Essays in Risk Management with Machine Learning

THIS IS A TEMPORARY TITLE PAGE
It will be replaced for the final print by a version
provided by the registrar's office.

Thèse n. XXXX 2022
présentée le XXXX
Collège du Management de la Technologie
Chaire Swissquote en finance quantitative
programme doctoral en finance
pour l'obtention du grade de Docteur ès Sciences
par

Marc-Aurèle Antoine DIVERNOIS

Proposition du jury:

Prof Name Surname, président du jury
Prof Name Surname, directeur de thèse
Prof Name Surname, rapporteur
Prof Name Surname, rapporteur
Prof Name Surname, rapporteur

Lausanne, EPFL, 2022

Acknowledgements

I wish to express my deepest gratitude to my supervisor, Damir Filipović, for his availability and continuous support during these five years. This thesis greatly benefited from his countless valuable comments and flow of smart ideas. He guided me and I learned a lot from him about doing research. For this, I am truly grateful.
Special thanks go to professor Michael Rockinger who convinced me to pursue a doctorate degree. He took me under his wing since my Bachelor studies and helped me becoming the researcher I am today. This thesis would not exist without him.
I would like to acknowledge the rest of my thesis committee, the professors Pierre Collin-Dufresne, Andreas Fuster and Simon Scheidegger for their advice and comments.
I am also thankful to the professors Elena Perazzi and Damien Challet. I was their TA for several years and they helped me become a better teacher.
During these years I had the chance to work with extremely nice and intelligent people. I am thankful to my colleagues and friends Antoine Didisheim and Coralie Jaunin for the many chess games and 'bridge tours' around the campus.
I would like to thank my family: my parents Jacques and Sonia, my sister Isabelle and her husband Edward, my nephew Dylan and my fiancée Cathy for their unconditional love and for bearing all my endless monologues about my latest models. I dedicate this thesis to them.

Lausanne, June 3, 2022 M.-A. D.

Abstract

This thesis consists of three applications of machine learning techniques to risk management.

The first chapter proposes a deep learning approach to estimate physical forward default intensities of companies. Default probabilities are computed using artificial neural networks to estimate the intensities of the inhomogeneous Poisson processes governing default process. The major contribution to previous literature is to allow the estimation of non-linear forward intensities by using neural networks instead of classical maximum likelihood estimation. The model specification allows an easy replication of previous literature using linear assumption and shows the improvement that can be achieved.

The second chapter, titled 'Causal Networks with Neural Networks' is a co-authored work with Damir Filipović (SFI & EPFL), Negar Kiyavash (EPFL) and Jalal Etesami (EPFL). We develop a data-driven framework to identify the interconnections between firms using an information-theoretic measure. This measure generalizes Granger causality and is capable of detecting nonlinear relationships within a network. Moreover, we develop an algorithm using recurrent neural networks and Granger causality to identify the interconnections of high-dimensional nonlinear systems. The outcome of this algorithm is the causal graph encoding the interconnections among the firms. These causal graphs can be used as preliminary feature selection for another predictive model or for systemic risk management. We evaluate the performance of our algorithm using both synthetic linear and nonlinear experiments and apply it to the daily stock returns of U.S. listed firms and infer their interconnections from 1990 to 2020.

The third chapter, titled 'StockTwits Classified Sentiment and Stock Returns' is a co-authored work with Damir Filipović (SFI & EPFL). We scrape 90 million messages from StockTwits over 10 years and classify them into bullish, bearish or neutral classes to create firm-individual sentiment polarity time-series. Polarity is positively associated with contemporaneous stock returns. On average, polarity is not able to predict next-day stock returns but when we focus on specific events (defined as sudden peak of message volume), polarity has predictive power on abnormal returns.

Keywords: Risk management, machine learning, neural networks, asset pricing, big data, alternative data.

Résumé

Cette thèse consiste en trois applications de techniques d'apprentissage automatique à la gestion des risques.

Le premier chapitre propose une approche d'apprentissage automatique profond pour estimer les probabilités physiques de défaut des entreprises. Les probabilités de défaut sont calculées en utilisant des réseaux de neurones artificiels pour estimer les intensités des processus de Poisson non-homogènes qui gouvernent les processus stochastiques de défaut. La contribution majeure apportée à la littérature existante est de rendre possible l'estimation non-linéaire des intensités en utilisant les réseaux de neurones artificiels au lieu de la classique estimation du maximum de vraisemblance. Les propriétés du modèle autorisent une réplication aisée de la littérature existante (qui utilise l'hypothèse de linéarité de l'intensité) et montre l'amélioration qui peut être obtenue.

Le deuxième chapitre, intitulé 'Causal Networks wih Neural Networks' est un travail conjoint avec Damir Filipović (SFI & EPFL), Negar Kiyavash (EPFL) et Jalal Etesami (EPFL). Nous développons un modèle axé sur les données et reposant sur une mesure d'information théorique pour identifier les interconnexions entre les entreprises. Cette mesure utilise la causalité de Granger et est capable de détecter des relations non-linéaires à l'intérieur d'un réseau. De plus, nous développons un algorithme qui utilise les réseaux de neurones récurrents ainsi que la causalité de Granger pour identifier les interconnexions dans les systèmes non-linéaires à haute dimension. Le résultat de cet algorithme est le diagramme causal encodant les interconnexions des entreprises. Ces diagrammes causaux peuvent être utilisés comme modèles préliminaires de sélection de variables d'un autre modèle de prédiction ou pour la gestion de risque systémique. Nous évaluons en premier lieu la performance de notre algorithme en utilisant des expériences synthétiques linéaires et non-linéaires puis nous appliquons notre modèle aux rendements journaliers d'actions américaines cotées pour en déduire leurs interconnexions de 1990 à 2020.

Le troisième chapitre, intitulé 'StockTwits Classified Sentiment and Stock Returns' est un travail conjoint avec Damir Filipović (SFI & EPFL). Nous récupérons 90 millions de messages (couvrant 10 ans) provenant de StockTwits et les classifions dans la classe haussière, baissière ou neutre pour créer des séries temporelles de polarité propres à chaque entreprise. La polarité est associée positivement aux rendements d'actions contemporains. En

moyenne, la polarité n'est pas capable de prédire les rendements du jour suivant mais lorsque nous nous focalisons sur des évenements spécifiques (définis par une augmentation soudaine du volume de messages), la polarité a de la puissance prédictive sur les rendements anormaux.

Mots-clés: Gestion des risques, apprentissage automatique, réseaux de neurones artificiels, évaluation d'actifs, mégadonnées, données alternatives.

Contents

List of Figures

List of Tables

Introduction

Improved computational power, the rise of Big Data and recent developments in machine learning have created new areas of research in finance. This thesis consists of three machine learning applications to risk management. The first two chapters use neural networks to mitigate default and systemic risk and the last chapter is a financial sentiment analysis using natural language processing.

The first chapter builds on the works from Duffie et al. (2007) and Duan et al. (2012). Both models use a doubly stochastic argument to derive multi-period default probabilities. In particular, they estimate the intensities of two Poisson processes, one governing default and the other governing other exits. Duffie et al. (2007) generates future random values for the covariates using a VAR process while Duan et al. (2012) relaxes this assumption and use forward intensities. The latter specifies the intensities as a linear function of state variables and uses maximum likelihood to estimate the parameters. The first chapter of this thesis extends existing literature by removing the assumption of linear intensities and uses artificial neural networks to estimate the intensities of the Poisson processes. Neural networks are well-suited in this framework because they allow an easy replication of the linear formulation in Duan et al. (2012). Increasing the network's width and depth allows for non-linearities and out-of-sample Lorenz Curves show that the neural network's approach outperforms the linear assumption for every horizon. Finally, I show what are the most important predictors of default in the short and longer term. Interdepencies in a network are at the heart of systemic risk. The second chapter - connected to the first one as another application of neural networks to risk management - employs Granger causality to identify interconnections among a set of institutions. We build an information measure known as directed information (DI) capable of capturing causal relationships in both linear and non-linear systems. The output of this approach is a directed graph that visualizes the interconnections among a set of time series. Computing DI has both computational and sample complexity which makes it not suitable for inferring the causal structure of large networks. To overcome this problem, we develop a novel approach based on recurrent neural networks that reduces complexity of evaluating DI in high-dimensional settings. We show that our approach performs well both in a linear and non-linear simulated enviornments, then apply it to infer the causal relationships among US firms from 1990 to 2020.

The last chapter uses natural language processing to assess the predictive power of social media on stock returns.We scrape 90 millions messages from Stocktwits - a microblogging platform similar to Twitter but designed for finance professionals. One of the challenge in this context is to create a classifier that understands the vocabulary of the message posted by the users. After preprocessing steps, we use TFIDF vectorization to compute the importance of each word in the message. This transforms the messages from words to a vector of numbers which is now in the same dimension as the vocabulary. Then, we build two adversarial logistic regressions using the TFIDF vectors as features and the user-labels as targets. The first model predicts positive or not positive, the second model predicts negative or not negative. When the models agree, we get our label and when they disagree, we treat the tweet as neutral. This procedure allows us to create an artificial neutral class that absorbs all the tweets that do not convey financial information. Finally, with daily intervals, we aggregate tweets predicted sentiments per ticker to compute daily polarity time-series. We then use the daily volume of messages on a given firm to identify sudden peak of activity, indicating a firm event. Computing cumulative average abnormal return and cumulative abnormal polarity in a 41 days window centered at the identified event, we show that abnormal polarities have significant predictive power on the type of event.

1 A deep learning approach to estimate forward default intensities

1.1 Introduction

The first default prediction models appeared forty years ago with the first generation model presented by Altman (1968). This work led to the so-called Altman Z-score formula which uses accounting data to compute the default probability of a firm in the next two years. However, when used for financial firms, Altman's Z-score formula needs to be used with care because, as I will discuss in this paper, financial firms have to be treated carefully due to their frequent use of off-balance sheet financing. Twenty years later, a second generation of reduced-form models used econometrical tools such as maximum likelihood, probit, and logit regressions. The major drawback of these models is that they do not provide multi-period forecasts. One innovative recent development is the use of doubly stochastic Poisson intensity model combined with multiple logit regression to account for multi-period default probability estimation. This model has been proposed by Duffie et al. (2007) in *Multi-period corporate default prediction with stochastic covariates*. The main contribution over prior work was to exploit the time-series dynamics of the explanatory covariates in order to estimate the likelihood of default over several future periods. Their model employs firm-specific and macroeconomic data to create a Markov state vector X_t in order to compute independent firm default intensities $\lambda(t)$ and other types of exit intensities $\phi(t)$. Duffie et al. (2007) is the first model capable of multi-period default probability estimation using time dynamics of covariates X_t. Applications of Duffie et al. (2007) works are various. We can find them in credit rating by credit rating agencies, banks who want to calculate the minimal amount of capital to be held and other researches analyzing the link between macroeconomic cycles and firm's default probabilities. Covariates used by Duffie et al. (2007) are firm's distance to default, firm's trailing one-year stock return, three-month Treasury bill rate and trailing one-year return on S&P500. Estimating their model on US-listed Industrial firms between 1980 and 2004, they find that distance to default and the current state of the economy have a significant impact on default hazard rates.

The two papers the closest to my paper are the work from Duffie et al. (2007) and Duan et al.

(2012). The first model uses the doubly stochastic argument to derive multi-period default probabilities. To do so, it requires some strong assumptions (i.e Vectorial Autoregressive process) regarding the behavior of the time-series of covariates to generate future random values for the covariates. Obviously, if the process is misspecified, biases are introduced both in the forecasted covariates and in the future default probabilities. Five years later, Duan et al. (2012) shows that we can relax the VAR assumption with the use of forward intensities. This paper explains how we can reduce biases by projecting current event realizations on past data. For convenience, they specify the intensities as a linear function of state variables. I wish to extend the latter by removing the assumption of linear intensities and use an artificial neural network to estimate the intensities of the Poisson processes governing both default and other exits. Many papers on estimating default probabilities with machine learning techniques have already been written. For instance, Altman et al. (2017) test several machine learning models (support vector machines, bagging, boosting, and random forest) to predict bankruptcy one year prior to the event. They report a substantial improvement in prediction accuracy using these machine learning techniques especially when, in addition to the original Altman's Z-score variables, six complementary financial indicators are included. To the best of my knowledge, such types of models do not rely on theoretical foundations and this is where the forward intensity model can contribute to the literature. It is able to provide multi-period predictions and is supported by a solid mathematical and econometrical background.

In Duffie et al. (2007), one of the main assumption is that the covariates governing both default and other exits intensities follow a high-dimensional vector auto-regressive (VAR) process. Using this type of process forces the model to greatly reduce either the number of firms in the sample or the number of state variables explaining firm attributes; if one does not restrict the number of firms or variables in the estimation, the dimension of the model will simply be too high and it will considerably increase computational time. A major step forward made in Duan et al. (2012) was to get rid off the VAR process in order to reduce computational time by using "a new reduced-form approach based on a *forward* intensity model". These forward intensities produce a term structure of bankruptcy probabilities without using any sort of high-dimensional process. Using this method allows the model to incorporate a lot more state variables or individual firms in the sample. Moreover, Duan et al. (2012) state that their model may also improve robustness to misspecification because in a VAR model, estimation of future values are highly sensitive to any biases. On the other side, the forward intensity model approach is a "direct projection of past data on current realizations" which does not involve random estimation of future values.
Regarding covariates used, both Duan et al. (2012) and Duffie et al. (2007) estimate their own Distance-to-Default (hereafter DtD). An important aspect to highlight is that distance to defaults specified in Duffie et al. (2007) differ from those estimated in Duan et al. (2012) since the first one are estimated using the Variance Restriction method (see *Measuring Distance-to-Default for Financial and Non-Financial Firms* (Duan and Wang, 2012)) and the second one using the transformed-data maximum likelihood estimation method to account for financial firms. The variance restriction method is a popular way to implement the Merton (1974)

model but fails to estimate properly the default point for financial firms. Following the KMV assumption (see Crosbie and Bohn (2003)), the default point in this method is specified as short term debt plus one half of long term debt and does not take into account other liabilities. However, it is well-known that financial firms such as banks specify a high portion of their debt as “other liabilities". Hence, to include financial firms in the sample, the default point has to be adjusted to the sum of short term debt, one half of long term debt and a fraction δ of other liabilities. The method used by Duan et al. (2012) is using a maximum likelihood estimation in order to estimate the unknown fraction δ. The Appendix provides additional methodological information on the DtD estimation using the variance restriction method and the maximum likelihood estimation.
To summarize, the forward intensity models require an assumption to link covariates to intensities. Duan et al. (2012) uses a linear assumption and a maximum likelihood to estimate those parameters. This paper contributes to previous literature by using neural networks to relax the linear assumption of forward intensities. Neural networks allow the estimation of highly non-linear functions without specifying the form of the relationships. The organization of the paper is as follows. Section 2 sets up the reduced-form model of default, develops the likelihood function used as loss function later on and describes the neural network approach. Section 3 discusses summary statistics of the dataset used. Section 4 presents the results. Section 5 concludes. Appendices at the end of the paper contains several proofs and more details on the Distance-to-Default estimation.

