}^{K} \left( \hat{\sigma}_k^{(n)} \right)^2 + \rho \sum_{k=1}^{K} \left( \hat{\mu}_k^{(n)} \right)^2 \right].$$

TABLE 1. Table of variables with descriptions and domains.

| Variable | Description | Variable domain |
|----------|-------------|----------------|
| $D$      | Total number of documents in the corpus | $N$ |
| $N_d$    | Total number of tokens in document $d$ | $N$ |
| $N_C$    | Total number of tokens in the corpus | $N$ |
| $V$      | Total number of words in the vocabulary | $N$ |
| $T$      | Total number of times in the time series | $N$ |
| $K$      | Total number of topics | $N$ |
| $\theta_{dk}$ | The document topic matrix | $D \times K$-dimensional simplices |
| $\beta_{kw}$ | The topic term matrix | $K \times V$-dimensional simplices |
| $z_{dwk}$ | Latent variable indicating which topic generated each token | $\{0, 1\}$ |
| $\theta_{tk}$ | Time series of the topics | $\mathbb{R}^{T \times K}$ |
| $\gamma_t$ | Time series | $\mathbb{R}$ |
| $\alpha$ | Hyperparameter for the prior of $\theta$ | $\mathbb{R}^+$ |
| $\eta$ | Hyperparameter for the prior of $\beta$ | $\mathbb{R}^+$ |
| $\tau$ | Hyperparameter weight of LDA vs elastic net | $\mathbb{R}^+$ |
| $\lambda_{\mu}$ | Hyperparameter for the regularization of elastic net | $\mathbb{R}^+$ |
| $\rho$ | Hyperparameter weight of Lasso vs Ridge | $[0, 1]$ |
\[ +\lambda \mu \left\{ \frac{1 - \rho}{2} \sum_{k=1}^{K} (\mu_k(n))^2 \sigma_k^2 + \rho \sum_{k=1}^{K} |\mu_k(n)| \sigma_k \right\} \], \tag{14} \]

where \( \theta_{dk} \) appears both explicitly and implicitly (through the \( \theta_{tk} \) and \( \sigma_k \) terms in the regression term, defined by (4) and (9) respectively). To our knowledge, there is not a closed form solution for (14). Nevertheless, this parameter can be optimized by noticing that we are dealing with a constrained convex optimization problem, since the nonnegative sum of convex functions is a convex function itself, and both negative logarithms, quadratic functions and norms are convex. For this type of problems, the global optimum can be found efficiently by using numerical methods. In particular, we will make use of proximal gradient algorithms [25]. We refer the reader to Appendix I for a complete description of the numerical optimization that we have employed here, which is summarized in Algorithm 3.

3) SECOND CM-STEP

In the second CM-step, we minimize with respect to \( \mu_k^{(n+1)} \) taking \( \beta_{kw}^{(n+1)} \) and \( \theta_{tk}^{(n+1)} \) as fixed parameters. Notice that \( \mu_k^{(n+1)} \) only appears in the second term of (6), so the task can be reduced to solving an elastic net regression problem like the one described in (2):

\[
\begin{align*}
\mu_k^{(n+1)} &= \arg\min_{\mu_k} \left\{ \frac{1}{2T} \sum_{t=1}^{T} \left( \hat{y}_t - \mu_0 - \theta_{tk}^{(n+1)} \mu_k \right)^2 
+ \lambda \mu \left\{ \frac{1 - \rho}{2} \sum_{k=1}^{K} (\mu_k^{(n+1)})^2 \sigma_k^2 + \rho \sum_{k=1}^{K} |\mu_k^{(n+1)}| \sigma_k \right\} \right\} \tag{15}
\end{align*}
\]

There is not a closed form solution for this optimization either, but it is a well-known quadratic programming problem that can be solved efficiently by using standard methods which have already been discussed in the literature [26], [27] (implemented in Python packages such as scikit-learn [28]) and that are immediately applicable here.

The E-CM steps described above must be repeated several times until a given convergence criterion for \( \mathcal{L} \) is satisfied. We summarize the process in Algorithm 1.

V. DESCRIPTION OF THE DATASETS

We will evaluate our algorithm on different corpora extracted from Reuters News with multiple time series data. We have divided the experimental discussion into two main use cases, corresponding to the following datasets:

- First, we will consider a corpus of political news and the US president’s disapproval ratings as a time series. The discussion of results for this use case will be mainly qualitative: we will focus on a particular period where the disapproval ratings exhibited a sharp peak and turn our attention to the obtained topics to see if we can interpret the events that drove the time series up to those levels.
- Second, we will consider a corpus of economic news and stock market data as a time series. The discussion of results for this use case will be mainly quantitative: the idea is to explore a broad range of companies and time periods to see if we are able to consistently find reliable correlations and thus validate our model in different scenarios.

A detailed description of the datasets and the preprocessing of both text and numerical data is presented below.

A. US PRESIDENT’S DISAPPROVAL RATINGS WITH POLITICAL NEWS

Presidential ratings are good candidates for finding correlations with text, since we expect that the support for governments may fluctuate based on political news. In order to avoid as much bias as possible, we will be using the FiveThirtyEight president’s disapproval rating, a daily time series which is obtained by averaging a comprehensive set of polls coming from different sources.\(^3\) Polls are weighted based on their methodological standards and historical accuracy, and also adjusted for house effects if they consistently show different results from the polling consensus. According to this averaged rating, the highest peak of the US president’s disapproval rating among likely or registered voters during 2019 occurred on the Jan 27th, with a 55.58% disapproval rating. Our goal in this experiment is to try to explain what kind of topics drove the disapproval ratings up to these levels according to the TSITM model. In order to do so, we will focus our analysis on the first quarter of 2019 (see Fig. 2).

1) TEXT DATA PREPARATION

The corpus for this experiment is made of documents in English extracted from Reuters News between 01/01/2019

\(^3\)https://fivethirtyeight.com/features/how-were-tracking-donald-trumps-approval-ratings/
and 03/31/2019. These documents have been retrieved from the Factiva database\footnote{https://professional.dowjones.com/factiva/} filtering by the “Politics/International Relations” category and the “United States” region; in order to polish this corpus, we have removed semi-automatic news, summaries and tabular information by deleting all the documents belonging to the following Factiva categories: “Routine General News”, “Routine Market/Financial News”, “Economic Performance/Indicators” and “News Digests”. Preprocessing of the resulting corpus consists of the following steps:

1) We remove standard English stop-words by looking at a list of 683 common words.
2) We use the \texttt{stanza} package \cite{29} to tokenize the documents, label the resulting tokens with their universal POS tags and generate the word lemmas. At this point, we also remove numbers and tokens whose POS tag is different from PROPN, NOUM, VERB or ADJ.
3) We drop documents with less than 50 tokens.
4) We use the \texttt{gensim} package \cite{30} to create bigrams from pairs of words that appears at minimum 5 times in the corpus.
5) We filter out tokens that appear less than 20 times in the corpus or that appear in more than 80% of the documents.
6) We create a bag-of-words model to obtain a \texttt{n\_doc} count matrix, where \texttt{d} is the number of documents and \texttt{w} the number of words of the vocabulary.
7) Finally, we remove duplicated documents by eliminating those that show a cosine similarity higher or equal to 95%.

The resulting dataset is made of 5419 documents and a vocabulary of 8329 terms. Since we are dealing with news, publication timestamps are readily available. We have truncated these timestamps to the unit of days to obtain a daily series of documents.

2) NUMERICAL DATA PREPARATION

We have downloaded the series of the US president’s disapproval rating based on polls of likely or registered voters between 01/01/2019 and 03/31/2019 from the FiveThirtyEight\footnote{https://projects.fivethirtyeight.com/trump-approval-ratings/} webpage, depicted in Fig. 2. Since we are interested in seeing how much the topics push up or down the ratings, we differentiate the numerical series to obtain the returns, \( r_t \), defined as:

\[
 r_t = \frac{p_t - p_{t-1}}{p_{t-1}}. \tag{16}
\]

We then standardize it by removing the mean and scaling it to unit variance. The resulting time series is shown as the color gray line in Fig. 4.

B. STOCK MARKET DATA WITH ECONOMIC NEWS

Stock market data is a publicly available type of time series that is expected to be sensitive to events described in the news (and vice versa). However, not all companies are expected to present correlations with the news during the same periods. Therefore, we are going to perform four independent sets of experiments, one for each of the four quarters of the year 2019, with a range of the top companies (measured by capitalization) of the Nasdaq stock market. We expect to find signals for different companies in each quarter.

1) TEXT DATA PREPARATION

We have created a corpus of documents in English extracted from Reuters News between 01/01/2019 and 12/31/2019, split in quarters of three months each. As in the political dataset, these documents have been retrieved from the Factiva database, but in this case using filters more appropriate for the financial market: we have included all the documents belonging to the category “Economic News” and region “United States” and then removed routine news by excluding the categories “Routine General News”, “Routine Market/Financial News” and “Economic Performance/Indicators”.

For the preprocessing, we have followed the exact same procedure described in section V-A1. As in the previous case, we have truncated publication timestamps to the unit of days; however, since trading hours for the Nasdaq stock market are from 9:30 a.m. to 4 p.m. (Eastern time) on weekdays, we have assigned the next day as date for all the news published after 4 p.m. and we have removed weekends and holidays.

After this process, we end up with the following datasets: for the first quarter, there are 1502 documents and a vocabulary of 3045 terms; for the second quarter, there are 2061 documents and a vocabulary of 3677 terms; for the third quarter, there are 1986 documents and a vocabulary of 3712 terms; for the last quarter, there are 2054 documents and a vocabulary of 3673 terms. Notice that the amount of news about economic issues is smaller than the amount of news about politics and international relations, but thanks to the scale introduced in (6) we don’t expect huge changes in the optimal value of \( \tau \).
2) NUMERICAL DATA PREPARATION
We have downloaded the series of stock’s close prices adjusted for splits, \( p_t \), for the top companies of the Nasdaq stock market between 01/01/2019 and 12/31/2019, split in quarters of three months each. In order to make these series stationary and scale-invariant, we differentiate them and obtain the series of financial returns, as defined in (16).

We assume that the series of returns for a given company will be determined by a mix of factors inherent to the general movement of the market and factors inherent to the peculiarities of each company. To remove the general movement of the market, we have performed a standard linear regression, where we have taken the Nasdaq-100 composite index as the market explanatory variable. The predicted returns obtained using this model, \( r'_t \), are subtracted from the observed returns, \( r_t \), to give the residuals \( y_t \). These residuals should be seen as the part of the returns that cannot be explained by the general movement of the market.

Finally, we standardize the residuals \( y_t \) by removing the mean and scaling the series to unit variance. We will take the resulting time series \( \hat{y}_t \) as the numerical data for our experiment, with the hope of predicting it by looking at the news.

Notice that if we had been able to perfectly predict the movement of the stock values of individual companies by using the time series of the market, then the residuals would be zero everywhere: the finite value of the residuals represent the error in predicting the stock values by merely using financial information. By introducing texts to predict the residuals, we expect to contribute to the reduction of this error, effectively explaining something that was not captured by the market prediction.

VI. EXPERIMENTAL SET-UP
A. METRICS
There is not a universally valid and objective metric to assess the performance of topic models [31]. However, our framework provides a simple way to measure the goodness of fit of TSITM thanks to the supervised nature of linear regression. We will train TSITM on a train set composed of documents and the associated time series, and then we will use a test set of documents to predict the values of the time series for them (see Fig. 3 for a system overview). We will then compare the predicted time series with the actually observed values, and compute metrics to measure the prediction error. There are many possible metrics to quantify the quality of a linear regression, but we will restrict ourselves to the three most common:

- The mean squared error (MSE), defined as
  \[
  \text{MSE} = \frac{1}{T_{\text{test}}} \sum_{t=1}^{T_{\text{test}}} (\hat{y}_t^{\text{obs}} - \hat{y}_t^{\text{pred}})^2, \tag{17}
  \]
  where \( T_{\text{test}} \) is the total number of observations in the test set, \( \hat{y}_t^{\text{obs}} \) is the observed value of the time series at time \( t \), and \( \hat{y}_t^{\text{pred}} \) is the value predicted by the model for the time series at time \( t \). Since TSITM optimizes a quadratic type of error (see (6)), the MSE is the main metric that we will be monitoring. The lower the MSE, the better the ability to fit the data.
- The mean absolute error (MAE), defined as
  \[
  \text{MAE} = \frac{1}{T_{\text{test}}} \sum_{t=1}^{T_{\text{test}}} |\hat{y}_t^{\text{obs}} - \hat{y}_t^{\text{pred}}|, \tag{18}
  \]
where the same notation as before is used. The MAE gives a complementary idea of the error, since it is linear instead of quadratic (therefore, all errors are weighted equally). The lower the MAE, the better the ability to fit the data.

- The out-of-sample coefficient of determination ($R^2_{\text{OS}}$), defined as

$$R^2_{\text{OS}} = 1 - \frac{\sum_{t=1}^{T_{\text{train}}} (\gamma_{\text{obs}}^t - \gamma_{\text{pred}}^t)^2}{\sum_{t=1}^{T_{\text{train}}} (\gamma_{\text{obs}}^t - \gamma_{\text{train}}^t)^2},$$

(19)

which differs from standard $R^2$ in that the mean value of the train set,

$$\langle \gamma_{\text{obs}} \rangle_{\text{train}} = \frac{\sum_{t'=1}^{T_{\text{train}}} \gamma_{\text{obs}}^t}{T_{\text{train}}},$$

(20)

is used in the denominator, rather than the mean value of the test set (as one would do for standard $R^2$). In the previous equation, we have denoted $T_{\text{train}}$ the total number of observations in the train set. The $R^2_{\text{OS}}$ is usually preferred over the standard $R^2$ when measuring the predictive power of a model over a test set. The $R^2_{\text{OS}}$ also measures a quadratic error (as in MSE), but it makes it easier to gauge the performance of the model since it does not depend on the scale of the response variable. The higher the $R^2_{\text{OS}}$ (with a maximum value of 1), the better the ability to fit the data.

### B. BASELINES

1) INTRINSIC BASELINE

As our first baseline, we are going to consider an intrinsic prediction that only makes use of the time series data (i.e. text data will not be taken into account). With this baseline, we would like to answer the following question: is it worth using topics as regressors for the time series? If the way our model processes the information contained in the news were not useful to predict the time series, then it would only introduce noise, therefore increasing (or not significantly affecting) the prediction error with respect to this intrinsic baseline.

Resembling the definition of the $R^2_{\text{OS}}$ metric (19), a good intrinsic baseline can be obtained by predicting that, at any time $t$ of the test set, the value of the time series will be equal to its mean value on the train set

$$\gamma_{\text{pred}}^t = \langle \gamma_{\text{obs}} \rangle_{\text{train}},$$

(21)

where $\langle \gamma_{\text{obs}} \rangle_{\text{train}}$ is given by (20). In particular, notice that for this intrinsic baseline the $R^2_{\text{OS}}$ will always be zero (by definition).

A clear understanding of this baseline can be achieved by looking at the sum-of-squares term in (2). If no external regressors are considered (i.e. $\sum_{k=1}^{K} X_{tj} \mu_j = 0$), then the problem reduces to optimizing $\sum_{t=1}^{T} (\gamma_t - \mu_0)^2$ with respect to $\mu_0$. It is straightforward to see that the solution for this problem is our proposed intrinsic baseline.

2) sLDA (Approach 1)

As an external baseline that also takes into account text data, we will consider the sLDA model [14] discussed in section II-B1, since it is the most widespread model for supervised topic modeling tasks and it has been implemented in many programming languages. In particular, we will make use of the `tomotopy` package for Python, which implements sLDA by making use of Gibbs sampling.

However, comparisons between sLDA and TSITM are not straightforward because in sLDA a single prediction is made for each document, while in TSITM we aggregate all the documents published in a given period of time and make a single prediction from them. We have considered two possible approaches to make these models comparable.

As a first approach, we follow the strategy described in [15] and combine all news articles published in a given period into a single document. We then label that unified document with the time series value for that period. This way, sLDA will make a single prediction for a group of documents, as in TSITM.

3) sLDA (Approach 2)

The second approach to make sLDA comparable with TSITM consists on making different predictions for each document and then averaging the error for each period of time. For example, if $n$ documents are published in the same day, we make $n$ predictions with sLDA, compute the regression errors with respect to the same time series value for each of them, and then average them to get a single error. This resulting error could then be compared with the error that is obtained with TSITM for that day.

### C. HYPERPARAMETER DETERMINATION

In order to determine the hyperparameters of TSITM (that is, $\tau$, $\lambda$, $\rho$, $\eta$ and $\alpha$), we will perform a cross-validation experiment consisting of a repeated random subsampling with 5 samples made of 10% of the training dates. We will choose the combination of hyperparameters that yields the best mean validation error. It is well known that different initializations for the optimization problem of a topic model will yield different results due to the presence of several local minima, so we will previously train 10 different LDA models with different initializations each time and will select the random seed that yields the highest log-likelihood.