1.2 Methodology

1.2.1 Default model

The model adds to the literature of reduced-form models of default for multiperiod corporate prediction using the doubly stochastic formulation as in Duffie et al. (2007) (DSW henceforth) and Duan et al. (2012) (Duan henceforth). The default's time is modeled as the stopping time

$$\tau_D = \inf\{t : N_t > 0, M_t = 0\}, \tag{1.1}$$

where N_t and M_t are the counting processes governing default and other exits respectively. Similarly, the stopping time for combined exits is denoted by

$$\tau_C = \inf\{t : N_t > 0 \wedge M_t > 0\}. \tag{1.2}$$

Naturally, we have the following :

$$\left.\begin{array}{r} \text{if the firm exits due to default, } \tau_{Ci} = \tau_{Di}, \\ \text{if the firm does not exits due to default, } \tau_{Ci} < \tau_{Di}. \end{array}\right\} \quad \Rightarrow \tau_{Ci} \leq \tau_{Di}$$

Let us denote by Z_{it} the set of firm-specific variables at time t for the firm i and Y_t the set of

macroeconomic variables at time t. The first time of entry of the firm i is denoted t_i^0. The econometrician's information set $\mathscr{F}_t$ at time t is thus

$$\mathscr{F}_t = \{Y_s : s \leq t\} \cup \mathscr{G}_{1t} \cup \mathscr{G}_{2t} ... \cup \mathscr{G}_{Nt}, \tag{1.3}$$

where

$$\mathscr{G}_{it} = \{(1_{\tau_{C_i} < u}, 1_{\tau_{D_i} < u}, Z_{iu}) : t_i^0 \leq u \leq \min(\tau_D, \tau_C, t)\}. \tag{1.4}$$

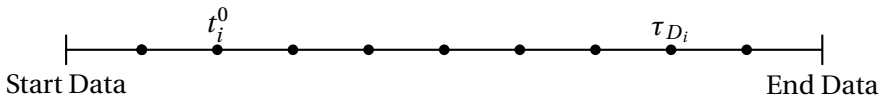


Figure 1.1: Example of the lifespan of a firm
Each dot corresponds to a time period where the econometrician gathers firm-specific (if available) and macroeconomic variables. t_i^0 is the entry time of the firm i and τ_{D_i} denotes the default time of the firm i.

In Figure 1.1, each dot corresponds to a period. At each period, the econometrician gathers firm-specific variables (DTD, Cash/TA, NI/TA, ...) and macroeconomic variables (S&P500 return, treasury rate). For a particular point in time t, the econometrician knows the whole time-serie of macroeconomic variables until t irrespective of t_i^0, and the time-series of firm-specific variables from t_i^0 to τ_{D_i} if $t_i^0 \leq t$ and $t \leq \tau_{D_i}$. Following Proposition 1 in Duffie et al. (2007), the conditional probability of default within s years can be computed as

$$\mathbb{P}[\tau_D < t + s | \mathscr{F}_t] = E_t \left[\int_t^{t+s} e^{-\int_t^z (\lambda(u) + \phi(u)) du} \cdot \lambda(z) dz \right]. \tag{1.5}$$

The probability of default is a function of intensities λ (default) and ϕ (other exits). However, these intensities are unknown and unobservable. In DSW model, the state variables governing Poisson intensities are assumed to follow a specific vector autoregressive (VAR) process. This assumption is relaxed in Duan's paper by using forward intensity rates. Instead of modeling λ_{it} and ϕ_{it} as some function of state variables available at time t, Duan et al. (2012) propose to deal with $f_{it}(\tau)$ and $g_{it}(\tau)$ directly as functions of state variables available at time t and the forward starting time of interest τ. The analogy to interest rates would be that λ_t is the short rate and $f_t(u)$ is the forward rate for horizon u. Duan et al. (2012) propose a model to predict corporate defaults at multiple horizons by estimating these forward intensities via maximum likelihood. To do so, they use a linear assumption in the relationship between the variables and the forward intensities (i.e $f_{it}(\tau) = \exp(\alpha_0(\tau) + \alpha_1(\tau) x_{it,1} + \alpha_2(\tau) x_{it,2} + ... + \alpha_k(\tau) x_{it,k})$. However, it is highly likely that default intensities depend on those covariates in a non-linear way. I propose to use an artificial neural network (ANN) to find the set of weights governing the process f_{it} and g_{it}. I show that I am able to capture potential non-linear relationships between the state variables and the forward intensities which significantly improves forecasts.

Forward intensities

Since we do not have the exact knowledge of λ and ϕ, Duan et al. (2012) propose to use forward intensity rates. However, we need to translate the default probability given in equation 1.5 (which depends on spot intensities) into a formula that depends on forward intensities. To do so, we first compute the probability of surviving as a function of forward combined intensity, which will be later used to compute the probability of default as a function of both combined and default forward intensities. Let us denote by $F_{it}(\tau)$ the conditional distribution function of the combined (default and other exits) exit time evaluated at $t+\tau$. Hence, 1-$F_{it}(\tau)$ is the probability of surviving in the interval $[t, t+\tau]$. Therefore, we have :

$$1 - F_{it}(\tau) = \mathbb{E}[e^{-\int_t^{t+\tau}(\lambda(s)+\phi(s))ds}]. \tag{1.6}$$

Then, let us introduce the quantity $\psi_{it}(\tau)$ to be :

$$\psi_{it}(\tau) \equiv -\frac{\ln(1-F_{it}(\tau))}{\tau} \equiv -\frac{\ln(\mathbb{E}[e^{-\int_t^{t+\tau}(\lambda(s)+\phi(s))ds}])}{\tau}. \tag{1.7}$$

Reverting equation 1.7 gives :

$$e^{-\psi_{it}(\tau)\cdot\tau} = 1 - F_{it}(\tau). \tag{1.8}$$

Where $e^{-\psi_{it}(\tau)\cdot\tau}$ is again the survival probability. We now need to compute $\psi_{it}(\tau)\cdot\tau$. At this point, Duan et al. (2012) makes the assumption that ψ_{it} is differentiable and defines the forward combined exit intensity as

$$g_{it}(\tau) \equiv \frac{F'_{it}}{1-F_{it}}. \tag{1.9}$$

Equation 1.9 comes from the definition of a hazard rate function. Referring P.87 of Schönbucher (2003), the definition of a hazard rate function is the following:

Definition 1.2.1 (Hazard rate)**.** Let denote by τ a stopping time and $F(T) \equiv \mathbb{P}[\tau \leq T]$ its distribution function. Assume that $F(T) < 1 \forall T$, and that F(T) has a density f(T). The hazard rate function h of τ is :

$$h(T) \equiv \frac{f(T)}{1-F(T)}.$$

A hazard rate is the local arrival probability of a stopping time per time interval. Please note that under suitable regularity conditions, intensities and hazard rates are closely similar. In particular, in our doubly-stochastic framework, hazard rates and intensities are equivalent. This is why in Duan et al. (2012), we have $\lambda(t) = h(t)$ and the distinction between hazard rates and intensity is often not made.

The relation between $g_{it}(\tau)$ and $\psi_{it}(\tau)$ is given by :

$$\begin{aligned} g_{it}(\tau) &= \frac{F'_{it}(\tau)}{1-F_{it}(\tau)} \\ &= \psi_{it}(\tau) + \psi'_{it}(\tau)\tau. \end{aligned} \tag{1.10}$$

Then, we can compute the quantity $\psi_{it}(\tau)\tau$ that we were looking for as :

$$\psi_{it}(\tau)\tau = \int_0^\tau g_{it}(s)ds. \tag{1.11}$$

For an interested reader, proofs of the above formulations can be found in the Appendix. Hence, the probability of surviving over [t,t+τ] is given by

$$\mathbb{P}[\tau_c > t+\tau | \mathbb{F}_t] = \exp(-\int_0^\tau g_{it}(s)ds). \tag{1.12}$$

The forward default intensity for horizon τ is defined as the limit for a small time step of the probability of defaulting in this small time step given that the firm has survived. The probability is Bayesian and the forward default intensity denoted $f_{it}(\tau)$ is the following :

$$f_{it}(\tau) \equiv \frac{\lim_{\Delta t \to 0} \frac{\mathbb{P}[t+\tau < \tau_{Di} = \tau_{Ci} \leq t+\tau+\Delta t]}{\Delta t}}{e^{-\psi_{it}(\tau)\cdot\tau}}. \tag{1.13}$$

Hence, the probability of defaulting between t and t+τ is given by :

$$\int_0^\tau e^{-\psi_{it}(s)s} f_{it}(s)ds = \int_0^\tau e^{-\int_0^s g_{it}(u)du} f_{it}(s)ds. \tag{1.14}$$

Likelihood function

In this setup, the likelihood function depends on three types of probabilities (default, other exit and surviving) which themselves depend on the types of intensities (default and other exits). The negative log-likelihood function has to be adjusted to the neural network framework and can be used as objective function to be minimized as long as we feed mini-batches and not single data points ("online learning"). Mini-batch feeding is a common practice in machine learning papers and consists of splitting the available data in batches of fixed size. Then, each backward pass takes one batch to perform one gradient decent step.

To allow further comparison with Duan et al. (2012), I employ the same discretization of time: $t = 0,1,2,...$ and $\tau = 0,1,2,...$ are time sequences of one month increment. Similarly, $f_{it}(\tau)$ and $g_{it}(\tau)$ are forward intensities computed at time t for the period [t+τ, t+τ+1]. The use of the τ index is to account for multiperiod prediction. When $\tau = 0$, the forward intensity model is com-

puting spot intensities. When we set $\tau = 1$, the forward intensity model produces estimates one step ahead, and so on so forth. I denote $X_{it} = (x_{it,1}, x_{it,2}, ...)$ the set of firm-specific and macroeconomic variables explaining both default and combined exit intensities. As specified in Duan et al. (2012) $f_{it}(\tau)$ and $g_{it}(\tau)$ are functions of X_{it} and can be specified as any form of function as long as they satisfy the following constraints. Since combined exit intensity has to be greater or equal than default intensity, we need to make sure that the forms specified for $f_{it}(\tau)$ and $g_{it}(\tau)$ satisfy the following conditions : $f_{it}(\tau) \leq g_{it}(\tau)$, $f_{it}(\tau) > 0$, $g_{it}(\tau) > 0$.

I have designed two neural networks, one will be trained to compute f_{it} and one will be trained to output h_{it} where $g_{it} = f_{it} + h_{it}$. I impose non-negativity on outputs of both models such that the combined exit intensity will never be smaller than the default intensity for all horizons. Let us denote by λ and μ the set of parameters (weights) tuned in the neural network for f_{it} and h_{it} respectively. $N^{(\lambda)}$ and $N^{(\mu)}$ stand for the output of the neural network for f_{it} and h_{it} respectively. The log-likelihood for prediction horizon τ is expressed as

$$\mathscr{L}(\lambda(s)) = \sum_{i=1}^{N} \sum_{t=0}^{T-s-1} \mathscr{L}_{i,t}(\lambda(s)), \qquad s = 0, 1, ..., \tau - 1 \tag{1.15}$$

$$\mathscr{L}(\mu(s)) = \sum_{i=1}^{N} \sum_{t=0}^{T-s-1} \mathscr{L}_{i,t}(\mu(s)), \qquad s = 0, 1, ..., \tau - 1 \tag{1.16}$$

where

$$\begin{aligned} \mathscr{L}_{i,t}(\lambda(s)) = & \underbrace{\mathbf{1}_{t_{0i} \leq t, \tau_{Ci} > t+s+1}}_{(1)} \cdot (-N_{it}^{(\lambda)}(s)) \Delta t) \\ & + \underbrace{\mathbf{1}_{t_{0i} \leq t, \tau_{Di} = \tau_{Ci} \leq t+s+1}}_{(2)} \cdot \ln(1 - \exp[-N_{it}^{(\lambda)}(s) \Delta t]) \\ & + \underbrace{\mathbf{1}_{t_{0i} \leq t, \tau_{Di} \neq \tau_{Ci}, \tau_{Ci} \leq t+s+1}}_{(3)} \cdot (-N_{it}^{(\lambda)}(s) \Delta t), \end{aligned} \tag{1.17}$$

$$\begin{aligned} \mathscr{L}_{i,t}(\mu(s)) = & \underbrace{\mathbf{1}_{t_{0i} \leq t, \tau_{Ci} > t+s+1}}_{(1)} \cdot (-N_{it}^{(\mu)}(s) \Delta t) \\ & + \underbrace{\mathbf{1}_{t_{0i} \leq t, \tau_{Di} \neq \tau_{Ci}, \tau_{Ci} \leq t+s+1}}_{(3)} \cdot \ln(1 - \exp(-N_{it}^{(\mu)}(s)) \Delta t)). \end{aligned} \tag{1.18}$$

The likelihoods above are a sum of indicator functions multiplied by their respective probability. The indicator function $\mathbf{1}_{A<B}$ is equal to one if $A < B$ or zero if $A \geq B$. Hence, the likelihoods above are specifying three mutually exclusive indicator functions which define three independent cases during time interval $[t, t + \tau + 1]$:

1. The firm does not exit the sample between t and $t+\tau$ and is considered as surviving. This case is specified as (1) in the likelihood because the combined exit time τ_{Ci} is not in the interval $[t, t+\tau+1]$.

2. The firm exits the sample due to default during the interval and is considered as defaulted. This case is specified as (2) because $\tau_{Ci} = \tau_{Di}$ when the firm exits due to default conjointly with τ_{Di} being in the interval $[t, t+\tau+1]$.

3. The firm exits due to other reasons specified as (3) during the interval since the stopping time $\tau_{Di} \neq \tau_{Ci}$ conjointly with $\tau_{Ci} \leq t+\tau+1$.

Proof of the above formulation can be found in the Appendix. As in previous studies, the likelihood functions still exhibit the decomposable property which allows to estimate the model for each horizon of prediction independently.

Since the intensities are directly driven by the covariates, Duan et al. (2012) requires an assumption on the mapping from the covariates to these intensities. In Duan et al. (2012) the mapping is made with a linear assumption, whereas in the framework of this paper, the mapping depends on the whole architecture of the neural network. When the neural network has only one hidden layer of one neuron coupled with an exponential activation function, the model boils down to Duan et al. (2012) as the intensities will be a linear combination of the covariates. As the width and depth of the network increases, we depart more and more from the linear assumption and we allow more non-linearities to be incorporated in these intensities.

1.2.2 Neural Networks

Neural networks can be seen as a very general function to map a given input (in this case a set of macroeconomic and firm-specific variables) into a desired output (forward intensities). It learns how to compute the output by tuning weights in order to minimize a given loss function. A neural network is constructed by juxtaposing several "hidden" layers of neurons. The input of each layer is a data transformation of the output of the previous layer. Initially, the weights of the network are assigned random values. Then, the training process starts and consists of many iterations of a forward pass and a backward pass. The forward pass takes as input a batch of data and computes the loss value, the backward pass then computes the gradient and adjust the weights of the network based on a learning rate hyperparameter. As an illustration, figure 1.2 shows how a neural network looks like with 2 hidden layers : 5 neurons in the first layer and 3 neuron in the second layer. In this example, 3 features (inputs) are fed to the network. Each feature is connected to the first hidden layer by a set of weights. The outputs of the first layer are also weighted to produce the inputs of the second hidden layer. Non-linearity is introduced in each node with a non-linear activation function (i.e sigmoid). Finally, the output of the second hidden layer is aggregated to produce the final output of the model.

The neural networks in this paper are implemented in Python using the library TensorFlow. Approximately one hour of computing time is needed to fit all networks on a 32GB RAM quad core 2.7 GhZ computer, and GPU computing is not necessary as the networks are rather small.

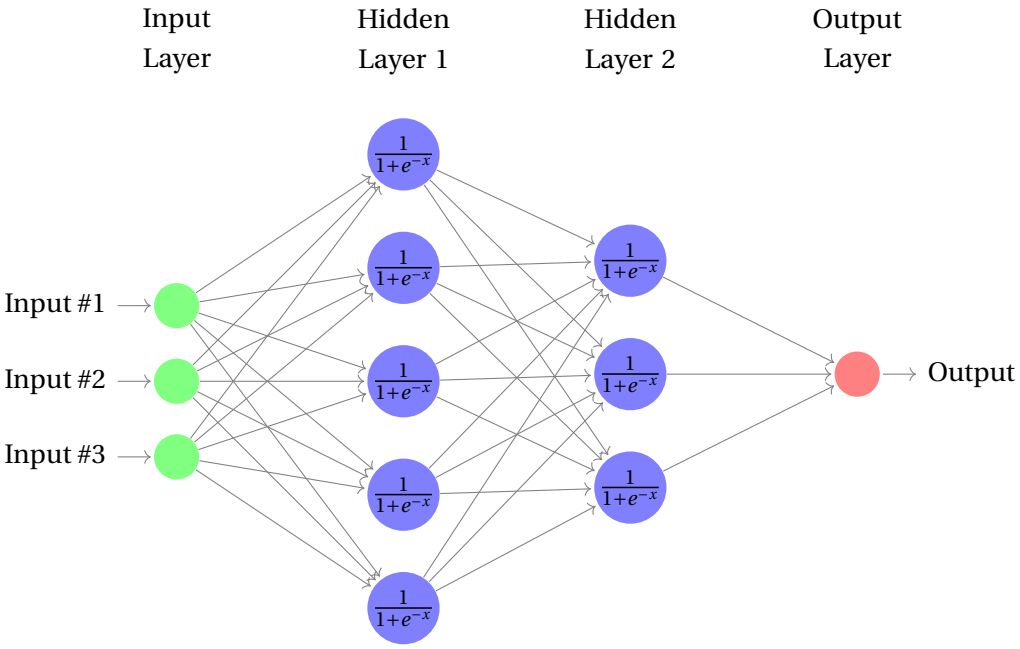


Figure 1.2: Illustration of a neural network [5, 3]
In this network, there are three input features, five neurons in the first hidden layer and 3 neurons in the second hidden layer. Each neuron is activated with a sigmoid function.

The use of neural networks can be motivated twofold. First, neural networks are well suited to approximate a function (in our case forward intensities) with the advantage of having different degrees of modularity. By definition, the architecture of the network generates the form of the function approximated. A deeper network allows for more non-linearities in the approximation of the function, at the expense of having more parameters to estimate. For instance, suppose that every observation comes with 12 input features (i.e : x is a vector of shape 1x12), a [5, 3] (i.e 2 layers neural network with 5 neurons in the first hidden layer and 3 neurons in the second hidden layer) can be viewed as a function computing the output $N_{it}^{(\lambda)}$ in the following way :

$$N_{it}^{(\lambda)} = \Big[\phi_1\big(\big[\phi_2\big(\big[x\big]_{1\times 12}\big[w_1\big]_{12\times 5} + \big[b_1\big]_{1\times 5}\big)\big]_{1\times 5}\big[w_2\big]_{5\times 3} + \big[b_2\big]_{1\times 3}\big)\Big]_{1\times 3}\big[w\big]_{3\times 1},$$

with the activation functions being for instance the sigmoid function $\phi_1(x) = \phi_2(x) = \frac{1}{1+e^{-x}}$, x being the data input, w_1, w_2 and w weight matrices and b_1 and b_2 biases matrices. In this setup, the number of parameters to estimate is equal to 12x5 + 1x5 + 5x3 + 1x3 + 3x1 = 86.
Neural networks are also well-suited for this paper because they allow an easy replication of the benchmark model Duan et al. (2012). More specifically, if the activation function $\phi(x)$ is

chosen as being an exponential $\exp(x)$, and the network architecture is [1] (i.e a single hidden layer with a single neuron), the output $N_{it}^{(\lambda)}$ becomes

$$\begin{aligned} N_{it}^{(\lambda)} &= \phi(\Big[x\Big]_{1\times 12} \cdot \Big[w\Big]_{12\times 1} + \Big[b1\Big]_{1\times 1}) \\ &= \exp(b1 + w_1 \cdot x_1 + w_2 \cdot x_2 + ... + w_{12} \cdot x_{12}), \end{aligned}$$

Results of this architecture are described in section 4.

1.3 Empirical section

1.3.1 Data

The accounting data is taken from Wharton Research Data Services (WRDS) using the CRSP/Compustat merged database. The macroeconomic data is taken from CRSP, the Federal Reserve Bank Reports and Datastream. The bankrupcty data is taken from the Compustat database, using the DLRSN item for the reason of deletion and the DLDTE item for the date of deletion. DLRSN contains the code that indicates the reason a company become inactive on the database. I consider firms with a DLRSN code 2 (bankruptcy) or 3 (liquidation) to be defaulted, any other DLRSN code as “other exits" and no DLRSN code as surviving. For additional information on DLRSN and DLDTE codes, please refer to the Wharton WRDS documentation. I focus on the period from 1991 to 2018 to match the accounting data with the bankruptcy data. Using WRDS database, I have downloaded accounting information for every company that has been listed someday on either NYSE, AMEX or NASDAQ between 1991 and 2018. This means I have 27 years of data where firms entered and/or exited anywhere in this sample. Using this kind of sample brings a problem of cylindric data : firm's entering/exiting time obviously are not the same for each company. We can see the whole dataset as a 3D matrix with x-axis being features, y-axis being time, and z-axis being firm. We only need to fill the matrix with NaN for elements where the firm i doesn't exist or already left at time t. Since neural networks need plenty of data points to be well trained, I chose not to remove firms even if it has a short lifespan. When a variable is completely missing for a firm, I drop the whole firm because the likelihood is not specified if a variable is fully missing. However, when the variable is not fully missing but only some data points are not available, I use the last available information before the missing entry. I winsorize all variables at the 2.5 and 97.5 percentile. Before dropping firms, the dataset has in total 12527 surviving firms, 1453 defaulted firms and 17950 firms that exited the sample for other reasons. Finally, I standardize all variables by substracting the mean of the variable and dividing the result by its standard deviation. The test set is also standardized using the mean and standard deviation from the training set. Table 1.1 shows the number of firms in the three categories for each horizon of prediction τ which corresponds to the firm-month observations used in the likelihood for horizon τ (see

Horizon	Surviving	Defaults	Other exits
0	2'025'094	499	10'392
3	1'972'063	514	10'746
6	1'918'061	499	10'713
12	1'811'323	454	10'393
24	1'610'242	382	9'193
35	1'457'240	343	8'295

Table 1.1: Number of firms in each category for each horizon of prediction τ
These correspond to the firm-month observations used in the likelihood for horizon τ.

equations 1.17 and 1.18). Figure 1.3 shows the total number of firms that defaulted, survived or exited for other reasons plotted on a year on year basis.

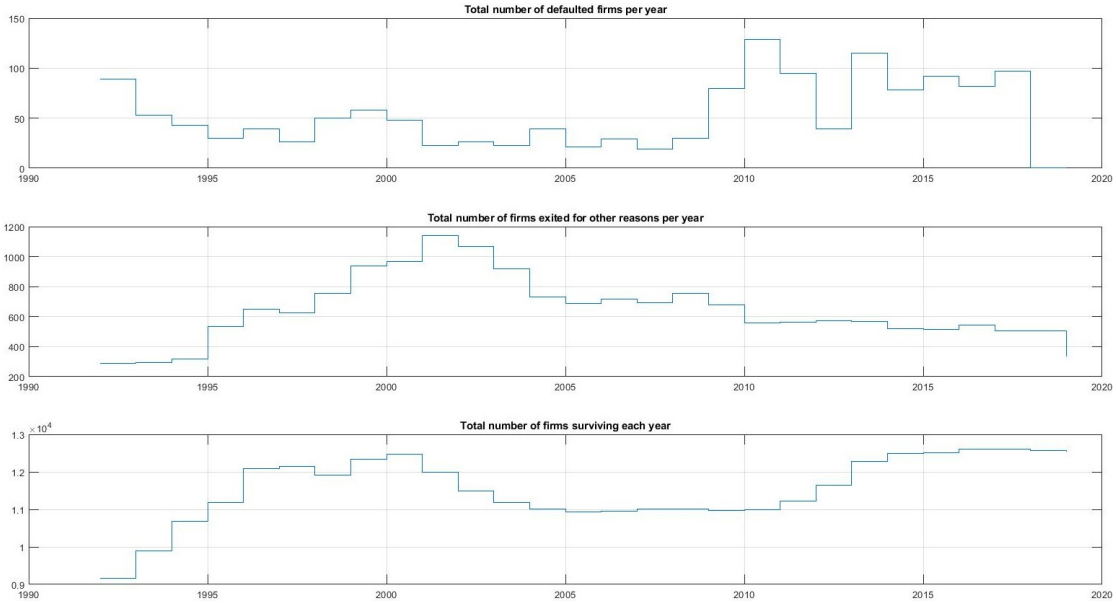


Figure 1.3: Number of firms defaulted, exited for other reasons, surviving each year

Leippold et al. (2012) used a theoretical model to show that the most powerful default predictor must incorporate both macroeconomic and accounting data. For more transparency and to allow better comparison with previous literature, I chose to work with a similar set of features similar than my benchmark model Duan et al. (2012). Firm-specific values are common to each firm, macroeconomics variables are function of market data. The exhaustive list of variables is the following :

1. SP500 : trailing 1-year return on the S&500 index.
2. Treasury : 3-month annualized US Treasury bill rate. I use this variable as the risk-free rate r in the model.

3. DtD : the Distance-to-default is a volatility adjusted leverage measure for gauging how far is the firm from default. It was first introduced by Merton (1974) model which treat firm's equity value as a call option on the underlying asset and the strike being the debt level. The DtD can be viewed as the number of standard deviations of annual asset growth by which the asset level exceeds the firm's liabilities. In this paper, the DtD is estimated using a maximum likelihood taking into account other liabilities of each firm to handle financial firm's bias. It is well-known in the literature that DtD is a significant measure to estimate default probabilities but has to be used conjointly with other variables. See the Appendix for additional information regarding the estimation procedure of DtD.
4. CASH/TA : ratio between cash and short-term investment to total assets. Both quantities are taken from the balance sheet of the company.
5. NI/TA : ratio between net income to the total assets. A loss is registered as a negative net income.
6. SIZE : logarithm of the ratio between a firm market equity value to the average market equity value of the whole S&P500. The market equity value is computed as the stock price multiplied by the number of outstanding shares at the end of each quarter. SIZE is negative if the firm has a smaller market capitalization than the average market capitalization and positive otherwise.
7. M/B : market-to-book ratio computed using the ratio between the asset market value from the DtD maximum likelihood estimation to the total book asset value.

To capture momentum of variables, I also compute one step rolling window differences for each firm-specific variables. These variables are called "Δ" followed by the name of variable. They are telling whether the firm has been improving or deteriorating with respect to this particular variable comparing to the last period performance. Given this model specification, the Δ is particularly interesting because if a firm shows many consecutive negative delta values, it really means the company is in danger. However, it is also important to look at the level value to compare a defaulted firm with a non-defaulted firm. The intuition tells us that prior default time, a defaulted company should have shown lower level values (for instance, DtD) than a non-defaulted firm.

1.3.2 Summary statistics

This section depicts the summary statistics of the dataset. Table 1.2 shows a summary of the means of variables for each three categories of firm. Please note that we have to be very careful when comparing two means in this table. Comparing means needs to be done with confidence intervals and tests of significance which involves standard deviations. This table is presented to give a rough idea without performing any statistical inference. The table shows that the average defaulted firm is smaller, has a lower DtD, a smaller Market-to-book ratio,

	Surviving	Default	Other exits
Cash/TA	0.1952	0.1896	0.1851
NI/TA	-0.0756	-0.2057	-0.129
Size	-2.7845	-4.6743	-3.621
DtD	10.641	6.441	8.401
MBratio	2.3774	1.6995	2.202
Δ CASH/TA	-0.0026	0.0023	-0.0074
Δ NI/TA	-0.0019	-0.0433	-0.0148
Δ Size	-0.0305	-0.5095	-0.0939
Δ DtD	-0.1366	-0.7346	0.0295
Δ MBratio	-0.0285	0.0520	0.0801

Table 1.2: Mean of variables for surviving firms, defaulted firms and other exits
Δ are one period lagged differences.

loses more money and has less cash. The prefixes Δ in front of the variables stand for the one-lag differences. Finally, Figure 1.4 shows the correlation matrix for the twelve covariates.

1.4 Results

As every neural network, we need to chose the optimal hyperparameters to achieve the highest performance. In my setup, this consists mainly of choosing the architecture of the model (i.e the number of neurons and layers). To do so, there is currently no other better method than trial and error. However, I need to be careful to not chose hyperparameters that overfit the test set. To avoid any overfitting, I perform a 5-fold cross validation for each horizon of prediction. I cut 15% of the dataset as test set, all results that I will talk about in this chapter are out-of-sample and performed on the observations of the test set that the model has never seen before. The remaining set of observations is partitioned into smaller subsets so that in every fold of the validation a different subset is used as validation set and the rest is used as training set. Finally, to measure the discriminatory power of the models, I use the Lorenz curve (Lorenz (1905)) and I use the Gini coefficient as a scalar performance measure to aggregate across folds to get the measure that I will use to discriminate models.

Definition 1.4.1 (Lorenz curve)**.** The Lorenz curve of a predictor P is the two-dimensional graph

$$(\mathbb{P}\{P \leq p\}, \mathbb{P}\{P \leq p | Y = 1\}),$$

$\forall\, p \in (-\infty, +\infty)$.

The Lorenz curve plots on the x-axis the cumulative percentage of observations against the fraction of defaults on the y-axis. These curves are often used in the literature for default prediction (for instance Leippold et al. (2012), Duan et al. (2012)) and I have seen very similar plots under two other names : power curves and cumulative accuracy profiles. They are

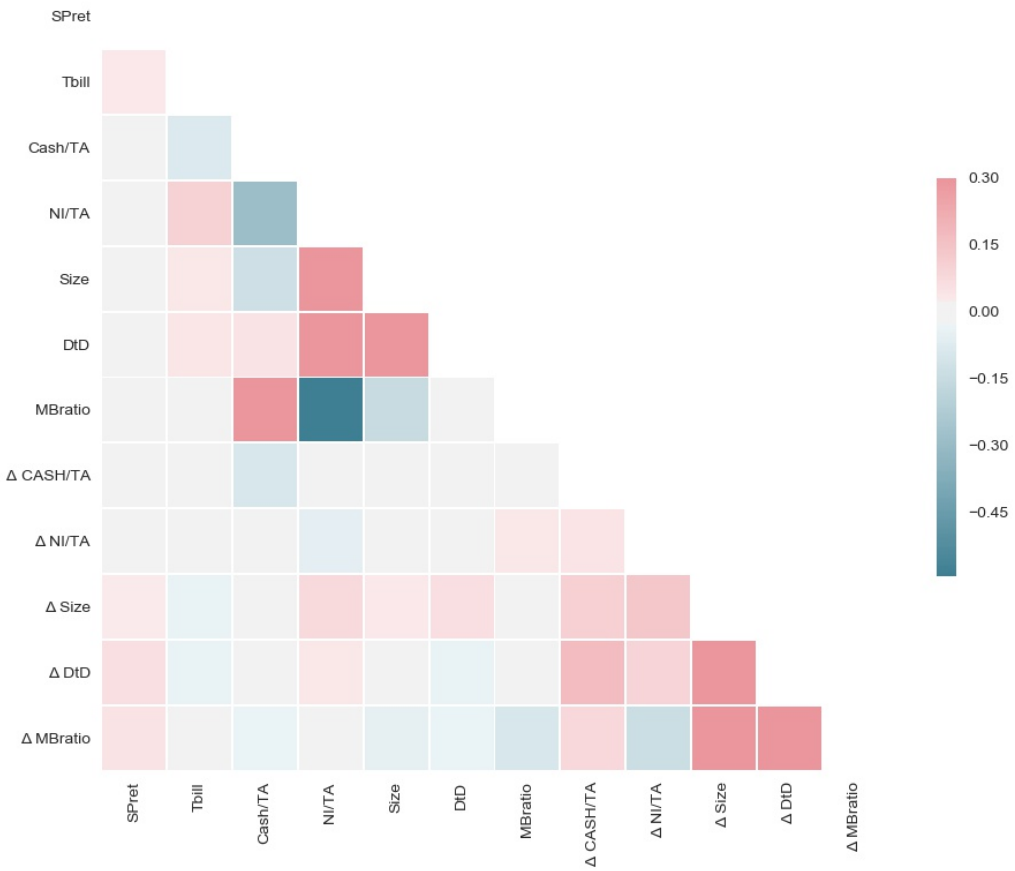


Figure 1.4: Correlation matrix for firm-specific and macroeconomic covariates

different from ROC curves and precision-recall curves because they don't rely on thresholds to discriminate of true positives against false positives. I believe that they are particularly well-suited as performance measure in this model because of the multiperiod framework involved. The idea is that if the model is outputting a false positive for an horizon x but the true positive is horizon x+1, the model should not be "too much" penalized. The ROC and precision-recall curves would treat this as a false positive even though the prediction was not that far from the target. The idea behind the Lorenz curve is to order default probabilities and look how they are distributed across defaulted and non-defaulted firms. We can easily see whether the model is outputting high probabilities for defaulted firms and small probabilities for surviving firms.
Finally, the Gini coefficient is used as my scalar summary statistic to compare models. It measures the degree of inequality of the Lorenz curve. A perfect model has a Gini coefficient close to 1 (depending on the fraction of defaults in the dataset) and a poor model has a Gini coefficient of 0 (perfect equality).