For an efficient selection of hyperparameters, we have used a Bayesian Optimization [32] with a Gaussian process regression as a method for statistical inference [33] and a probabilistic mixture of negative expected improvement, negative probability of improvement and lower confidence bound as an acquisition function. We used the implementation provided by the `scikit-optimize` package [34]. As a range of possible values for $\tau$ (which is the most relevant hyperparameter for this discussion), we have set $0 < \tau \leq 5$.

For sLDA, we will use the same $\alpha$ and $\eta$ that was obtained for TSITM to make comparisons as close as possible.
D. DATA SPLITs
In order to avoid spurious correlations, we repeat 5 times each experiment, selecting different train-test splits each time. This way, we mitigate the possibility of incurring a selection bias by overfitting a particular train-test split. If the signals discovered by our model are robust enough, then we should expect good results across different train-test splits.

Although TSITM could be adapted for the purpose of forecasting the future values of the time series under certain conditions, this is not the main focus of our experimental set-up. As we already mentioned in section III, we aim instead at discovering signals during a specified period of time in order to explain the events observed in the numerical series from the corresponding topics. Since the topics that appear in news agencies like Reuters tend to vary significantly with time, the signals discovered by TSITM might vanish outside that specified period. Thus, the test set will consist of a hold-out set of dates randomly chosen from the given period.

VII. EXPERIMENTAL RESULTS
A. RESULTS FOR THE US PRESIDENT’S DISAPPROVAL RATINGS WITH POLITICAL NEWS
In this section, we present the results for the dataset described in section V-A. As we said before, we have repeated the same experiment 5 times, performing different train-test splits each time. This way, we expect to detect statistically significant signals and exclude accidental correlations.

In Table 2 we report the metrics values for TSITM and the three baselines for each data split. TSITM has achieved top performance for most metrics in most of the splits. In particular, we observe that its MSE is smaller than that of the intrinsic baseline in 4 out of 5 splits. It is interesting to see that, in this dataset, the external baselines actually perform worse than the intrinsic baseline (as demonstrated by the fact that \( R^2_{\text{TSITM}} \) is negative in most of the splits). This is probably due to the lack of a regularization mechanism in sLDA, which may result in severe overfitting if the number of topics is not adjusted ad-hoc. Similar results are obtained for the MAE. In Table 9 we report the value of \( \tau \) used for each TSITM. This parameter will inform us about how strong the coupling between the topic model and the regression is. Its mean value across splits is of 1.54.

In Table 3, we show the top 10 words of the topics generated with the TSITM model that yielded the best MSE and their impact coefficient, \( \hat{\mu}_k \). This result reveals well-known issues which are widely considered to be relevant for the popularity of the US president. We see positive correlations between disapproval rates and two topics that determined the political debate during the first months of 2019: news regarding the federal government shutdown which took place from December 22, 2018, to January 25, 2019, gave place to a topic with a strong impact coefficient \( \hat{\mu} = 0.19 \) and news regarding the Mueller report on Russian interference in the presidential campaign gave place to a topic with a weak impact coefficient \( \hat{\mu} = 0.02 \). According to the TSITM model, the peak of disapproval rating observed during this period should be attributed to these two topics. This analysis seems to coincide with the conclusions reached by human analysts at FiveThirtyEight during that period. On the other hand, we see a strong negative correlation (\( \hat{\mu} = -0.25 \)) between disapproval rates and a topic regarding the president’s warnings to OPEC pressuring to reduce the price of oil (which hit record low prices in this quarter after US sanctions on Iranian exports) as well as trade wars with China and India. This qualitative analysis illustrates that our model can converge to issues that were previously expected to affect the time series.

The prediction made by our model, \( \hat{y}^{\text{pred}} \), (based solely on the presence of the three topics discussed above over time, \( \theta_{\text{dis}} \), and their impact coefficient \( \hat{\mu}_k \)) is depicted as a black color line in Fig. 4, where we compare it with the observed numerical data, \( y_{\text{obs}} \) (after the preprocessing discussed in section V-A2), depicted in gray.

---

TABLE 2. Results for the politics dataset for the 1st quarter of 2019 and US president’s disapproval rating (best results for each metric in bold).

| Split | MSE Intrinsic Baseline sLDA Approach 1 sLDA Approach 2 TSITM MAE Intrinsic Baseline sLDA Approach 1 sLDA Approach 2 TSITM R² Intrinsic Baseline sLDA Approach 1 sLDA Approach 2 TSITM |
|-------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1     | 1.03           | 1.16           | 1.04           | **0.96**       | 0.84           | 0.89           | 0.85           | 0.78           | 0.00           | -0.13          | -0.01          | 0.05           |
| 2     | 0.36           | 0.69           | 0.39           | **0.28**       | 0.48           | 0.78           | 0.51           | 0.41           | 0.00           | -0.93          | -0.08          | 0.22           |
| 3     | 0.21           | 0.20           | 0.23           | **0.12**       | 0.31           | 0.35           | 0.35           | 0.27           | 0.00           | 0.04           | -0.07          | 0.45           |
| 4     | 0.79           | 0.83           | 0.80           | **0.69**       | 0.58           | 0.65           | 0.58           | 0.65           | 0.00           | -0.05          | -0.01          | 0.12           |
| 5     | **0.52**       | 0.62           | 0.56           | 0.54           | 0.44           | 0.51           | 0.46           | **0.43**       | **0.00**       | -0.18          | -0.07          | -0.04          |

TABLE 3. Topics and their impact on the US president’s disapproval rating and 1st quarter news.

| \( \hat{\mu} \) | Top words |
|---------------|-----------|
| 0.19          | congress, democrats, shutdown, wall, senate, president, house, republican, government, lawmaker |
| 0.02          | investigation, report, mueller, committee, special_counsel, russia, campaign, president, cohen, barr |
| -0.25         | oil, export, market, percent, import, india, price, china, soybean, trade |

---

6https://fivethirtyeight.com/features/americans-increasingly-blame-trump-for-the-government-shutdown/
7https://fivethirtyeight.com/features/independents-trust-mueller-which-could-be-bad-news-for-trump/
8https://www.eia.gov/todayinenergy/detail.php?id=42415
TABLE 4. Results for the economic dataset for each quarter of 2019 and different companies (best results for each metric in bold).

| Split # | MSE | Intrinsic Baseline | sLDA Approach 1 | sLDA Approach 2 | TSITM | MAE | Intrinsic Baseline | sLDA Approach 1 | sLDA Approach 2 | TSITM | $R^2_{OS}$ | Intrinsic Baseline | sLDA Approach 1 | sLDA Approach 2 | TSITM |
|--------|-----|--------------------|-----------------|-----------------|--------|------|--------------------|-----------------|-----------------|--------|---------|--------------------|-----------------|-----------------|--------|
| QT1 - Economic | 1 | 0.49 | 1.09 | 0.56 | 0.37 | 0.52 | 0.88 | 0.61 | 0.53 | 0.00 | -1.20 | -0.14 | 0.26 |
|          | 2 | 0.90 | 0.65 | 1.01 | 0.88 | 0.66 | 0.60 | 0.71 | 0.74 | 0.00 | 0.28 | -0.13 | 0.02 |
|          | 3 | 3.54 | 3.22 | 3.33 | 2.94 | 1.18 | 1.18 | 1.18 | 1.06 | 0.00 | 0.09 | 0.06 | 0.17 |
|          | 4 | 0.94 | 0.48 | 0.92 | 0.54 | 0.80 | 0.57 | 0.83 | 0.65 | 0.00 | 0.50 | 0.02 | 0.43 |
|          | 5 | 3.43 | 3.04 | 3.28 | 3.02 | 1.21 | 1.31 | 1.20 | 1.10 | 0.00 | 0.11 | 0.04 | 0.12 |
| QT2 - Economic | 1 | 2.01 | 1.82 | 1.93 | 1.94 | 1.18 | 1.12 | 1.13 | 1.21 | 0.00 | 0.10 | 0.04 | 0.03 |
|          | 2 | 0.43 | 0.54 | 0.48 | 0.37 | 0.52 | 0.63 | 0.58 | 0.55 | 0.00 | -0.25 | -0.11 | 0.14 |
|          | 3 | 0.83 | 0.78 | 0.62 | 0.82 | 0.75 | 0.82 | 0.64 | 0.71 | 0.00 | 0.05 | 0.25 | 0.01 |
|          | 4 | 0.53 | 0.47 | 0.36 | 0.43 | 0.65 | 0.55 | 0.51 | 0.58 | 0.00 | 0.12 | 0.32 | 0.19 |
|          | 5 | 0.97 | 1.70 | 1.29 | 0.81 | 0.86 | 1.00 | 0.98 | 0.70 | 0.00 | -0.76 | -0.33 | 0.16 |
| QT3 - Economic | 1 | 1.54 | 1.39 | 1.58 | 0.94 | 0.90 | 0.90 | 0.90 | 0.81 | 0.00 | 0.10 | -0.02 | 0.39 |
|          | 2 | 0.56 | 0.89 | 0.61 | 0.36 | 0.59 | 0.85 | 0.61 | 0.43 | 0.00 | -0.58 | -0.08 | 0.37 |
|          | 3 | 0.24 | 0.25 | 0.28 | 0.29 | 0.45 | 0.47 | 0.46 | 0.37 | 0.00 | -0.03 | -0.20 | -0.21 |
|          | 4 | 1.79 | 1.66 | 1.91 | 1.55 | 0.85 | 0.82 | 0.93 | 0.77 | 0.00 | 0.07 | -0.07 | 0.14 |
|          | 5 | 0.48 | 0.65 | 0.51 | 0.21 | 0.64 | 0.73 | 0.66 | 0.44 | 0.00 | -0.36 | -0.07 | 0.56 |
| QT4 - Economic | 1 | 1.32 | 1.76 | 1.40 | 1.27 | 0.94 | 1.01 | 0.98 | 0.95 | 0.00 | -0.34 | -0.06 | 0.04 |
|          | 2 | 1.16 | 1.88 | 1.22 | 1.10 | 0.92 | 1.05 | 0.93 | 0.86 | 0.00 | -0.61 | -0.05 | 0.05 |
|          | 3 | 0.20 | 0.22 | 0.23 | 0.17 | 0.31 | 0.42 | 0.36 | 0.33 | 0.00 | -0.12 | -0.15 | 0.16 |
|          | 4 | 1.05 | 1.40 | 1.10 | 1.11 | 0.90 | 0.96 | 0.89 | 0.86 | 0.00 | -0.34 | -0.04 | -0.06 |
|          | 5 | 0.73 | 0.50 | 0.84 | 0.64 | 0.81 | 0.57 | 0.86 | 0.64 | 0.00 | 0.33 | -0.14 | 0.13 |

FIGURE 4. Predicted vs. observed standardized disapproval rate return.

B. RESULTS FOR STOCK MARKET DATA WITH ECONOMIC NEWS

In this section, we present the results for the dataset described in section V-B. For each quarter of 2019, a different company has been selected: Amazon.com, Inc. for the first quarter (QT1), Starbucks Corporation for the second (QT2), PayPal Holdings, Inc. for the third (QT3) and Alphabet Inc. for the last one (QT4). As in the previous section, we have repeated the same experiment 5 times for each quarter, performing different train-test splits each time.

Results are presented in Table 4. Let us first analyze how TSITM performs with respect to the intrinsic baseline. The first thing to note is that TSITM’s MSE is consistently lower (or equivalently, its $R^2_{OS}$ is consistently higher). This can be seen by noting that its $R^2_{OS}$ is positive in 18 out of 20 splits. The MAE is also generally smaller in TSITM than in the intrinsic baseline, although this metric shows less impressive scores (improvement occurs in 14 out of 20 sets). This result can be understood by noting that TSITM optimizes a quadratic type of error. This means that the model is more sensitive to large response sizes in the time series, while the MAE does not exhibit this property. Note that the signals that can be discovered in financial data are typically very small, so one should not be discouraged by the apparently modest values of $R^2_{OS}$ that have been obtained with TSITM.

With respect to the external baselines (sLDA approaches 1 and 2), we note that the lack of regularization in these models leads to volatile results. While they are the best performant models in some splits (particularly, in the QT2 dataset), they do not exhibit consistent reliability across datasets, as proved by the fact that they are not even able to
TABLE 6. Topics and their impact for Starbucks Corporation and 2nd quarter news.

| \( \hat{\mu} \) | Top words                           |
|----------------|------------------------------------|
| 0.14           | mexico, mexican, border, deal, tariff, canada, threat, country, agreement, migrant |
| 0.13           | plan, state, business, tax, fund, investment, pay, create, project, work         |
| 0.11           | president, policy, call, support, go, political, economic, people, campaign, white_house |

TABLE 7. Topics and their impact for PayPal Holdings, Inc. and 3rd quarter news.

| \( \hat{\mu} \) | Top words                           |
|----------------|------------------------------------|
| 0.37           | yield, fall, rise, wednesday, stock, market, tuesday, thursday, bond, data      |
| 0.11           | eu, washington, european, europe, european_union, tax, france, britain, johnson, government |

TABLE 8. Topics and their impact for Alphabet Inc. and 4th quarter news.

| \( \hat{\mu} \) | Top words                           |
|----------------|------------------------------------|
| 0.20           | tax, france, french, company, europe, nato, country, trump, government, budget   |
| −0.11          | sanction, russia, company, oil, russian, ship, export, shipping, port, project   |

TABLE 9. Optimal coupling parameter, \( \tau \), for each dataset and split number.

| Split # | QT1 Politics | QT1 Econ. | QT2 Econ. | QT3 Econ. | QT4 Econ. |
|---------|--------------|-----------|-----------|-----------|-----------|
| 1       | 0.77         | 3.00      | 5.00      | 5.00      | 2.14      |
| 2       | 3.82         | 4.82      | 5.00      | 5.00      | 2.36      |
| 3       | 1.69         | 2.06      | 5.00      | 3.81      | 2.89      |
| 4       | 0.68         | 0.48      | 1.32      | 0.36      | 0.46      |
| 5       | 0.73         | 5.00      | 0.41      | 2.31      | 0.27      |

Average \( 1.54 \), \( 3.07 \), \( 3.35 \), \( 3.30 \), \( 1.62 \)

beat the intrinsic baseline in the vast majority of splits in QT3 and QT4. Furthermore, notice that, in the splits where sLDA does not perform well, predictions are very misleading. For example, while in TSITM the worst \( R^2_{OS} \) is \(-0.21\), this metric goes down to \(-1.21\) in sLDA (approach 1) and to \(-0.33\) in sLDA (approach 2).

In Table 9 we report the value of \( \tau \) used for each model, as well as the mean across the different train-test partitions. These optimal values prove that the model benefits from a certain degree of coupling between topic modeling and linear regression. Actually, we see that in some cases the model might have benefited if an even stronger coupling had been allowed (this seems to be the case for the second and third quarters).

In Tables 5 - 8 we show, for each quarter, the top 10 words of the topics generated with the TSITM model that yielded the best MSE and their impact coefficient, \( \hat{\mu}_k \). Notice that we have only reported the topics with an impact coefficient \( \hat{\mu} \) different from zero, i.e., the topics that present a significant correlation with the time series according to the TSITM model. Although we trained these models with a significantly higher number of topics, the \( L_1 \) constraint of the elastic net regularization drove to zero the coefficient of most of them, as we discussed in section IV-A. This is a significant asset in the context of topic modeling and unsupervised learning, since the determination of the optimal number of topics is usually a human-made choice. For TSITM, on the other hand, one simply has to choose a sufficiently large number of topics (typically, from 10 to 30) and the system drives most of the coefficients to zero.

Lastly, observe that the resulting topics were expected to influence big corporations and multinationals like the ones that we have analyzed: the top words reveal that the stock values of these companies are sensible to international commerce agreements, tariffs, investments, taxes and other economic topics that appear consistently in all the experiments.

C. HYPOTHESIS TESTING

In the previous two sections, we have observed that TSITM tends to perform better than the baselines. However, when comparing two group of samples, the difference in their values may be a result of random variations. In order to claim that the results are significant, we need a statistical proof that our model shows an advantage over the baselines. Hypothesis testing will be used to discard the possibility that differences between models are accidental.

A standard statistical methodology to perform hypothesis testing for model comparison is the following [35]. Let \( M \) and \( M' \) be two models, and let \( X_i \) and \( X'_i \) be the respective metric values for each model on the \( i \)-th data split (\( i = 1 \ldots N \)), where \( N \) is the total number of data splits. We can define the differences in metrics for each split as \( \delta x_i = X_i - X'_i \). The average difference \( \bar{\delta} x \) and the standard deviation \( \sigma_{\delta x} \) are estimated by

\[
\bar{\delta} x = \frac{\sum_{i=1}^{N} \delta x_i}{N}, \quad (22)
\]

\[
\sigma_{\delta x} = \sqrt{\frac{\sum_{i=1}^{N} (\delta x_i - \bar{\delta} x)^2}{N - 1}}, \quad (23)
\]

and the error on the estimated average is \( \sigma_{\delta x} / \sqrt{N} \). To perform hypothesis testing, we assume that the differences \( \delta x_1 \ldots \delta x_N \) are sampled from a Student’s \( t \)-distribution with \( (N - 1) \) degrees of freedom,

\[
t = \frac{\bar{\delta} x - \mu}{\sigma_{\delta x} / \sqrt{N}}. \quad (24)
\]

The null-hypothesis is that the mean \( \mu \) is 0 (i.e. there is no significant difference between models). We compute \( t \) for each dataset and check the corresponding \( p \)-value for a one-tailed test (we want to test whether TSITM performs better than each baseline). A \( p \)-value of 0.05 or less is typically considered as strong evidence that the null-hypothesis can be
discarded, although a p-value as high as 0.10 could also be accepted.