1.4.1 Model choice

The choice of the optimal architecture of the neural network has been made using k-fold cross-validation. The average Gini coefficient across all folds for every horizon is used as comparison tool to determine the best model architecture. If the architecture is too deep, the implicit function computed in the network to output forward intensities incorporates too many parameters and the risk of overfitting is larger, resulting in a lower accuracy measure. Similarly, if the network is not deep enough, the forward intensities are computed using a too simplistic representation and result in a low accuracy measure as well. This is usually known in the machine learning literature as the bias-variance tradeoff. Figure 1.5 shows the average Gini coefficient compute on the validation test across all folds of the cross validation for each horizon. I compare the scores obtained with different network architectures with those using the linear assumption. In this setup, one can interpret "architecture" as "non-linearity degree" as higher architecture involves more weights in the output function to be estimated. By disantangling the spaghettis, we can clearly see the bias-variance tradeoff. Increasing the depth of the networks from the simplest neural network [1] to a two layers network [2, 1] increases the Gini coefficients, in particular at mid horizons. This is probably due to the non-linearities introduced via the second layer. Next, it looks like the [3, 2] performs similarly as the [2, 1]; but we clearly see that [5, 3] dominates any other previous models. Finally, increasing the depth again to [10, 5] decreases Gini coefficient at all horizons, suggesting a severe overfitting of the forward intensities function. For next sections, I will now only show out-of-sample results of the [5, 3] architecture using the test set preliminary set on the side.

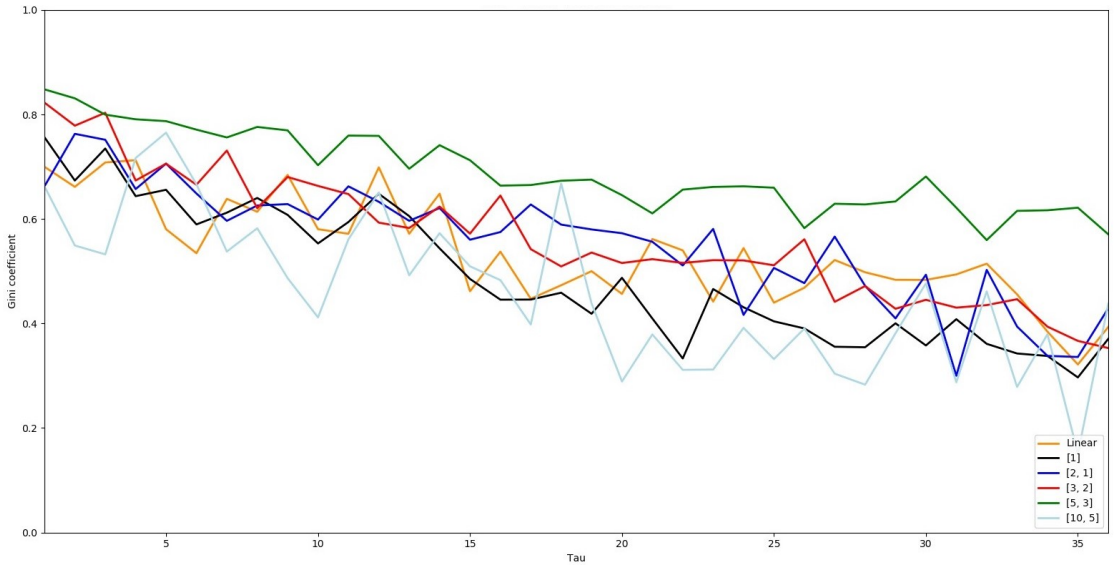


Figure 1.5: Gini coefficient : Linear vs Neural Networks

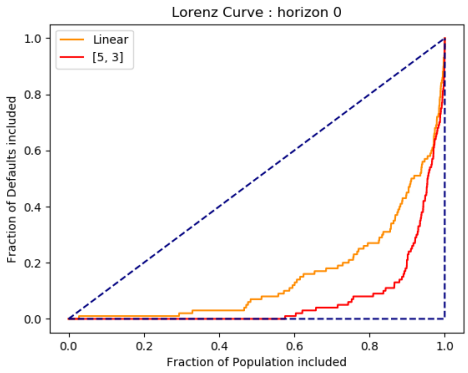
1.4.2 Performance

Figure 1.6 shows Lorenz curves for horizon 0, 3, 6, 12, 24 and 35. Please note that all results generalize well to all others horizons. The curves are completely out-of-sample since they are computed on the test set that the model has never seen before. Table 1.5 shows Gini coefficients for both the linear assumption Duan et al. (2012) and for the [5, 3] for the same horizons. Overall, as expected the neural network clearly outperforms the linear assumption, suggesting that the linear assumption from Duan et al. (2012) can be greatly improved by adding non-linearities in the specification of the intensities. Unfortunately, it is difficult to tell which kind of non-linearities should be taken into account since neural networks are often seen as a black box. However, I will still try to answer this question by looking how the average intensity outputted by the model changes when we change a feature ceteris paribus (see section dedicated to “sensitivities"). Another attempt at answering this question is described in the section “computational graph”, where I deep dive into the weights of the network to understand how the output is computed.

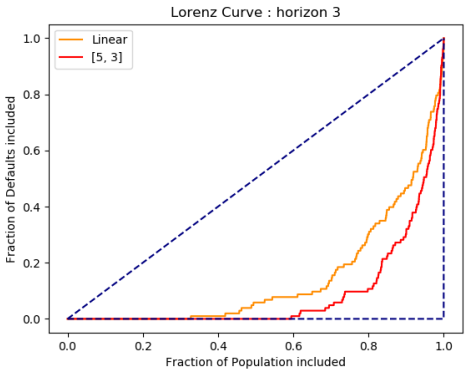
For comparison purposes, figure 1.7 shows the Lorenz curves for both linear assumption, its replication in the neural network framework, and the [5, 3] model. The linear assumption and its replication have similar performance, suggesting again that the neural network “NN+exp+[1]" is able to easily replicate the linear framework depicted in Duan et al. (2012).

1.4.3 Computational graph

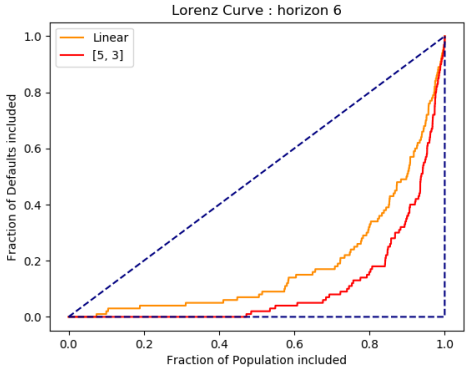
Figure 1.8 is a representation of a fully trained neural network for the forward default intensity f at horizon 0. Negative weights are drawn as blue lines connecting neurons where orange



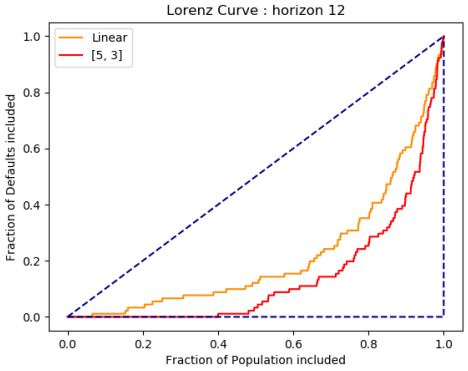
Horizon 0



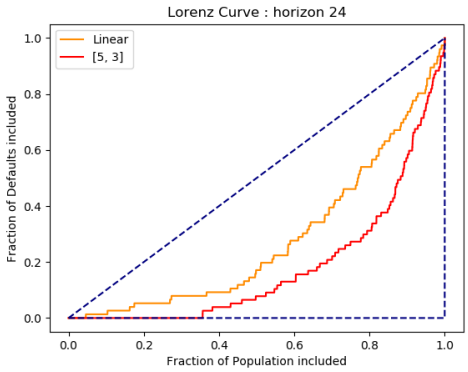
Horizon 3



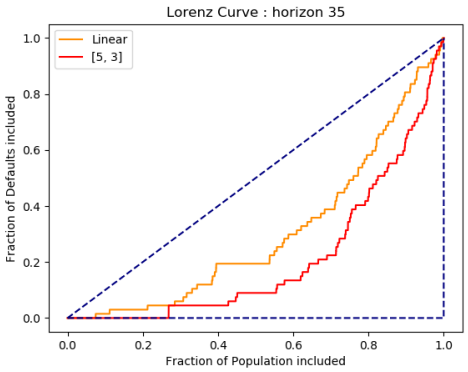
Horizon 6



Horizon 12



Horizon 24



Horizon 35

Figure 1.6: Out-of-sample Lorenz Curves for each horizon
The orange line shows the performance of Duan's model and the red line shows the performance of the [5,3] neural network.

Horizon	[5, 3]	Linear
0	0.85	0.70
3	0.79	0.71
6	0.76	0.64
12	0.70	0.57
24	0.66	0.44
35	0.57	0.39

Table 1.3: Gini coefficients

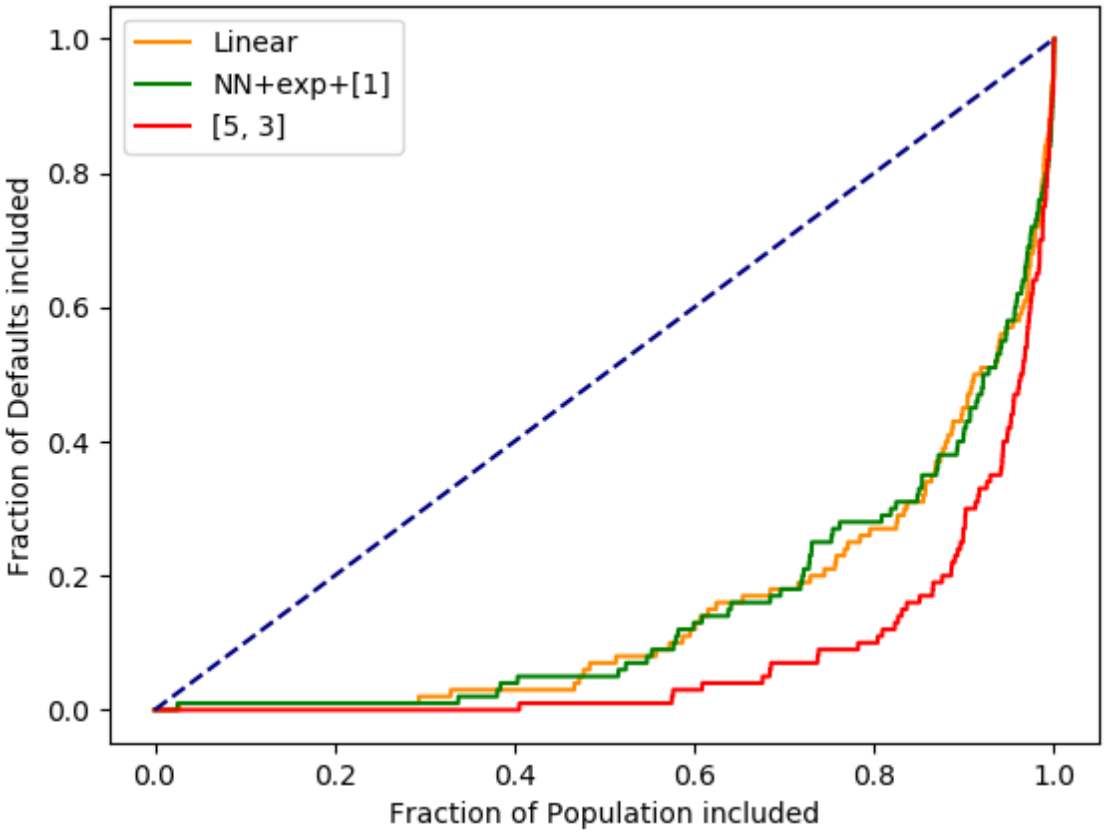


Figure 1.7: Comparison with the benchmark model Duan et al. (2012)
The green line shows the performance of the [1] neural network (i.e a single hidden layer with a single neuron) with an exponential activation function. The orange line shows the performance of Duan's model and the red line shows the performance of the [5,3] neural network.

lines show positive weights. The thicker the line, the higher the weight is in absolute value. Recall that each neuron is activated with a sigmoid function and the biases are not shown on the graph. In the input layer, all variables seem to be used in the computation of the first layer. In the first layer however, the second neuron presents a higher weight in the network than the others. In return, the inputs connected to the second neuron of the first hidden layer all present low relative weights. Even though the computational graph gives an overview of the neural network and is useful to understand the way the output is computed, it is not trivial to see which variables have more impact on the output. In the following section, I plot sensitivities of each variable in each horizon to better understand the causality of each input.

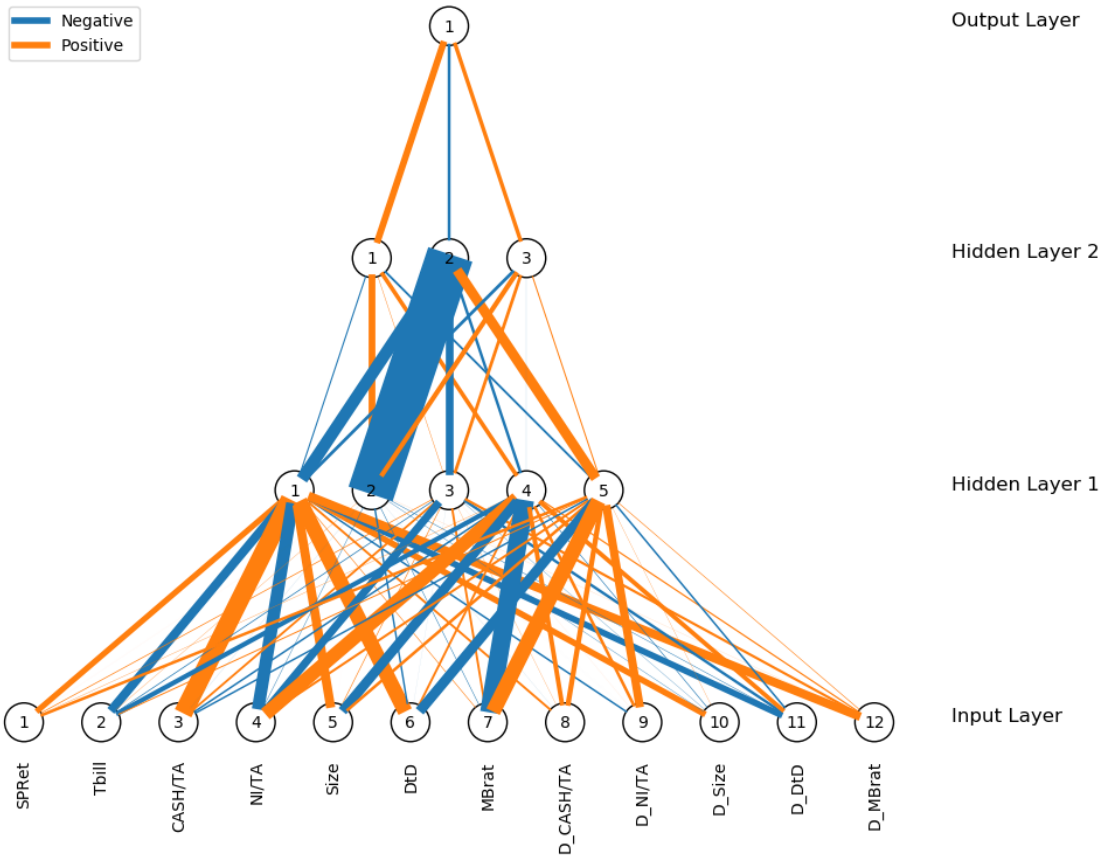


Figure 1.8: Trained [5, 3] Neural Network for forward default intensity at horizon 0
Negative weights are drawn as blue lines connecting neurons where orange lines show positive weights. The thicker the line, the higher the weight is in absolute value.