Results of the hypothesis testing are shown in Table 10, where we report the p-value that TSITM performs better than each baseline for each metric in each of the 5 datasets considered (the politics dataset and the four economics datasets), as well as the total results where we consider all the splits from all datasets combined. We observe that our results exhibit statistical significance in most cases. It is specially interesting to look at the results for $R^2_{OS}$, since this metric does not depend on the scale of the response variables and thus hypothesis testing is particularly robust in this case. The p-value with respect to the intrinsic baseline for this metric is smaller than 0.10 for all the datasets. The improved performance of TSITM with respect to the external baseline, sLDA (both approaches 1 and 2) cannot be established that easily for every dataset (QT1 Econ. and QT2 Econ. seem to be problematic), but this may be due to the limited number of splits that we have considered. Indeed, when we consider the total results where all datasets are combined to form a set of 25 splits, the superiority of TSITM with respect to the three baselines is clear: the p-value is below $1 \times 10^{-3}$ for each metric and each baseline. This proves that the best results obtained with TSITM are statistically significant.

VIII. DISCUSSION AND IMPLICATIONS

A. DISCUSSION OF RESULTS

Results presented in the previous section prove that the TSITM model satisfactorily addresses the three research goals stated in section III:

(RG1) Thanks to the elastic net regularization, only the topics that showed significant correlation with the time series were preserved (see Tables 3 and 5 - 8). This alleviates the problem of choosing the appropriate number of topics as an input parameter, as opposed to previous works in which the number of correlations was chosen arbitrarily [17]. This regularization also prevents overfitting of the time series, a problem that appeared in other models such as sLDA [15].

(RG2) The regression coefficients $\mu_k$ objectively quantified the impact of each of these topics on the time series (see Tables 3 and 5 - 8), which helped us interpret the movement of each time series based on the events described by the news. This approach, which can have various applications from a practical perspective, is different from others that appeared in previous works, such as [13].

(RG3) Finally, a prediction of the values of the time series on a hold-out dataset was made by previously training TSITM with the appropriate hyperparameters and feeding the model with unobserved documents. TSITM’s errors were consistently smaller than those of the baseline models (see Tables 2 and 4, as well as the hypothesis testing in Table 10), thus proving that robust correlations between the time series and the news could be found by making use of TSITM. Our approach diverges from previous models, in which a single document is used to predict an entire time series [16] or a fixed time-lag [19].

Notice that we have included standardization of the inputs as a core feature of TSITM. Standardization is a crucial step, since the solutions to a regularized linear regression problem are not equivariant under scaling of the input. This feature was not present in many of the previous works discussed in section II, such as [19].

Compared to other text regression approaches that do not make use of topics (such as those based on neural networks and news sentiments [36], [37], [38]), the main advantage of TSITM is the improved interpretability. Topic modeling allows us to not only make predictions of unobserved values of the time series, but also to understand the reasons behind the predictions and explain why certain news impacted the time series.

We believe that this model may contribute to the quantification of the concept of reputational risk by providing a strategy to objectively determine how much the value of a certain entity (reflected by its stock values or popularity rates) has been affected by adverse events or damages to the entity’s reputation. TSITM could also be employed for reputation polarity analysis [39], one of the core tasks of Online Reputation Management, which consists on determining if the publication of a text about an entity will impact positively or negatively the entity’s reputation.

B. LIMITATIONS

A potential limitation of our experimental framework is that parameters have been optimized with the goal of discovering signals across the entire period, regardless of its position. This could be undesirable if the main goal were to forecast the future (for example, for stock trading purposes) instead of understanding the past, since the resulting signals may be far from the last point of the time series. However, as we explicitly stated in section III, this was not our objective.

Notice that in this work, we do not claim that TSITM is the best way to explain the proposed series (either financial returns or disapproval ratings) from the available information at a given time. For example, presidential disapproval ratings show clear autoregressive features, and the previous day’s rate could be used together with news to improve the results. A similar case could occur with stock market data when there is a varying trend superimposed on the returns. We leave these questions open for a further research work.

Lastly, it must be noted that, as we increase $\tau$, the number of iterations required to achieve a certain degree of convergence for $L$ increases. This is illustrated in Fig. 5, where we plot, for the politics dataset, the number of ECM steps given to obtain the same rate of convergence in $L$ for different values of $\tau$, while keeping all the remaining hyperparameters fixed. The reason behind this behavior is the fact that the optimization of $\theta_{dk}$ involves a numerical algorithm (rather
The main technical difficulty that we faced with the implementation of the model was the optimization of $\theta_{dk}$ during the first CM-step, given by (14). The fact that $\theta_{dk}$ appears both in the LDA and the regression part of the equation forced us to apply numerical algorithms for constrained convex optimization problems, as described in Appendices I and II. This numerical optimization can take a long time if not performed efficiently, so we had to implement an ad-hoc version of these algorithms to accelerate the process as much as possible. Nonetheless, the first CM-step is still responsible for most of the time taken by the optimization of the model (for comparison, notice that in standard LDA the E-step takes longer than the M-step). Training time in TSITM is about an order of magnitude higher than in LDA because of this issue.

It is important to note that choosing the right set of documents for the analysis of a particular time series typically requires domain knowledge. For example, in [40] they use the output of an LDA topic decomposition applied to a financial news dataset as features of a naïve Bayes classifier in order to predict the ups and downs of asset volatilities and close prices. They found a significant signal in the case of volatilities, while their prediction for close prices changes didn’t do better than random choice. Their work is close in spirit to our research, but they only try to classify ups/downs and there is no coupling between the time series and the topics, in the sense that topics are trained alone.

When dealing with a new dataset, selection of hyperparameters is a crucial step. As described in section VI-C, we used Bayesian Optimization for this task. For the initial setting of the optimization space of the topic hyperparameters, we suggest using the range $[0, 1]$ for $\eta$ and $[10^{-5}, 50]$ for $\alpha$, to account for the different types of smoothing. For the regression parameters, we allow $\rho$ to vary in the range $[0, 1]$ ($\rho = 0$ would correspond to a Ridge penalty, while $\rho = 1$ would be Lasso) and $\lambda$ to vary between $[0, 1]$. For the coupling between topic modeling and linear regression, namely $\tau$, we restricted ourselves to the range $[0, 5]$. A better coupling may be achieved if greater values of $\tau$ were allowed but, as explained in VIII-B, training time would scale up.

With respect to the number of topics, $K$, for a new dataset, we suggest fixing it before applying the Bayesian Optimization for the rest of hyperparameters. We do so by training several LDA models with different $K$ and then calculating the reconstruction error over a test set for each of them. We define this error as the Kullback-Leibler divergence between the observed documents and the “reconstructed” documents (i.e. $\sum_{k=1}^{K} \theta_{dk} \hat{\beta}_{kw}$). The rate of improvement on this error as more topics are added is then compared with the one that would be obtained on a noisy synthetic dataset, and an optimal $K$ companies for which relevant signals were discovered using a corpus made of economic news; however, we also found other companies for which no clear correlations with this type of news were found, yielding a MSE compatible with noise. This is natural, since not all companies are expected to be impacted by the same type of news, but it highlights the fact that more elaborated filters may have to be used if one wants to perform a fine-grained analysis of a particular entity.

Choosing the appropriate time series also requires domain knowledge. For example, in [40] they use the output of an LDA topic decomposition applied to a financial news dataset as features of a naïve Bayes classifier in order to predict the ups and downs of asset volatilities and close prices.

| Table 10. Hypothesis testing for results of Tables 2 and 4. |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|
| **Dataset**                     | **p-value for TSTIM’s MSE with respect to…** | **p-value for TSTIM’s MAE with respect to…** | **p-value for TSTIM’s $R^2_{M}$ with respect to…** |
|                                | Baseline | sLDA Approach 1 | sLDA Approach 2 | Baseline | sLDA Approach 1 | sLDA Approach 2 | Baseline | sLDA Approach 1 | sLDA Approach 2 |
| QT1 Politics                   | 0.025    | 0.022           | 0.005           | 0.231    | 0.059           | 0.123           | 0.063    | 0.049           | 0.040           |
| QT1 Econ.                      | 0.020    | 0.219           | 0.003           | 0.138    | 0.191           | 0.035           | 0.022    | 0.238           | 0.017           |
| QT2 Econ.                      | 0.016    | 0.181           | 0.308           | 0.165    | 0.173           | 0.400           | 0.021    | 0.119           | 0.301           |
| QT3 Econ.                      | 0.034    | 0.026           | 0.019           | 0.003    | 0.028           | 0.003           | 0.068    | 0.073           | 0.019           |
| QT4 Econ.                      | 0.123    | 0.070           | 0.025           | 0.099    | 0.134           | 0.037           | 0.083    | 0.056           | 0.031           |
| **Total**                      | $2 \times 10^{-4}$ | $6 \times 10^{-4}$ | $9 \times 10^{-5}$ | $5 \times 10^{-4}$ | $7 \times 10^{-4}$ | $4 \times 10^{-4}$ | $9 \times 10^{-5}$ | $8 \times 10^{-4}$ | $5 \times 10^{-5}$ |
is determined. This optimal $K$ can also be employed for any other method used as baseline (such as sLDA).