1.4.4 Sensitivities

Neural networks are often seen as black boxes because their outputs are coming from a general function involving many parameters. It is incorporating non-linearities via the layers and the activation functions. Figure 1.9 is an attempt at gauging how the model reacts to a change of

an input variable. It plots the average default forward intensity against a shift in the specified variable and is computed the following way:

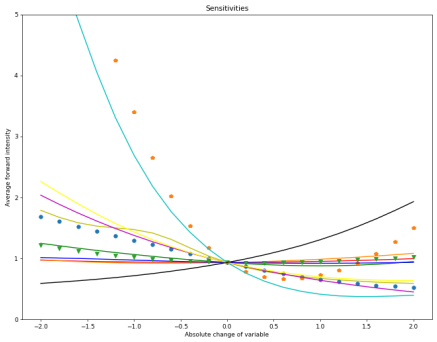
1. Compute the forward default intensity for all the observations in the test set and average the result. Then, keeping everything else equal, change the value of one feature by an absolute value and feed the “updated" observations to the network as new input and compute the new average forward default intensity.
2. Repeat step 1 for all absolute values in some interval
3. Plot the average intensity against the absolute change
4. Repeat step 1-3 for all 11 other features.

Keep in mind that there is a non-negativity constraint on forward intensities. A decreasing relationship means that an increase of the associated variable decreases the probability of default of the firm. A flat relationship means that the associated variable has a limited impact on the probability of default. We should expect CASH/TA, size, DtD, NI/TA, Market-to-book ratio and all their lagged differences (Δ) to be decreasing.

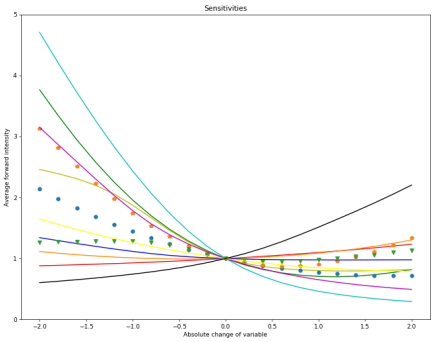
First of all, most graphs show non-linear relationships, which is not surprising given the nature of the neural network specification. Moreover, all relationships are intuitive and expected. Forward intensities in both Size and ΔSize are decreasing, suggesting that small firms tend to have higher likelihood of defaulting which is consistent with the “too big to fail" paradigm. Similarly, firms with decreasing cash (ΔCASH/TA < 0) or low levels of CASH/TA appear to have higher probability of default. The model also predicts that firms with low NI/TA or decreasing NI/TA should have higher probability of default. Finally, forward intensities in DtD should be decreasing for all horizons to reflect that a higher distance to default makes the firm less likely to default. At horizon 0, a negative change in distance-to-default has a substantially greater effect on forward intensities than any other variables. This result is consistent with Duffie et al. (2007). Overall, it appears that the most important predictors of default in this model are in short term Market-to-book ratio, DtD and CASH/TA, and in the long run NI/TA, CASH/TA, ΔCASH/TA and DtD.

1.5 Conclusion

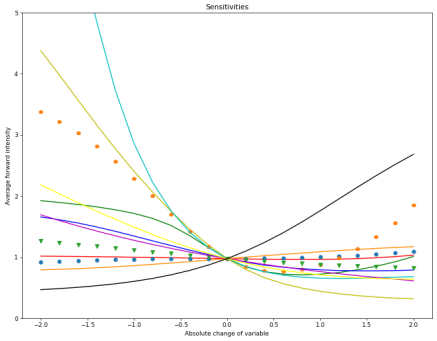
I propose an approach to estimate default forward intensities which relies on using machine learning techniques. The key improvement over previous estimation methods is the introduction of possible highly non-linear relationships between covariates and forward intensities. Neural networks are nothing else than a very general mapping of input data to an output which is obtained by tuning weights while minimizing a given loss function. Non-linearities are introduced via the juxtaposition of layers and the activation functions. The econometric



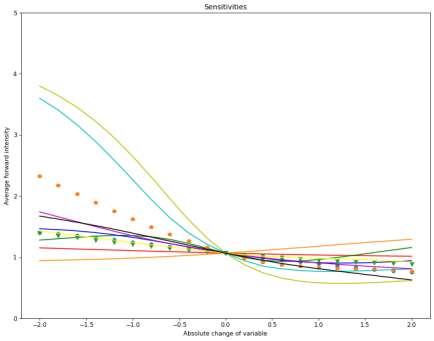
Horizon 0



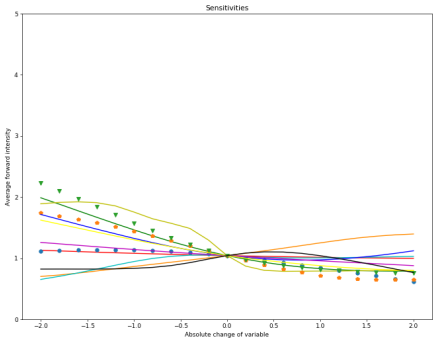
Horizon 3



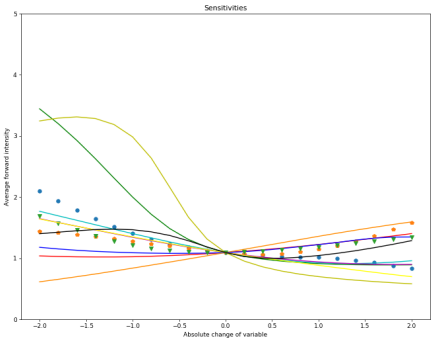
Horizon 6



Horizon 12



Horizon 24



Horizon 35

Figure 1.9: Sensitivities for each horizon

model governing the forward intensity written by Duan et al. (2012) has been adapted to this new framework to allow the use of neural networks. Neural networks are also well-suited for this paper because they allow an easy replication of the benchmark model Duan et al. (2012). More specifically, if the network architecture is [1] (i.e : one layer and one neuron) and the activation function is chosen as being an exponential $\exp(x)$, the neural network boils down to a logit regression. The dataset used in this paper is the same as previous literature to allow easier comparison. It consists of 5 firm-specific variables computed from accounting data and 2 macroeconomic variables to control for the health of the economy. I also account for momentum of these variables by feeding the model the one lagged differences of each variable. Looking at summary statistics only, the average defaulted firm is small, has low cash, low market-to-book ratio, low Distance-to-Default and has large and negative profits. To measure the discriminatory power of the models, I follow previous literature and use Lorenz curves (also known as “Cumulative accuracy profile" or “Power curves"). The idea behind Lorenz curves is to order default probabilities and look how they are distributed across defaulted and non-defaulted firms. The average Gini coefficient across all folds of the cross-validation is used as comparison tool to gauge the accuracy of the model. Results show that the architecture [5, 3] (i.e 2 layers with 5 neurons in the first hidden layer and 3 neurons in the second hidden layer) seems to outperform others architecture. In this setup, one can interpret “architecture” as “non-linearity degree” as higher architecture involves more weights in the output function to be estimated. Out-of-sample Lorenz Curves and Gini coefficients show that the neural network approach outperforms the linear assumption for every horizon, suggesting the presence of non-linearities in forward default intensities. Finally, even if neural networks are known as being black boxes, one can try to see how the model reacts to a change of input variables. Most sensitivities plots show non-linear relationships, which is not surprising given the nature of the neural network specification. It appears that the most important predictors of default in this model are in the short term Market-to-book ratio, Distance-to-default and NI/TA, and in the long run NI/TA, CASH/TA, ΔCASH/TA and Distance-to-default. Further works could involve more variables in the estimation. In particular, a challenging (due to lack of data) but nonetheless exciting study would be to gauge the effect of market sentiment on default intensities.

2 Causal Networks with Neural Networks

2.1 Introduction

The causal network of a dynamical system provides important information that may help to better understand its behavior and ultimately design better policies to predict and control it. Large number of banks, insurances, hedge funds, and other financial institutions around the globe are interacting daily and thus their causal network is of great importance in econometrics.

There have been many attempts during the past decades to capture and visualize the network of interconnections among a set of financial institutions. The most widely used concept of causality in time series econometrics is due to Granger (1969). This is based on statistical analysis of the financial series such as their stock prices over a finite time period. Granger's definition of causality states that a time series X is a cause of another time series Y, if the mean squared error of the 1-step ahead forecast for Y is smaller when the history of X is included in the forecasting information set. Otherwise, when the forecast does not improve by including the information of X, then it is declared that X does not cause Y. This idea is reflected in the information-theoretic measure that we use in this work to infer the causal interactions among a network of time series.

In the great majority of practical applications, Granger-causality has been studied in the context of Vector Autoregressive (VAR) models. For instance, Billio et al. (2012) proposes several measures based on Granger-causality to capture the connections between the monthly returns of different financial institutions. It uses principle component analysis and "pairwise" Granger-causality tests to identify the causal networks. Other related works are Diebold and Yılmaz (2014) and Barigozzi and Hallin (2016) in which the authors propose connectedness measures based on generalized variance decomposition. However, the measures introduced in these works are again limited to linear systems and they are based on pairwise comparison which as we show in Section 2.2.2 fails to infer the true causal relationships.

Contributions of this paper are both in network identification literature as well in finance. Our

contribution to network identification are as follows:

- We use an information-theoretic measure known as directed information (DI) to infer the Granger-causalities among a set of time series. This measure is non-parametric, i.e., it does not depend on the underlying model of the dynamics and it is capable of capturing causal relationships in both linear and nonlinear systems. The output of this approach is a directed graph known as Directed Information Graph (DIG) that visualizes the interconnections among a set of time series such as stock returns.
- Computing DI has both high computational and sample complexity which makes it not suitable for inferring the causal structure of large networks. To overcome this problem, we develop a novel approach based on Recurrent Neural Networks (RNNs) that reduces the complexity of evaluating DIs in high-dimensional settings.

Applications of DIG are various in finance. In particular, we recommend to use it as a preliminary feature selection of another predictive model. Feature selection is a process often used in machine learning and statistics which consists of keeping only a subset of relevant features, usually to avoid overfitting or to reduce dimensionality. For example, Piramuthu (2004) and Huang and Wang (2006) show that extraneous features are prone to reduce model's performance measures. Finance applications of feature selection models are various and include credit scoring, stock market behavior analysis or even fraud detection (Altinbas and Biskin (2015)). Tsai (2009) states that feature selection preprocessing is not addressed carefully enough in the bankruptcy prediction literature. They compare five feature selection methods used in bankruptcy prediction: t-test, correlation matrix, step-wise regression, Principle Component Analysis (PCA) and factor analysis and show that any of these methods improves performance. Yuqinq et al. (2013) uses a Sequential Forward Selection algorithm to select relevant features predicting the Turkish market index. The use of such feature selection model reduces model prediction error compared to the case where all features are used. This is due to information embedded in several economic factors already included in the market index. They show that only the lagged value of the market index is enough to predict the forthcoming value of the index.

2.1.1 Related Work

In recent years, several approaches have been developed to generalize the applicability of Granger-causality to non-linear and large dynamics. To mention a few, Psaradakis et al. (2005) that introduces different terminologies for causality based on Granger's ideas and provide a set of parametric non-causality constraints in the context of Markov switching VAR models. In a similar context, Bianchi et al. (2019) investigates time-varying systemic risk based on a range of multi-factor asset pricing models and develops a Markov Chain Monte Carlo (MCMC) scheme to infer their model parameters and consequently obtain their corresponding networks. Another attempt is Bonaccolto et al. (2019) in which the authors explore quantile-

based methods of Granger causality. This method is consistent with Hong et al. (2009) and Corsi et al. (2018) that focus on causality among tail events. These methods are suitable for capturing causal relationships that are not in the center of their distributions, or in the mean but they are in the tails of their distributions. It is important to emphasize that our proposed approach using DI is also capable of capturing such causal relations.

Most of the above aforementioned approaches are developed and tested for small size networks. Often, the problem of network identification in high dimensional settings requires more considerations and even its own techniques. For instance, Billio et al. (2019) proposes a Bayesian non-parametric Lasso prior (BNP-Lasso) for high-dimensional VAR models that can improve efficiency and accuracy. In order to overcome overparametrization and overfitting issues in large VAR models, BNP-Lasso clusters the VAR coefficients into groups and shrinks the coefficients of each group toward a common location. However, this method is limited to linear models with Gaussian innovations. To overcome this limitation, Kalli and Griffin (2018) proposes a Bayesian non-parametric approach that allows for nonlinearity in the conditional mean, heteroskedasticity in the conditional variance, and non-Gaussian innovations. However, unlike the BNP-Lasso, it does not allow sparsity in the model. Petrova (2019) proposes yet another non-parametric, quasi-Bayesian likelihood estimation methodology for high dimensional setting with time varying parameters. The work in Iacopini and Rossini (ults) tackles the curse of dimensionality by a two-stage approach. In the first stage, a spike-and-slab prior distribution is used for each entry of the coefficient matrix which also identifies the interconnection network. In the second stage, it imposes prior dependence on the coefficients by specifying a Markov process for their random distribution. A closely related work is Bernardi and Costola (2019) that proposes a shrinkage and selection methodology designed for network inference in high-dimensional settings. It uses a regularized linear regression model with spike-and-slab prior on the parameters. However, both methods are limited to VAR models.

The rest of the paper is organized as follows. Section 2.2 reviews the notion of Granger causality and formally introduce directed information graphs which is suitable for linear and nonlinear systems. In Section 2.3, we introduce a novel approach for inferring the Granger-causal network of high dimensional nonlinear systems. Finally, in Section 2.4, we apply our method to learn the causal network of both synthetic and real-world dataset. For the real-world experiment, we used the daily stock prices of major US firms.

2.2 Causal Network

In this section, we present a statistical approach to learn the causal interconnections in a dynamical systems based on Granger causality Granger (1969). We begin by introducing some notations. Plain capital letters denote random variables or processes, while lowercase letters denote their realizations. Bold letters are used for column vectors, matrices, and tensors and calligraphy letters are used for sets. We use $X_{j,t}$ to denote the value of a time series X_j at time t and X_j^t to denote the time series X_j up to time t. For a set $\mathscr{A} = \{a_1, ..., a_n\}$ and an index set

$\mathscr{I} \subseteq \{1, ..., n\}$, we define $\mathscr{A}_{-\mathscr{I}} := \mathscr{A} \setminus \{a_i : i \in \mathscr{I}\}$.

2.2.1 Granger Causality

Researchers from different fields have developed various frameworks and graphical models to capture and represent interconnections among variables or processes. One of the most popular and widely used frameworks in economics is the notion of Granger causality. The basic idea in this framework was originally introduced by Wiener Wiener (1956), and later formalized by Granger Granger (1969). The idea is as follows: "we say that X is causing Y if we are better able to predict the future of Y using all available information than if the information apart from the past of X had been used."

Despite broad philosophical viewpoint of Granger (1963), his formulation for practical implementation was done using multivariate autoregressive (MVAR) models and linear regression which has been widely adopted in econometric and other disciplines. More precisely, in order to identify the influence of X_t on Y_t in a MVAR comprises of three time series $\{X, Y, Z\}$, Granger's idea is to compare the performance of two linear regressions: the first one predicts Y_t given $\{X^{t-1}, Y^{t-1}, Z^{t-1}\}$ and the second one predicts Y_t given $\{Y^{t-1}, Z^{t-1}\}$. Clearly, the performance of the second predictor is bounded by the first predictor. If they have the same performance, then we say X does not Granger cause Y. It is important to emphasize that this formulation is only applicable in linear systems.

To go beyond linear systems, works such as Quinn et al. (2015) and Massey (1990) use information-theoretical measures and generalize Granger causality. In this work, we introduce and apply directed information (DI) Quinn et al. (2015), an information-theoretical tools to measure interconnections among firms. DI has been used in many applications to infer causal relationships. For example, Quinn et al. (2011) and Kim et al. (2011) used it for analyzing neuroscience data and Etesami and Kiyavash (2014) and Etesami et al. (ults) applied to market data.

In order to visualize the inferred interconnections among time series using DI, directed information graphs (DIGs) have been developed Quinn et al. (2015). DIGs are a type of graphical models in which nodes represent time series and arrows indicate the direction of causation. We use DIG to represent the causal network among the covered firms.

2.2.2 Directed Information Graphs (DIGs)

In the rest of this section, we describe how the DI can capture the interconnections in causal[1] dynamical systems (linear or non-linear) and formally define DIGs.

Consider a dynamical system comprised of three time series $\{X, Y, Z\}$ that we assume they

[1] In causal systems, given the full past of the system, the present of the processes become statistically independent. In other words, there are no simulations relationships between the time series.

have a joint probability density function $p(X, Y, Z)$. To answer whether X has influence on Y or not over time horizon $[1, T]$, following the idea of Granger, we compare the average performance of two particular predictors over this time horizon. The first predictor uses the history of all three time series while the second one uses the history of all processes excluding process X. On average, the performance of the predictor with less information (the second one) is upper bounded by the performance of the predictor with more information (the first one). However, if the prediction of both predictors are close over time horizon $[1, T]$, it is an indication that X does not cause Y in this time horizon. To rigorously formalize this idea, we need the predictors and a measure to compare their performances.

In the definition of DI, the predictors belong to the space of probability measures. More precisely, the prediction of the first predictor at time t is $p(Y_t|Y^{t-1}, Z^{t-1}, X^{t-1})$ that is the conditional density function of Y_t given the history of all time series. Similarly, the prediction of the second predictor is $p(Y_t|Y^{t-1}, Z^{t-1})$ that is the conditional density function of Y_t given the history of all time series except time series X.

Given the predictions of the first and the second predictors at time t for an outcome $y_t \in \mathscr{Y}$, the goodness of these predictions are measured by the log-loss that are defined respectively by

$$-\log p(Y_t = y_t|Y^{t-1}, Z^{t-1}, X^{t-1}),$$
$$-\log p(Y_t = y_t|Y^{t-1}, Z^{t-1}).$$

According to the above measures of goodness, the better the predictor is, the smaller its log-loss will be. This loss function has meaningful information-theoretical interpretations. Namely, the log-loss is the Shannon's code length[2], i.e., the number of bits required to efficiently represent y_t.

At time t for an outcome $y_t \in \mathscr{Y}$, the difference between the log-losses of the two predictors compares their performances. This difference is also called *regret*,

$$r_t := -\log p(Y_t = y_t|Y^{t-1}, Z^{t-1}) - \big(-\log p(Y_t = y_t|Y^{t-1}, Z^{t-1}, X^{t-1})\big) \tag{2.1}$$
$$= \log \frac{p(Y_t = y_t|Y^{t-1}, Z^{t-1}, X^{t-1})}{p(Y_t = y_t|Y^{t-1}, Z^{t-1})}. \tag{2.2}$$

Note that the regrets are non-negative for all t and all outcomes y_t. The average regret over the time horizon $[1, T]$ is given by

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[r_t], \tag{2.3}$$

[2] It is also called the description length of y_t. For more information see Cover and Thomas (2012).

where the expectation is taken over the joint density function[3] of X, Y, and Z, i.e.,

$$\mathbb{E}[r_t] = \int p(y^t, z^{t-1}, x^{t-1}) \log \frac{p(y_t|y^{t-1}, z^{t-1}, x^{t-1})}{p(y_t|y^{t-1}, z^{t-1})} dy^t dx^{t-1} dz^{t-1}. \tag{2.4}$$

The average regret in (2.3) is called *directed information* (DI) and will be our measure of causation in this work. This measure is always positive and if it is zero, it indicates that the history of time series X contains no significant information that would help in predicting the future of time series Y given the history of Y and Z. This definition can be generalized to more than three time series as follows,

Definition 2.2.1. Consider a network of m time series $\mathscr{R} = \{R_1, ..., R_m\}$ with the joint probability density function p. The directed information from R_i to R_j over time horizon $[1, T]$ is given by

$$I(R_i \to R_j || \mathscr{R}_{-\{i,j\}}) := \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\log \frac{p(R_{j,t}|\mathscr{R}_{-\{i,j\}}^{t-1}, R_j^{t-1}, R_i^{t-1})}{p(R_{j,t}|\mathscr{R}_{-\{i,j\}}^{t-1}, R_j^{t-1})} \right], \tag{2.5}$$

where $\mathscr{R}_{-\{i,j\}}^{t-1} := \{R_1^{t-1}, ..., R_m^{t-1}\} \setminus \{R_i^{t-1}, R_j^{t-1}\}$. We declare R_i influences R_j over time horizon $[1, T]$, if and only if

$$I(R_i \to R_j || \mathscr{R}_{-\{i,j\}}) > 0. \tag{2.6}$$

An interpretation of R_i influencing R_j is that varying R_i will change the value of R_j even if all the other variables within the network remains unchanged. In another words, if R_i does not influence R_j, then varying R_i would not change R_j when the values of the remaining times series are fixed. This can be seen from the fact that DI compares two conditional distributions of R_j over a time horizon of length T; one is given the history of all the time series while the other one is given all the history except the history of R_i. Thus, if DI in (2.5) is zero, then these two conditional distributions are equal over this time horizon. This implies that the history of R_i does not contain any useful information for R_j.

It is important to emphasize that the definition of DI does not rely on any model assumption, thus DI is capable of inferring the causal relationships in general (linear or non-linear) dynamical systems. Next, we define the graphical model that we use in this work to visualize the causal network among firms.

Definition 2.2.2. Directed information graph (DIG) of a set of m time series $\mathscr{R} = \{R_1, ..., R_m\}$ is a directed graph $G = (\mathscr{V}, \mathscr{E})$, where nodes represent time series ($\mathscr{V} = \mathscr{R}$) and arrow $(R_i, R_j) \in \mathscr{E}$ denotes that R_i influences R_j.

A simple way to represent the DIG G of a dynamical system is via the adjacency matrix

[3] For the sake of notational simplicity, we use $p(y^t, z^{t-1}, x^{t-1})$ to denote $p(Y^t = y^t, Z^{t-1} = z^{t-1}, X^{t-1} = x^{t-1})$.

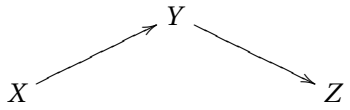


Figure 2.1: DIG of the system in (2.8)

DIG $= [d_{i,j}]_{m\times m}$ that is defined by

$$d_{j,i} = \begin{cases} 1 & \text{if } I(R_i \to R_j || \mathscr{R}_{-\{i,j\}}) > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

Given a DIG $G = (\mathscr{V}, \mathscr{E})$, we define the parent set of node R_j denoted by $\mathscr{PA}_j \subset \mathscr{V}$ to be the set of all times series that have direct influences on R_j, i.e., $\mathscr{PA}_j := \{R_k : d_{j,k} = 1\}$. Similarly, the children set of node R_j is given by $\mathscr{CH}_j := \{R_k : d_{k,j} = 1\}$. Next example demonstrates the DIG of a simple linear system.

Example 1. *Consider a network of three times series* $\{X, Y, Z\}$ *with the following linear dynamic,*

$$\begin{pmatrix} X_t \\ Y_t \\ Z_t \end{pmatrix} = \begin{pmatrix} 0.5 & 0 & 0 \\ 0.4 & 0.5 & 0 \\ 0 & -0.2 & 0 \end{pmatrix} \begin{pmatrix} X_{t-1} \\ Y_{t-1} \\ Z_{t-1} \end{pmatrix} + \begin{pmatrix} N_{X_t} \\ N_{Y_t} \\ N_{Z_t} \end{pmatrix}, \tag{2.8}$$

where N_X, N_Y, *and* N_Z *are three independent stationary Gaussian processes with zero mean and a diagonal covariance matrix (1, 0.9, 1). Since the dynamic is linear and the exogenous noises are Gaussian, we can compute the DIs using the following expression*[4] *Etesami and Kiyavash (2014).*

$$I(Z \to Y || X) = \frac{1}{2T} \sum_{t=1}^{T} \log \frac{|\Sigma_{Y_{t-1}, Y_t, X_{t-1}}||\Sigma_{Z_{t-1}, Y_{t-1}, X_{t-1}}|}{|\Sigma_{Y_{t-1}, X_{t-1}}||\Sigma_{Z_{t-1}, Y_{t-1}, Y_t, X_{t-1}}|}, \tag{2.9}$$

where $|\Sigma_{Y_{t-1}, Y_t, X_{t-1}}|$ *denotes the determinant of the covariance matrix of* $\{Y_{t-1}, Y_t, X_{t-1}\}$. *Using* (2.9), *we computed the DIs of this system,*

$$I(Y \to X||Z) = 0, \quad I(Z \to X||Y) = 0, \quad I(Z \to Y||X) = 0, \quad I(Z \to Z||X, Y) = 0,$$
$$I(X \to Z||Y) = 0, \quad I(X \to Y||Z) \approx 0.1, \quad I(Y \to Z||X) \approx 0.03.$$

Figure 2.1 illustrates the DIG of this system. In this example, $\mathscr{PA}_Z = \{Y\}$ *and* $\mathscr{CH}_Z = \{\}$.

Inference methods based on pairwise comparison has been developed and applied in the literature to identify the causal structure of time series. The methods in Billio et al. (2012), Billio et al. (2010), and Allen et al. (2010) are three such examples. However, pairwise comparison is not a correct approach in general and may fail to capture the true underlying network. For instance, considering the pairwise comparison in Example 1 between X and Z leads to a conclusion that X directly influences Z, which would be inaccurate. More precisely, without

[4] Equation (2.9) does not hold in general setting.

conditioning on Y, we obtain

$$I(X \to Z) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[\log \frac{p(Z_t|Z^{t-1}, X^{t-1})}{p(Z_t|Z^{t-1})}\right] \approx 0.002 > 0.$$

Notice that the DI in (2.5) is not a measure based on pairwise comparison. On the contrary, it measures the influence by conditioning on the remaining time series within the network.

Remark 1. *A causal model allows a factorization of the joint density function in some specific ways. It was shown in Quinn et al. (2015) that under a mild assumption, the joint density function of a causal discrete-time dynamical system with DIG $G(\mathscr{R},\mathscr{E})$ can be factorized as follows,*

$$p(R_1, ..., R_m) = \prod_{i=1}^{m}\prod_{t=1}^{T} p(R_{i,t}|\mathscr{R}^{t-1}_{\mathscr{P}_{\mathscr{A}_i\cup\{i\}}}). \tag{2.10}$$

Such factorization is called a generative model.

2.2.3 Inferring DIGs

Inferring the DIG of a dynamical system requires estimating the DIs between all ordered pairs of time series within that system. More precisely, inferring the DIG of a network of m time series requires computing $m(m-1)$ number of DIs. On the other hand, estimating DI requires estimating all the expectation terms in (2.5). In information theory this expectation is known as *conditional mutual information*[5], i.e.,

$$I(R_{j,t}; R_i^{t-1}|\mathscr{R}^{t-1}_{-\{i,j\}}, R_j^{t-1}) := \mathbb{E}\left[\log \frac{p(R_{j,t}|\mathscr{R}^{t-1}_{-\{i,j\}}, R_j^{t-1}, R_i^{t-1})}{p(R_{j,t}|\mathscr{R}^{t-1}_{-\{i,j\}}, R_j^{t-1})}\right]. \tag{2.11}$$

Using this notation, (2.5) can be written as follows

$$I(R_i \to R_j||\mathscr{R}_{-\{i,j\}}) = \frac{1}{T}\sum_{t=1}^{T} I(R_{j,t}; R_i^{t-1}|\mathscr{R}^{t-1}_{-\{i,j\}}, R_j^{t-1}), \tag{2.12}$$

Therefore, parametric and non-parametric estimators for the conditional mutual information can be used to estimate the DIs. There are different methods that can be used to estimate the terms in (2.12) given i.i.d. samples of the time series such as plug-in empirical estimator and k-nearest neighbor estimator. For an overview of such estimators see the Appendix and the articles in Paninski (2003), Noshad et al. (2019), and Jiao et al. (2013).

In general, estimating the DI in (2.12) is a complicated task and has high sample complexity. This is due to the fact that it requires estimating high dimensional conditional distributions. However, knowing some side information about the underlying dynamic can simplify the learning task of the DIG. For instance, in Example 1, since the underlying dynamic is linear with Gaussian exogenous noises, the DIs can be computed via the covariance matrices (2.9). Clearly,

[5] For more details see Cover and Thomas (2012).

the covariance matrix can be estimated with lower complexity compared to conditional mutual information. For our experimental results, we used (2.9) for the linear Gaussian experiment and the k-nearest method in Sricharan et al. (2011) for the non-linear experiment. The main reason for selecting k-nearest method is because it has shown relatively better performance compared to the other non-parametric estimators. For the sake of completeness, we describe the steps of this method in Appendix.

Side information can also help to infer the DIG of a dynamical system without directly estimating the DIs but instead it provides an alternative approach to identify the DIG. For example, if it is given that the underlying dynamic is linear, i.e., $\mathbf{X}_t = \mathbf{A}\mathbf{X}_{t-1} + \mathbf{N}_t$, then it has been shown in Etesami and Kiyavash (2014) that the support[6] of the coefficient matrix $\mathbf{A}$ is equal to the adjacency matrix of its corresponding DIG. This result implies that in linear systems, one can obtain the DIG by estimating the coefficient matrix. The latter problem has lower complexity and it can be done using e.g., linear regression. For similar examples in econometric models see Etesami et al. (ults).

2.2.4 DIG in High-dimensional Settings

For large networks with thousands nodes and millions of edges such as social or financial networks, DIGs become too complex to infer and analyze. The main reason is that without any side information, estimating the DI has high computational and sample complexity. Furthermore, the estimating complexity of DI increase with the dimension of the network. This is due to the fact that the DI in (2.5) measures the influence from R_i to R_j by conditioning on the information from the remaining network $\mathscr{R}_{-\{i,j\}}$. Therefore, the size of the conditioning set grows with the size of the network. This motivates the prior works to reduce the complexity of estimating DIs and thus make it more suitable for inferring the DIG of large networks by reducing the size of the conditioning set.

One such approaches is proposed by Quinn et al. (2013), in which they developed an efficient algorithm to identify the best directed tree approximation of a given network. This means reducing the size of the conditioning set to zero, i.e., no conditioning. However, this approach comes with the price of an approximation error and furthermore it fails to identify many interconnections between the processes.

The authors in Quinn et al. (2017) presented a more generalized version of the above approximation in which they identify the optimal connected bounded in-degree[7] approximations. This method reduces the size of the conditioning set in (2.5) to some constant value (bound of the in-degrees) which is independent of the network size. Although, this approach improves upon the approximation error but there is still a trade-off between the sample complexity

[6] The support of a matrix $\mathbf{B} = [b_{i,j}]$ is a binary matrix of the same dimension as $\mathbf{B}$ such that its entry (i, j) is one if and only if $b_{i,j} \neq 0$.