D. OPEN LINES OF RESEARCH

An implicit assumption was introduced in our model: the publication of documents was expected to have an immediate impact on the time series, i.e., documents published at a day $t$ were used to predict the value of the time series at that same day, $y_t$. This may not be reasonable in some real-case scenarios, where there might exist a delay between the publication of news and its impact on the series, and such impact might be prolonged for more than one day after its initial publication. Generalizing the TSITM model in order to handle these cases is a task that remains open for future research.

Another line of research that was left open is the possibility of using an alternative topic model (instead of the standard LDA) as the foundation of TSITM. For example, the Biterm Topic Model [41] might be more appropriate than LDA if microblogs (which are common in social media) were used as a text source instead of news; and for certain domains with large and heavy-tailed vocabularies, TSITM might benefit from using a neural topic model with pre-trained word-embeddings [42] or BERT-based representations [43]. However, to the best of our knowledge, topic models based on transformers have not yet been successfully applied for time series prediction. Transformers alone could be used for regression tasks (see, for example, [44]), but then one would lose the interpretability provided by topics, which is an essential feature of our work. On the other hand, using traditional topic models for regression tasks is still a very active field of research, as demonstrated by the number of articles published in recent years ( [15], [19], [20], just to name a few).

Lastly, notice that we are using Latent Dirichlet Allocation instead of a time-sensitive topic model such as the Dynamic Topic Model [18] because we are not expecting to work with very long periods of time. For example, most of the numerical time series that are interesting from a financial point of view are very sensitive to structural changes, and should only be applied to short periods between these changes. Therefore, we have not taken into account the semantic evolution of topics across time.

IX. CONCLUSION

In this paper, we have proposed a novel approach to jointly model a time-series of numerical data and a sequence of time-ordered texts. The likelihood function for the TSITM model was defined as a combination of the LDA topic model with an elastic net linear regression term, and the ECM algorithm was used to numerically find its local minima. The appearance of $\theta_{dk}$ in both terms of the likelihood introduced a coupling between topic modeling and linear regression, which allowed us to obtain topics specifically customized to fit a particular time series. The coupling hyperparameter, $\tau$, was used for weighting the relative importance of the regression term with respect to the topic model.

A strategy to empirically determine the appropriate values for the hyperparameters was discussed and successful applications of the algorithm were demonstrated in two different domains: with the US president’s disapproval ratings and political news, we focused on one specific period to illustrate how the obtained results were easily interpretable and consistent with our expectations; with stock market data and economic news, we proved that it is possible to consistently find reliable signals across different corpora and time series. Both the intrinsic baseline and the two approaches for sLDA used as state of the art baselines were significantly outperformed by TSITM in MSE, MAE and $R^2_{OS}$, according to our hypothesis tests.

The results of these experiments proved that the three research goals proposed in section III were fulfilled by the TSITM model. The results also illustrated that TSITM could be used to quantitatively determine which topics damage the reputation of a given entity.

APPENDIX I. PROXIMAL GRADIENT ALGORITHM FOR THE OPTIMIZATION OF $\theta_{dk}$

Proximal gradient algorithms [25] are a standard tool for solving constrained convex minimization problems such as the one that we have encountered in (14). In its most general form, proximal gradient algorithms can be used to optimize any function $F(x)$ such that

$$F(x) = f(x) + g(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is closed proper convex and differentiable and $g : \mathbb{R}^n \to \mathbb{R} \cup \infty$ is closed proper convex (constraints on $x$ are typically encoded on $g$). The proximal gradient method for optimizing (25) consists on applying iteratively steps of the form

$$x^{(l+1)} = \text{prox}_{\lambda g} \left( x^{(l)} - \lambda^{(l)} \nabla f \left( x^{(l)} \right) \right)$$

where $\lambda^{(l)}$ is a step size chosen in each step and the proximal operator, $\text{prox}_{\lambda g} (v) : \mathbb{R}^n \to \mathbb{R}^n$, is defined as

$$\text{prox}_{\lambda g} (v) := \arg \min_{x} \left\{ g(x) + \frac{1}{2\lambda} \|x - v\|^2 \right\}.$$  

The proximal operator $\text{prox}_{\lambda g} (v)$ can be seen as a compromise between minimizing $g$ and being near to $v$. The step size is typically chosen with a line search such as the backtracking rule proposed in [45], which we reproduce in Algorithm 2 with the notation used in [25]. For the stopping condition in Algorithm 2, the following upper bound of $f$, $\hat{f}_k$, was introduced:

$$\hat{f}_k (x, y) = f(y) + \nabla f(y)^T (x - y) + \left( \frac{1}{2\tau} \|x - y\|^2 \right).$$

To see why the proximal gradient algorithm is suitable for our purposes, let us rewrite the objective function in (14) (which we precede here by a minus sign, since the proximal
gradient algorithm is typically framed as a minimization technique, rather than a maximization) as
\[ \mathcal{L}(\theta) = f(\theta) + g(\theta) \] (29)
with
\[ f(\theta) = \frac{1}{2} \sum_{i=1}^{T} \tau' \left( \hat{y}_i - \mu_0 - \sum_{k=1}^{K} \mu_k \theta_{dk} \right)^2 + \sum_{k=1}^{K} (a_k \sigma_k^2 + b_k \sigma_k), \] (30)
\[ g(\theta) = -\sum_{d=1}^{D} \sum_{k=1}^{K} z_{dk}' \log \theta_{dk} + \delta_S(\theta), \] (31)
where we have defined, for simplicity,
\[ z_{dk}' = \sum_{w=1}^{V} n_{dw'} (z_{dwk}) + \alpha_k, \] (32)
\[ \tau' = \frac{N_c \tau}{1000T}, \] (33)
\[ a_k = \frac{\tau' T \lambda_k (1 - \rho)}{2} \mu_k^2, \] (34)
\[ b_k = \tau' T \lambda_k \rho |\mu_k|, \] (35)
and we have encoded the constraints of (14) into (31) through the function \( \delta_S(\theta) \) which is zero in the region of allowed parameters, \( S \), and infinite elsewhere, that is,
\[ \delta_S(\theta) = \begin{cases} \infty, & \theta \notin S \\ 0, & \theta \in S \end{cases}, \] (36)
\[ S := \left\{ \theta_{dk} \mid \theta_{dk} \geq 0, \sum_{k=1}^{K} \theta_{dk} = 1 \right\}. \] (37)
Since both (30) and (31) are closed proper convex and (30) is differentiable, then (29) has the same form as (25) and the proximal gradient algorithm can be applied to find its global optimum. Notice that, for each \( n^{th} \) step of the ECM algorithm, the proximal operator will be iteratively applied until convergence, so we will denote by \( l \) each step of the proximal gradient algorithm to avoid notation ambiguities. Also notice that, to simplify, we have written \( \theta_{dk}, z_{dwk} \) and \( \mu_k \) to refer to \( \theta_{dk}^{(n+1)}, z_{dwk}^{(n)} \) and \( \mu_k^{(n)} \).