[7] Connected bounded in-degree graphs with bound k are connected directed graphs in which each node has at most k number of parents.

and the approximation error. In another words, as the in-degree bound increases, the sample complexity increases but the approximation error decreases.

In this work, we propose a new method that reduces the size of the conditioning set in (2.5) to only one for any given network while introducing less approximation error compared to the prior works. In this method, we estimate the directed information from R_i to R_j by conditioning on an auxiliary time series. This auxiliary time series is defined such that it comprises the information that the remaining of the network $\mathscr{R}_{-\{i,j\}}$ has about R_j. Next section explains this idea in more details.

2.3 Methodology

In order to present our method, we need the following preliminary result that characterizes an important property of DI in (2.5). All the proofs are presented in Appendix A.

Lemma 1. *Consider a network of m time series $\mathscr{R} = \{R_1, ..., R_m\}$ with corresponding DIG $G = (\mathscr{V}, \mathscr{E})$. Let $\mathscr{C}$ be a subset of $\mathscr{R}_{-\{i,j\}}$ such that $\mathscr{PA}_j \subseteq \mathscr{C}$. If $R_i \notin \mathscr{PA}_j$, then we have*

$$I(R_i \to R_j || \mathscr{C}) = 0. \tag{2.13}$$

Note that if $\mathscr{C} = \mathscr{R}_{-\{i,j\}}$ and R_i is not a parent of R_j, then by the definition of DIG, Equation (2.13) holds. On the other hand, this result states that to detect whether there is an influence from R_i to R_j in a network of time series, it suffices to find a subset of time series that either contains the parents of R_j or their information. In the remaining of this section, we first clarify the above statement via a simple linear system and later generalize it to non-linear models using neural networks.

Remark 2. *It is important to emphasize that the reverse of Lemma 1 does not hold. In another words, if there exists a subset $\mathscr{C} \subset \mathscr{R}_{-\{i,j\}}$ such that* (2.13) *holds, we cannot conclude that R_i has no direct influence on R_j.*

2.3.1 Linear Systems

Consider a first order vector autoregression model (VAR) with m time series,

$$\mathbf{X}_t = \mathbf{A}\mathbf{X}_{t-1} + \mathbf{N}_t, \tag{2.14}$$

where $\mathbf{X}_t, \mathbf{N}_t \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{m \times m}$, and $\mathbf{N}_t$ is a vector of m independent exogenous noises. As we discussed earlier, the result in Etesami and Kiyavash (2014) implies that the DIG of this VAR model is encoded in the support of its coefficient matrix $\mathbf{A} = [a_{i,j}]$, i.e.,

$$I(X_i \to X_j || \mathscr{X}_{-\{i,j\}}) = 0 \Longleftrightarrow a_{j,i} = 0. \tag{2.15}$$

In another words, the parents of time series X_j are the ones whose corresponding coefficients are non-zero in the j-th row of matrix **A**. This also can be seen from the j-th row of the matrix equation in (2.14),

$$X_{j,t} = \sum_{k=1}^{m} a_{j,k} X_{k,t-1} + N_{j,t}. \tag{2.16}$$

Another way to interpret the above equation is to say that the information of the network about time series X_j is in the form of a "portfolio", i.e., a linear combination of the other time series. Therefore, it is possible to summarize the network's information about X_j into only one time series, namely a well-designed portfolio. Next result shows the form of such portfolio.

Lemma 2. *In the linear system of* (2.14)*, X_i has no direct influence on X_j if and only if*

$$I(X_i \to X_j || Q) = 0, \tag{2.17}$$

where Q is a time series which we call the ideal portfolio and it is defined by $Q_{t-1} := \boldsymbol{u}_t^T \boldsymbol{X}_{-\{i\},t-1}$, where

$$\boldsymbol{u}_t := \arg\min_{\boldsymbol{w}\in\mathbb{R}^{m-1}} \mathbb{E}\left[||X_{j,t} - \boldsymbol{w}^T \boldsymbol{X}_{-\{i\},t-1}||_2^2\right],$$
$$\boldsymbol{X}_{-\{i\},t-1} := [X_{1,t-1}, \cdots, X_{i-1,t-1}, X_{i+1,t-1}, \cdots, X_{m,t-1}]^T.$$

According to the above Lemma, projecting X_j on $\mathscr{X}_{-\{i\}}$ results in an ideal portfolio Q that contains all the information for deciding whether there is an influence from X_i to X_j. Hence, instead of estimating $I(X_i \to X_j || \mathscr{X}_{-\{i,j\}})$ whose complexity depends on the network size, one can estimate $I(X_i \to X_j || Q)$. Note that the sample complexity of the latter DI does not grow with the size of the network and thus it is suitable for estimating the DIG of large networks.

2.3.2 Non-linear Systems with Additive Noise

Inferring the causal network of non-linear systems is a challenging problem that its complexity increases exponentially with the dimension of the network. In this section, we study the causal inference problem in non-linear systems whose dynamic can be captured by

$$X_{j,t} = F_j(\mathscr{X}^{t-1}) + \varepsilon_{j,t}, \quad j = 1, ..., m, \tag{2.18}$$

where $\mathscr{X}^{t-1} = \{X_1^{t-1}, ..., X_m^{t-1}\}$, $\{F_j(\cdot)\}$ is a set of non-linear continuous functions, and $\{\varepsilon_{j,t}\}$ is a set of independent exogenous noises. We call this model non-linear with *additive noise* due to the noise term that is added to the non-linear term[8]. This is a general non-linear dynamic that can be used to model the behavior of wide range of physical dynamical systems. The dynamic is called *Markovian* if $\mathscr{X}^{t-1}$ is replaced by $\mathscr{X}_{t-1} = \{X_{1,t-1}, ..., X_{m,t-1}\}$.

[8] In contrary to additive noise, there are systems in which the exogenous terms are multiplicative, e.g., $X_{j,t} = X_{i,t-1}\varepsilon_{j,t} + X_{j,t-1}$.

Below, we generalize the result of Lemma 2 to the non-linear system in (2.18) by showing that in such systems, it is possible to reduce the conditioning set in the DI to one time series.

Lemma 3. *In* (2.18)*, X_i has no direct influence on X_j if and only if*

$$I(X_i \to X_j || Q) = 0, \tag{2.19}$$

where Q is a time series defined by $Q_{t-1} := F_j(\mathscr{X}_{-\{i\}}^{t-1})$.

In the remaining of this section, we propose two methods to obtain the time series Q introduced in the above Lemma.

Koopman-based lifting technique Consider a particular sub-class of the non-linear system in (2.18) whose dynamic is defined by

$$F_j(\mathscr{X}^{t-1}) = \sum_{k=1}^{K} w_{j,k} h_k(\mathscr{X}_{t-1}), \;\; j = 1, ..., m, \tag{2.20}$$

where $\{w_{j,k} \in \mathbb{R}\}$ are the weights and $\{h_k(\cdot)\}$ denotes a set of library functions that are assumed to be known. This model is Markovian and the library functions can be seen as a set of basis that are used to approximated $F_j(\cdot)$. Examples of such library functions are monomials and Gaussian radial basis functions.

In this setting, the results of Lemma 3 implies that the following time series can be substituted in the conditioning of the DI.

$$Q_t = \sum_{k=1}^{K} w_{j,k} h_k(\mathscr{X}_{-\{i\},t-1}). \tag{2.21}$$

However, in this formulation, the weights $\{w_{,k}\}$ are unknown. An approach to obtain the weights is a non-linear filtering technique known as Koopman-based lifting Koopman (1931). This technique takes observational data and a set of library functions as inputs and obtains the unknown coefficients $\{w_{,k}\}$. The main steps of this technique are; transforming the data (lifting the data), applying a linear identification on the lifted data, and finally applying another transformation to bring down the results into the original vector field. Figure 2.2 illustrates the main steps. For more details see Appendix and Mauroy and Goncalves (2019).

Although, the Koopman-based lifting technique is theoretically sound but it has some shortcomings facing real-world applications. First, the Koopman's performance depends on the choice of the library functions and second, it often fails to estimate the real time series Q. More precisely, this technique involves the computation of matrix $\mathbf{L} := log(\mathbf{P}_x^{\dagger}\mathbf{P}_y)/T_s$, where $\mathbf{P}_x$ and $\mathbf{P}_y$ are estimated from the observational data[9]. Matrix $\mathbf{P}_x^{\dagger}$ denotes the pseudo-inverse, and the function $\log(\cdot)$ denotes the (principal) matrix logarithm. On the other hand, Koopman

[9] See Appendix for more details.

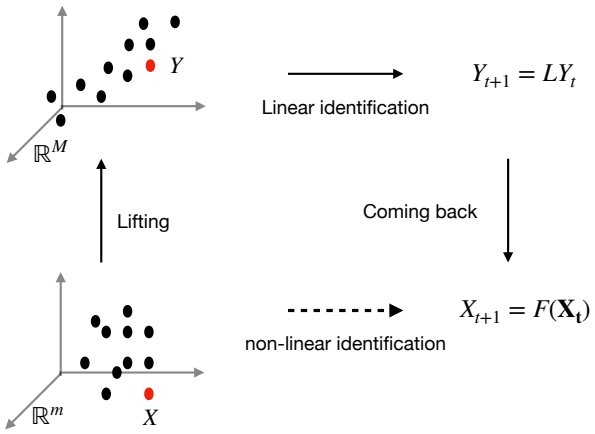


Figure 2.2: Koopman lifting technique compared to classical non-linear identification

Algorithm 1: Infer-DIG

Input: Observational data of m time series up to time T, $\mathscr{X}^T$, Threshold $\alpha > 0$;
Output: Adjacency matrix of **DIG** $= [d_{i,j}]$;
for $i, j = 1, ..., m$ **do**
  Train an RNN $R_j(\cdot;\Theta^*)$ that maps $\mathscr{X}^{t-1}_{-\{i\}}$ to $X_{j,t}$;
  Define $Q_{t-1} = R_j(\mathscr{X}^{t-1}_{-\{i\}};\Theta^*)$;
  if $I(X_i \to X_j||Q) > \alpha$ **then**
    $d_{j,i} = 1$
  else
    $d_{j,i} = 0$

lifting technique is applicable for estimating the time series Q only when the resulting matrix $\mathbf{L}$ is real[10]. However, this is not always the case in real-world applications due to observational noises and lack of sufficient data. To overcome such shortcomings, we propose an alternative approach to estimate Q using recurrent neural networks (RNNs).

RNNs method Recurrent neural networks are a specific class of Neural Networks well suited to learn time series. They are distinguished by their memory as they are able to remember information from prior inputs to influence their current outputs. The universal approximation theorem states that a neural network with enough hidden layers can approximate any non-linear continuous function such as $F_j(\cdot)$ in (2.18) (see Hornik et al. (1989)).

Given the aforementioned result, we train an RNN using the observational data to estimate the time series Q defined in Lemma 3. More precisely, our RNN maps $\mathscr{X}^{t-1}_{-\{i\}}$ as the inputs to $X_{j,t}$ as the output. Let $R_j(\mathscr{X}^{t-1}_{-\{i\}};\Theta^*)$ denotes the trained RNN with parameters Θ^*. In this

[10] See Culver (1966) for conditions under which a real matrix has a real logarithm.

case, the time series Q can be written as $Q_{t-1} = R_j(\mathscr{X}_{-\{i\}}^{t-1}; \Theta^*)$. Finally, we use (2.19) to detect whether X_i has influence on X_j or not. Algorithm Infer-DIG in 1 summarizes the steps of our RNN method.

2.4 Experimental Results

Since the true empirical DIG of firms is unknown, to evaluate the performance of our approach, we use different simulated environment. In this section, we first describe the simulation methodology in a linear Gaussian framework. We then show that our results generalize well to nonlinear setting by conducting an experiment on a nonlinear system. Finally, we apply our approach to a set of empirical data describing the daily stock prices of US firms and obtain their corresponding causal network.

2.4.1 Linear Gaussian Framework

In this experiment, we consider a linear system, a VAR(1) model whose dynamic is given by

$$\mathbf{X}_t = \mathbf{A}\mathbf{X}_{t-1} + \mathbf{N}_t \tag{2.22}$$

with m being the number of asset returns, $\mathbf{X}_t = (X_{1,t}, X_{2,t}, ..., X_{m,t})^\top$ being the vector of returns at time t, $\mathbf{A} = [a_{i,j}]$ being a $m \times m$ matrix and $\mathbf{N}_t$ being a $\mathcal{N}(0, \mathbf{I})$ vector of noises. As we discussed earlier, in such linear systems, $a_{i,j}$ captures the influence of asset j on asset i, i.e., there is an influence from j to i if and only if $a_{i,j} \neq 0$.

To reflect an important property of the market that some firms are more connected than others in our experiment, we divided the m time series into two parts. First part ($1 \leq i \leq s$) indicates assets with high degrees of connectedness and the second part ($1 + s \leq i \leq m$) are the ones with low degrees of connectedness. Parameter $1 < s < m$ denotes the numbers of assets with high degrees of connectedness. Afterward, for every entry (i, j) of $\mathbf{A}$, we independently generated a random number $x \sim U(-0.9, 0.9)$ and decided on value $a_{i,j}$ as follows,

$$a_{i,j} = \begin{cases} x1_{|x|>\underline{\epsilon}}, & \text{if } 1 \leq i \leq s, 1 \leq j \leq s, \\ x1_{|x|>\bar{\epsilon}}, & \text{if } 1 + s \leq i \leq m, 1 \leq j \leq m, \\ 0, & \text{if } 1 \leq i \leq s, 1 + s \leq j \leq m, \end{cases} \tag{2.23}$$

where $1_{a>b}$ denotes the indicator function which is equal to 1 when $a > b$ and 0 otherwise and $\underline{\epsilon}$ and $\bar{\epsilon}$ are thresholds to define non-zero entries in the upper-left and the lower part of $\mathbf{A}$, respectively. Figure 2.3 illustrates the structure of the resulting $\mathbf{A}$. We select these thresholds such that $\underline{\epsilon} < \bar{\epsilon}$. This ensures that the upper-left of $\mathbf{A}$ is denser than its lower part or equivalently, assets with indices $\{1, ..., s\}$ are more connected than the ones with indices $\{1 + s, ..., m\}$. In our experiment, we select $(s, m) = (85, 100)$ and $(\underline{\epsilon}, \bar{\epsilon}) = (0.4, 0.7)$. Finally, to guarantee the stability

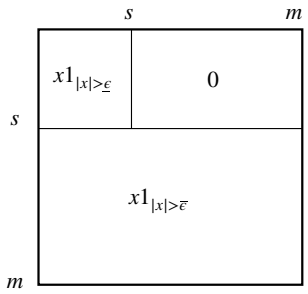


Figure 2.3: Structure of matrix **A** in (2.22) that is build using (2.23).

of the time series, we rescale[11] **A** such that its spectral radius is strictly less than one, i.e., $\rho(\mathbf{A}) < 1$. Once the matrix **A** is defined, we simulate the time series using (2.22) for a period of $T = 30000$ and use the resulting data for our estimations.

To study the effect of the conditioning set on detecting the influences, in our experiments, we consider four different conditioning sets. More precisely, to measure whether asset i influences asset j, we estimate $I(X_i \to X_j || \mathscr{C}_j)$ for the following choices of the conditioning set;

1. *True parents:* In this approach, we select $\mathscr{C}_j$ to be the true parents of X_j excluding X_i, i.e., $\mathscr{C}_j = \mathscr{PA}_j \setminus \{X_i\}$. Note that this approach is not practical[12] and we use it only as the benchmark to better understand the performances of the other approaches.

2. *Most correlated:* In this case, we define $\mathscr{C}_j$ to be the set of k most correlated assets with X_j (except X_i).

3. *Ideal portfolio:* In this scenario, $\mathscr{C}_j$ contains the portfolio Q, where Q is defined in Lemma 2. For further discussion see Appendix.

4. *RNN:* This method applies Algorithm 1 to estimate the time series Q and defines $\mathscr{C}_j = \{Q\}$.