Each step of the proximal gradient algorithm is given by (26), which for \( \theta_{dk} \) at iteration \( l + 1 \) takes the form of a vector equation on the combined index \( dk \):
\[ \theta^{(l+1)} = \text{prox}_{\lambda^{(l)}} \left( \theta^{(l)} - \lambda^{(l)} \nabla f(\theta^{(l)}) \right), \] (38)
with the derivative in (38) being immediate to compute, yielding
\[ \frac{\partial f(\theta^{(l)})}{\partial \theta_{dk}} = -\mu_k \tau' (\hat{y}_{id} - \mu_0 - \mu_k \theta_{dk}) + \frac{2}{T} (\theta_{dk} - \bar{\theta}_k) \left( a_k + \frac{1}{2s_k^2} \bar{b}_k \right). \] (39)
By the definition of proximal operator (27), the step (38) can be written as
\[ \theta^{(l+1)} = \arg\min_{\theta_{dk}} \left\{ -\sum_{d=1}^{D} \sum_{k=1}^{K} z_{dk}' \log \theta_{dk} + \frac{1}{2\lambda^{(l)}} \sum_{d=1}^{D} \sum_{k=1}^{K} (\theta_{dk} - x_{dk})^2 \right\}, \] (40)
where we have defined, to avoid clutter,
\[ x_{dk} = \theta_{dk}^{(l)} - \lambda^{(l)} \frac{\partial f(\theta^{(l)})}{\partial \theta_{dk}}. \] (41)
To satisfy the constraints for \( \theta_{dk} \), we introduce a set of \( D \) Lagrange multipliers, \( \mu_d \), into (40),
\[ \theta^{(l+1)} = \arg\min_{\theta_{dk}} \left\{ -\sum_{d=1}^{D} \sum_{k=1}^{K} z_{dk}' \log \theta_{dk} + \frac{1}{2\lambda^{(l)}} \sum_{d=1}^{D} \sum_{k=1}^{K} (\theta_{dk} - x_{dk})^2 \right\} + \sum_{d=1}^{D} \mu_d \left( \sum_{k=1}^{K} \theta_{dk} - 1 \right). \] (42)
By taking derivatives of (42) with respect to \( \theta_{dk} \) and setting them equal to zero, we finally obtain
\[ \theta_{dk}^{(l+1)} = \frac{\lambda^{(l)}}{2} \left( a_{dk} + \sqrt{a_{dk}^2 + 4x_{dk}^2} \right), \] (43)
where we have defined, for simplicity,
\[ a_{dk} = -\left( \mu_d - \frac{x_{dk}}{\lambda^{(l)}} \right). \] (44)
Equation (43) is the final form of each of the steps that must be taken in the proximal gradient algorithm. However, there are two parameters that remain to be determined in (43): the set of Lagrange multipliers \( \mu_d \) and the step size \( \lambda^{(l)} \).
To determine the value of the Lagrange multipliers, recall that solution (43) must satisfy the condition \( \sum_{k=1}^{K} \theta_{dk} = 1 \), which implies that
\[ 1 = \frac{1}{2} \sum_{k=1}^{K} (x_{dk} - x_d) + \sqrt{(x_{dk} - x_d)^2 + \mu_{dk}^2} \] (45)
where we introduced the following definitions
\begin{align}
    x_d &= \lambda^{(i)} \mu_d, \\
    u_{dk}^2 &= 4z_{dk} \lambda^{(i)}.
\end{align}

From (45), \( \mu_d \) can be numerically determined by using the Newton’s method. We refer the interested to Appendix II for a detailed description of this solution.

The step size \( \lambda^{(i)} \) is determined by following the backtracking stepsize rule described in Algorithm 2, where the stopping condition in this case takes the form
\begin{equation}
    f(\theta_{dk}) = f \left( \theta_{dk}^{(i)} \right) + \left( \theta_{dk} - \theta_{dk}^{(i)} \right) \frac{\partial f(\theta_{dk})}{\partial \theta_{dk}} + \frac{1}{2\lambda} \left( \theta_{dk} - \theta_{dk}^{(i)} \right)^2.
\end{equation}

We have encountered that, for \( \beta = 1/2 \), the stopping condition of the backtracking rule is typically reached after a very small number of iterations.

To summarize, Algorithm 3 contains the key results obtained in this appendix. To initialize \( \theta_{dk}^{(i)} \), the value of \( \theta_{dk} \) used in the previous step of the E-CM algorithm can be employed; for \( \lambda \), it is common to set \( \lambda^{(0)} = 1 \).

**Algorithm 3** Proximal Gradient Algorithm for the Optimization of \( \theta_{dk} \)

1: Initialize parameters \( \theta_{dk}^{(1)}, \lambda^{(0)} \) and set \( \beta \in (0, 1) \).
2: repeat
3: \quad Let \( \lambda = \lambda^{(i-1)} \).
4: \quad Set \( \mu_d \) numerically with Algorithm 4.
5: \quad Set \( z = (43) \) with \( \lambda^{(i)} = \lambda \).
6: \quad while Equation (48) is True do
7: \quad \quad Update \( \lambda = \beta \lambda \).
8: \quad \quad Set \( \mu_d \) numerically with Algorithm 4.
9: \quad \quad Set \( z = (43) \) with \( \lambda^{(i)} = \lambda \).
10: \quad end while
11: \quad Set \( \theta_{dk}^{(i+1)} = z \) and \( \lambda^{(i)} = \lambda \).
12: until \( \theta_{dk} \) converged.

**APPENDIX II. NEWTON’S METHOD FOR FINDING THE ROOTS OF THE LAGRANGE MULTIPLIERS EQUATION IN THE PROXIMAL GRADIENT ALGORITHM**

The Newton’s method is one of the most used techniques for finding the roots of a differentiable function, \( f : (a, b) \rightarrow \mathbb{R} \), starting from an initial approximation \( x^{(0)} \). It is based on iteratively finding the intersection of the x-axis and the tangent line of \( f \) at the current iteration guess \( x^{(m)} \), that is,
\begin{equation}
    x^{(m+1)} = x^{(m)} - f \left( x^{(m)} \right) / f' \left( x^{(m)} \right)
\end{equation}

where \( f' \) is the function’s derivative.

Equation (45) is a perfect candidate for being solved with the Newton’s method if we rewrite it as a root finding problem for
\begin{equation}
    f(x_d) = 1 - \frac{1}{2} \sum_{k=1}^{K} \left( (x_{dk} - x_d) + \sqrt{(x_{dk} - x_d)^2 + u_{dk}^2} \right).
\end{equation}

The initial point \( x_d^{(0)} \) used for the first step of the algorithm can be estimated by noting that \( x_d \gg x_{dk}, u_{dk} \), so that
\begin{equation}
    1 = \frac{1}{2} \sum_{k} \left( (x_{dk} - x_d) + \sqrt{(x_{dk} - x_d)^2 + u_{dk}^2} \right)
    \approx \frac{1}{2} \sum_{k} \frac{1}{2} \left( x_{dk} - x_d \right) \approx \frac{1}{4} x_d \sum_{k} u_{dk}^2.
\end{equation}

Therefore, our proposed initial point estimation is
\begin{equation}
    x_{d}^{(0)} \approx \frac{1}{4} \sum_{k} u_{dk}^2.
\end{equation}

We summarize the results of this appendix in Algorithm 4. Once \( x_d \) has been determined with the Newton’s method, \( \mu_d \) is immediate to obtain using definition (46).

**Algorithm 4** Newton’s Method for Finding the Roots of (45)

1: Initialize \( x_{d}^{(0)} \) using (52).
2: repeat
3: \quad Set \( x_{d}^{(m+1)} \) using (49) for function (50).
4: until \( x_d \) converged.

**ACKNOWLEDGMENT**

The authors would like to thank Management Solutions for useful comments, feedback, and discussions, and also would like to thank Daniel Levy for his technical contributions.

**REFERENCES**

[1] F. Z. Xing, E. Cambria, and R. E. Welsch, “Natural language based financial forecasting: A survey,” *Artif. Intell. Rev.*, vol. 50, no. 1, pp. 49–73, Jun. 2018.
[2] Z. Lin, A. Madotto, G. I. Winata, Z. Liu, Y. Xu, C. Gao, and P. Fung, “Learning to learn sales prediction with social media sentiment,” in *Proc. 1st Workshop Financial Technol. Natural Lang. Process.*, 2019, pp. 47–53.
[3] V. H. Larsen, L. A. Thorsrud, and J. Zhulanova, “News-driven inflation expectations and information rigidities,” *J. Monetary Econ.*, vol. 117, pp. 507–520, Jan. 2021.
[4] H. Nobanee, M. Alhajjar, G. Abushairah, and S. Al Harbi, “Reputational risk and sustainability: A bibliometric analysis of relevant literature,” *Risks*, vol. 9, no. 7, p. 134, Jul. 2021.
[5] J. Chang, W. Tu, C. Yu, and C. Qin, “Assessing dynamic qualities of investor sentiments for stock recommendation,” *Inf. Process. Manag.*, vol. 58, no. 2, Mar. 2021, Art. no. 102452.
[6] Y. Qian, Z. Li, and H. Yuan, “On exploring the impact of users’ bullish-bearish tendencies in online community on the stock market,” *Inf. Process. Manag.*, vol. 57, no. 5, Sep. 2020, Art. no. 102209.
[7] C. C. Aggarwal, *Machine Learning for Text*, 1st ed. Cham, Switzerland: Springer, 2018.
[8] S. Kogan, D. Levin, B. R. Routledge, J. S. Sagi, and N. A. Smith, “Predicting risk from financial reports with regression,” in *Proc. Hum. Lang. Technol.*, Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics (NAACL), Boulder, CO, USA: Association for Computational Linguistics, Jun. 2009, pp. 272–280.
D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

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 Appl.*, vol. 78, no. 11, pp. 15169–15211, Jun. 2019.

T. Hofmann, “Probabilistic latent semantic analysis,” in *Proc. 15th Conf. Uncertainty Artif. Intell.* Stockholm, Sweden: Morgan Kaufmann, 1999, pp. 289–296.

S. Sivanandam, A. S. Kumar, R. Pradeep, and R. Sridhar, “Analysing research trends using topic modelling and trend prediction,” in *Soft Computing and Signal Processing*, V. S. Reddy, V. K. Prasad, J. Wang, and K. T. V. Reddy, Eds. Singapore: Springer, 2021, pp. 157–166.

H. D. Kim, M. Castellanos, M. Hsu, C. X. Zhai, T. Riez, and D. Diermeier, “Mining causal topics in text data: Iterative topic modeling with time series feedback,” in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, San Francisco, CA, USA: Association for Computing Machinery, 2013, pp. 885–890.

D. M. Blei and J. D. McAuliffe, “Supervised topic models,” in *Proc. 20th Int. Conf. Neural Inf. Process. Syst.*, Vancouver, BC, Canada: Curran Associates, 2007, pp. 121–128.

P. Glassereman, K. Krstovski, P. Laliberte, and H. Mamaysky, “Choosing news topics to explain stock market returns,” in *Proc. 1st ACM Int. Conf. Al Finance*, Oct. 2020, pp. 1–5.

J. Staines and D. Barber, “Topic factor models: Uncovering thematic structure in equity market data,” *Intell. Data Anal.*, vol. 19, no. S1, pp. S69–S85, Oct. 2015.

S. Park, W. Lee, and I.-C. Moon, “Associative topic models with numerical time series,” *Inf. Process. Manage.*, vol. 51, no. 5, pp. 737–755, Sep. 2015.

D. M. Blei and J. D. Lafferty, “Dynamic topic models,” in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, Pittsburgh, PA, USA: Association for Computing Machinery, 2006, pp. 113–120.

N. Kanunguokassem and T. Leelanupap, “Financial latent Dirichlet allocation (FinLDA): Feature extraction in text and data mining for financial time series prediction,” *IEEE Access*, vol. 7, pp. 71645–71664, 2019.

V. H. Larsen and L. A. Thorssrud, “The value of news for economic developments,” *J. Econometrics*, vol. 210, no. 1, pp. 205–218, May 2019.

H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *J. Roy. Statist. Soc., B, Stat. Methodol.*, vol. 67, no. 2, pp. 301–320, Mar. 2005.

C. Cunne, Y. D. Zhuo, H. S. Moat, and T. Preis, “Quantifying the diversity of news and stock market moves,” *J. Netw. Theory Finance*, vol. 3, no. 1, pp. 1–20, Mar. 2017.

X. Wang and A. McCallum, “Topics over time: A non-Markov continuous-time model of topical trends,” in *12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Philadelphia, PA, USA: Association for Computing Machinery, 2006, pp. 424–433.

G. J. McLachlan and T. Krishman, *The EM Algorithm and Extensions*, 2nd ed. Hoboken, NJ, USA: Wiley, 2007.

N. Parikh and S. Boyd, “Proximal algorithms,” *Found. Trends Optim.*, vol. 1, no. 3, pp. 123–231, Jan. 2013.

J. Friedman, T. Hastie, and R. Tibshirani, “Regularization paths for generalized linear models via coordinate descent,” *J. Statist. Softw.*, vol. 33, no. 1, pp. 1–22, 2010.

S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, “An interior-point method for large-scale l1-regularized least squares,” *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, “Stanza: A Python natural language processing toolkit for many human languages,” in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 101–108.

R. Rehůřek and P. Sojka, “Software framework for topic modelling with large corpora,” in *Proc. LREC Workshop New Challenges NLP Frameworks*, Valletta, Malta: Univ. of Malta, 2010, pp. 46–50.

E. Amigó, J. Gonzalez, J. Artilles, and F. Verdejo, “A comparison of extrinsic clustering evaluation metrics based on formal constraints,” *Inf. Retr.*, vol. 12, no. 4, pp. 461–486, 2009.

S. Snoek, H. Larochelle, and R. Adams, “Practical Bayesian optimization of machine learning algorithms,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25. Red Hook, NY, USA: Curran Associates, 2012, pp. 2951–2959.

C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

T. Head, M. Kumar, H. Nahrstaedt, G. Lopupe, and I. Shcherbatiy, “scikit-optimize:scikit-optimize,” Version v0.8.1, Zenodo, Sep. 2020. [Online]. Available: https://zenodo.org/record/4014775#.Yw491lwuaUk, doi: 10.5281/zenodo.4014775.

C. C. Aggarwal, *Data Mining*, 1st ed. Cham, Switzerland: Springer, 2015.

X. Zhang, S. Qu, J. Huang, B. Fang, and P. Yu, “Stock market prediction via multi-source multiple instance learning,” *IEEE Access*, vol. 6, pp. 50720–50728, 2018.

X. Li, P. Wu, and W. Wang, “Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong,” *Inf. Process. Manage.*, vol. 57, no. 5, Sep. 2020, Art. no. 102122.

X. Ji, J. Wang, and Z. Yan, “A stock price prediction model based on deep learning technology,” *Int. J. Crowd Sci.*, vol. 5, no. 1, pp. 55–72, Apr. 2021.

A. Giachanou, J. Gonzalez, and F. Crestani, “Propagating sentiment signals for estimating reputation polarity,” *Inf. Process. Manag.*, vol. 56, no. 6, Nov. 2019, Art. no. 102079.

A. Atkins, M. Niranjan, and E. Gerdng, “Financial news predicts stock market volatility better than close price,” *J. Finance Data Sci.*, vol. 4, no. 2, pp. 120–137, Jun. 2018.

X. Cheng, X. Yan, Y. Lan, and J. Guo, “BTM: Topic modeling over short texts,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 12, pp. 2928–2941, Dec. 2014.

A. B. Dieng, F. J. R. Ruiz, and D. M. Blei, “Topic modeling in embedding spaces,” *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 439–453, Dec. 2020.

F. Bianchi, S. Terragni, and D. Hovy, “Pre-training is a hot topic: Contextualized document embeddings improve topic coherence,” in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Proc.*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 1–8.

M. Li, W. Li, F. Wang, X. Jia, and G. Rui, “Applying BERT to analyze investor sentiment in stock market,” *Neural Comput. Appl.*, vol. 33, no. 10, pp. 4663–4676, May 2021.

A. Beck and M. Teboulle, “Gradient-based algorithms with applications to signal-recovery problems,” in *Convex Optimization in Signal Processing and Communications*, D. P. Palomar and Y. E. Eldar, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2009, pp. 42–88.
Julio Gonzalo is currently a Full Professor with the Universidad Nacional de Educación a Distancia (UNED), Spain, where he leads the Research Group in natural language processing and information retrieval. His publications cover many topics at the intersection of information retrieval and natural language processing. His current research interests include evaluation metrics and methodologies, semantic textual similarity, online reputation monitoring, and information access technologies for social media. He has co-chaired evaluation activities, such as RepLab (for Reputation Monitoring tasks), Web People Search (WePS), and iCLEF (interactive Cross-Language search and Question Answering). In this context, his research made a special emphasis on the formal assessment of evaluation metrics, which led to a Google Faculty Research Award (with Stefano Mizzaro and Enrique Amigó), in 2012. He has recently been appointed as the General Co-Chair for the ACM SIGIR 2022 Conference.

Marcos Galletero received the B.Sc. degree in physics from the Universidad Complutense de Madrid (UCM), in 2017, and the M.S. degree in data science & engineering from the Universidad Nacional de Educación a Distancia (UNED), in 2021. His background lies in reservoir computing and control in dynamic systems. His research interests include opinion mining in social networks and complex systems.

Ivar Zapata received the Ph.D. degree in physics from the Universidad Autónoma de Madrid (UAM), in 1997. Since 2017, he has been involved in the creation and leading the Data Science Department, mRHouston Tech Solutions. The department’s focus is on latent analysis of web articles, mainly topic classification, sentiment extraction, and the financial measurement of reputation in listed companies. His research interests include superconductivity, Bose-Einstein condensation, graphene, and stochastic processes in condensed matter physics.

* * *