Note that we also applied the Koopman-based lifting techniques but due to its mentioned shortcomings, it was unable to robustly identify the interconnections. Hence, we could not compare its performance with the other methods. In this experiment, since the dynamic is linear and the noises are Gaussian, we use Equation (2.9) to estimate the DIs. Finally, we obtain the adjacency matrix of the corresponding DIGs by comparing the estimated DIs with

[11] Formally, we use $\mathbf{A}/(\rho(\mathbf{A}) + \epsilon)$, where $0 < \epsilon$.

[12] This is because in structural learning problems, we do not know the true parents of each asset. In another words, if we had access to the true parents of each asset, we would have the DIG of the system and there is no need to compute the DIs.

a threshold $\alpha > 0$, i.e.,

$$[\mathbf{DIG}]_{j,i} = \begin{cases} 1 & \text{if } \widehat{I}(X_i \to X_j || \mathscr{C}_j) > \alpha, \\ 0, & \text{otherwise,} \end{cases} \tag{2.24}$$

where $\widehat{I}(\cdot \to \cdot || \cdot)$ denotes the estimated DI from the data. In order to compare the performances of the aforementioned four approaches, we use the precision and recall measure between the true DIG (obtained from $\mathbf{A}$) and their estimated DIGs. Formally, the precision and the recall are defined by

$$Precision := \frac{TP}{TP+FP}, \ Recall := \frac{TP}{TP+FN},$$

where

$$TP := \sum_{i,j=1}^{m} 1_{a_{j,i}\neq 0} 1_{[\mathbf{DIG}]_{j,i}\neq 0}, \ FP := \sum_{i,j=1}^{m} 1_{a_{j,i}=0} 1_{[\mathbf{DIG}]_{j,i}\neq 0},$$
$$FN := \sum_{i,j=1}^{m} 1_{a_{j,i}\neq 0} 1_{[\mathbf{DIG}]_{j,i}=0}.$$

Figure 2.4 shows the performances of the four aforementioned approaches in the linear framework. It is not surprising that the *true parents* approach achieves 100% accuracy, as it is anticipated by Lemma 1. The *ideal portfolio*'s performance is guaranteed by Lemma 2 and it is verified by our experiment. However, it is important to emphasize that the *ideal portfolio* shows ideal performance because the underlying model is linear. As we will see in the next section, its performance declines when the underlying model deviates from being linear. For the *most correlated* approach, we used $k = 10$ but as it is shown in Figure 2.4, it has the worst performance among the four conditioning methods. This is due to the fact that the set of the ten most correlated assets with a given asset j does not necessarily contains the true parents of asset j. On the other hand, we observe high accuracy from the *RNN* approach which is a striking result. This result is an evidence that an RNN is capable of estimating the ideal portfolio, i.e., the time series Q in Lemma 2 without any side information about the underlying model.

2.4.2 Non-Linear framework

To compare the performances of the different approaches from the previous section in non-linear environment, we simulate a set of quadratic processes whose dynamic is given below,

$$X_{i,t} = b_i \mathbf{X}_{t-1}^T \mathbf{A}_i \mathbf{X}_{t-1} + N_{i,t}, \ i = 1, ..., m, \tag{2.25}$$

where $\mathbf{A}_i \in \mathbb{R}^{m\times m}$, $\mathbf{X}_t = (X_{1,t}, X_{2,t}, ..., X_{m,t})^T$, $N_{i,t} \sim \mathscr{N}(0, \sigma^2)$, and $b_i \sim U(-0.9, 0.9)$. Note that the term $|[\mathbf{A}_i]_{j,k} + [\mathbf{A}_i]_{k,j}|$ captures the effect of $X_{j,t-1}X_{k,t-1}$ on $X_{i,t}$. Thus, it is possible to

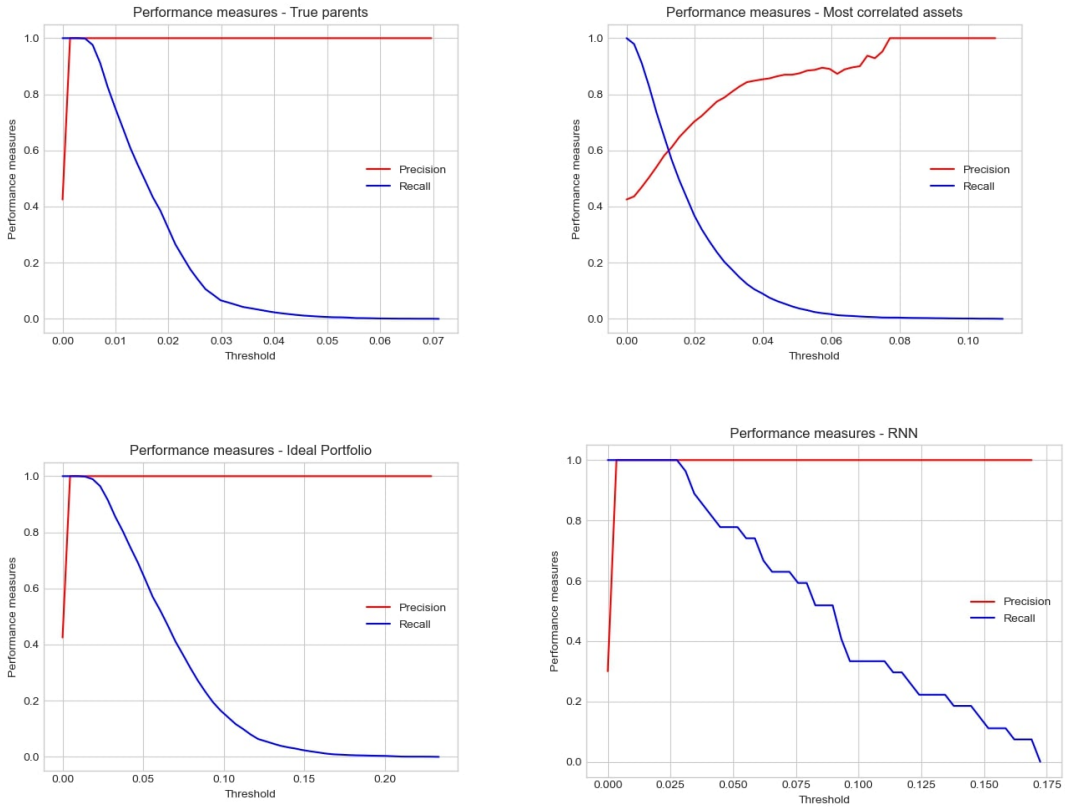


Figure 2.4: Precision and recall curves in the linear framework.
Precision and recall curves for the *True parents*, *Most correlated*, *Ideal portfolio*, and the *RNN*, respectively.

obtain the true parents of asset i as follows,

$$\mathscr{PA}_i = \{X_j : [\mathbf{1}^T \cdot \left(|\mathbf{A}_i^T + \mathbf{A}_i|\right)]_j > 0\}, \tag{2.26}$$

where $\mathbf{1}$ denotes all-one vector of length m. Each matrix $\mathbf{A}_i$ is simulated independently by following the similar procedure as in Section 2.4.1. In this experiment, since the model is non-linear, we could not apply (2.9) to estimate the DIs but instead we used the k-nearest method to estimate the mutual information and applied Equation (2.12).

Herein, we again compare the performances of the four different conditioning approaches. Figure 2.5 shows the precision-recall curves for these approaches in the quadratic model with $m = 15$. Precision-recall curves are a standard tools to illustrate and compare the performances of different learning methods. In this curve the precision is demonstrated in the y-axis vs. the recall on the x-axis for all potential values of the threshold α.

Similar to the linear setting, we use the *true parents* as a benchmark since it has the ideal performance. It is however important to emphasize that this conditioning approach has higher complexity compared to the others. This is because in the *true parent* approach, the size of the conditioning set is relatively larger than the other approaches.

For the *most correlated* approach, we use $k = 5$, i.e., the size of the conditioning set is five. With this method, we could slightly reduce the estimation complexity of the DIs compared to the *true parent* approach but this comes with the price of losing the performance. clearly, the performance of the *most correlated* approach can be improved by increasing k but this will increase the complexity.

The performance of the *ideal portfolio* approach (using the time series in Lemma 2 as the conditioning) is worse than all others which is not surprising as the model is no longer linear. This means that the information embedded in the linear portfolio is not sufficient to decide the non-linear influences among the time series.

Finally, as it is shown in Figure 2.5, the *RNN* approach outperforms the *most correlated* and the *ideal portfolio* approaches and it shows close performance to the *true parents* but with the size of the conditioning set equal to one. This result once more fortifies our claim that with an RNN we can summarize the information of the network into one time series and use it for detecting the causal relationships. This claim is due to Lemma 3 and the universal approximation theorem which states that a neural network with enough hidden layers can approximate any non-linear function Hornik et al. (1989). The slight difference between the performance of the *RNN* and the *true Parents* is because of the estimation error in the recurrent neural network.

2.4.3 Empirical DIG

This section describes how to apply our approach to empirical data and obtain the DIG of some US firms. We extracted the daily stock prices and the daily US Treasury rate as risk free

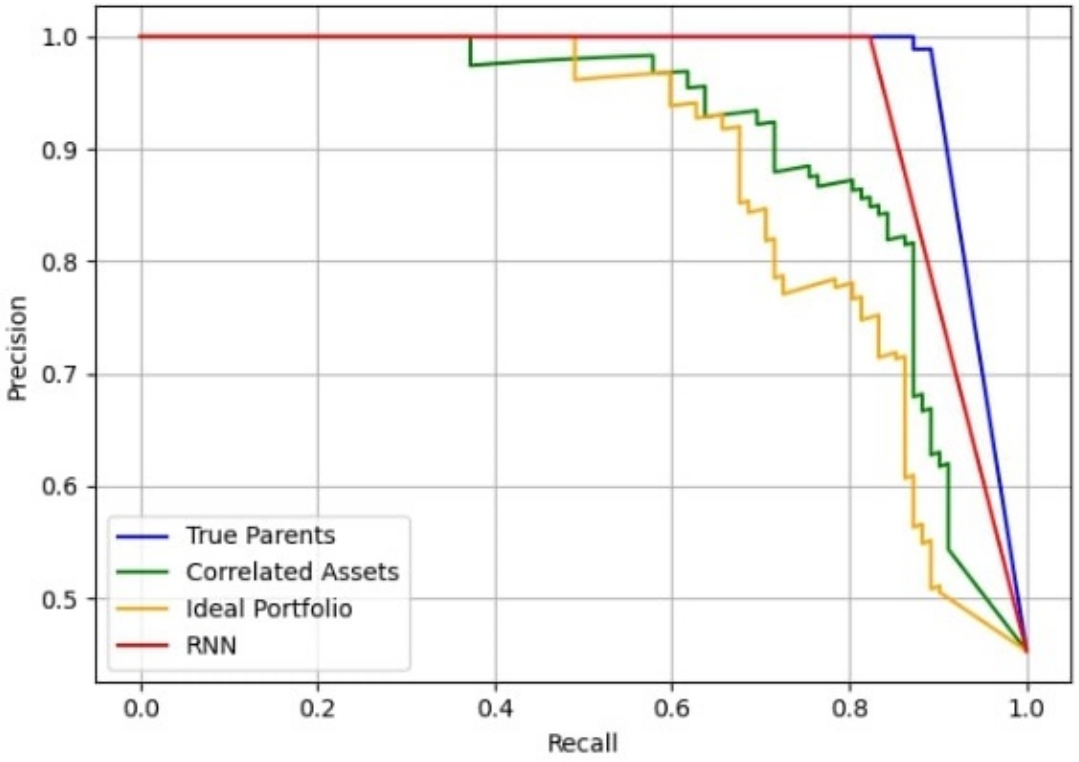


Figure 2.5: Precision-Recall curves for the quadratic model.

returns from the CRSP database from 1990 to 2020. As the market is likely to evolve through these years, we chose to divide the dataset into six subsets, each of which has a length of five years and estimate the corresponding DIG of each subset separately. Herein, we assume that the causal structure of the market evolved but its rate was slow enough such that during a period of five years, the DIG of the market remained unchanged.

For every subset, we keep the data of the 1000 firms with the highest maximum market capitalization and compute their excess return time series $X_{i,t}$, using the following relationship,

$$X_{i,t} = ln(P_{i,t}) - ln(P_{i,t-1}) - r_t, \tag{2.27}$$

where $P_{i,t}$ denotes the stock price of the firm i at time t and r_t is the risk free rate at time t. Afterwards, we apply Algorithm 1 with the excess returns as the input to estimate the corresponding DIG of each subset. We use the k-nearest neighbor method to estimate the DIs. We define the threshold α to be the unconditional mean across the estimated DIs. Note that in this experiment, the true DIGs of the market are not known, hence, we could not compute the precision-recall curves.

For the sake of presentation, instead of the complete DIGs with 1000 nodes, we draw the sub-graphs consisting of the 30 largest firms in Figures[13] 2.6, 2.7, and 2.8 . Each graph consists of 60 nodes illustrating the cause firm on the top hemisphere and the effect firm on the bottom hemisphere. For instance, if there is an edge between "from: AAPL" on the top and "to: GOOGL" on the bottom, it means that Apple influences Google. The dynamic evolution of the DIGs through time can often be explained by real events that happened in the market. For instance, in the DIG 2010-2014, Apple was not influencing General Electric (GE). However, on the 17th October 2017, Apple announced a partnership with GE to bring Predix, GE's data and analytics platform, to their iPhones and iPads. We are able to capture this partnership in the DIG 2015-2019 as an edge is now present from Apple to GE. Another example is the announced collaboration between AT&T and Cisco to manage IoT devices and launch 5G service at the end of the 2010s: there was neither an edge from AT&T to Cisco nor from Cisco to AT&T during the first half of the 2010s, but the DIG for the second half of the 2010s shows a mutual influence, reflecting an increased relationship between the two companies.

Table 2.1 shows the Degree of Granger Causality (DGC) defined as the fraction of relationships in the network among all potential relationships. Formally,

$$DGC = \frac{1}{N^2} \sum_i \sum_j [\mathbf{DIG}]_{j,i}, \tag{2.28}$$

These results show that the DGC increased both in the DotCom bubble and in the Subprime Crisis, suggesting an increase of the connectedness in turmoil periods. This finding is consistent with Longin and Solnik (2001) stating that correlation increases in bear markets.

[13] For a better presentation, interactive plots are available at https://marcaureledivernois.github.io/firm-network/

1990-1994	1995-1999	2000-2004	2005-2009	2010-2014	2015-2019
0.23	0.27	0.18	0.28	0.32	0.25

Table 2.1: Degree of Granger Causality (DGC) for each sub-graph.
DGC is defined as the fraction of relationships in the network among all potential relationships.

Tables 2.2 and 2.3 show the outdegree and indegree of every firm in the six subsets. Outdegree is defined as the number of edges going out of a specific node. Indegree is the number of edges going to a specific node. These tables also reveal interesting facts. For instance, the SPY ticker, an ETF launched in 1993 and aiming at tracking the S&P500 return, enters in the 30 biggest market capitalizations in 2010 and has the highest number of outdegrees in the periods 2010-2014 and 2015-2019 but relatively low number of indegrees. This result suggests that the market return is influencing a high number of firms, but the converse is not necessarily true. One way to intuitively interpret this finding is that bullish and/or bearish markets are likely to influence next period stock returns (as a persistence effect known in business cycles), but individual stock returns struggle at predicting next period market returns.

2.5 Conclusion

In this work, we introduce an information-theoretic measure known as directed information that is capable of capturing nonlinear Granger-causality in an interactive system. We develop a novel algorithm based on recurrent neural network utilized with directed information. This algorithm can infer the interconnections within a large network with less complexity than previous works. As a proof of concept, we show that our approach performs well both in a linear and in a non-linear simulated environments. Finally, we apply this algorithm to infer the causal relationships among the major US firms during 1990 to 2020.

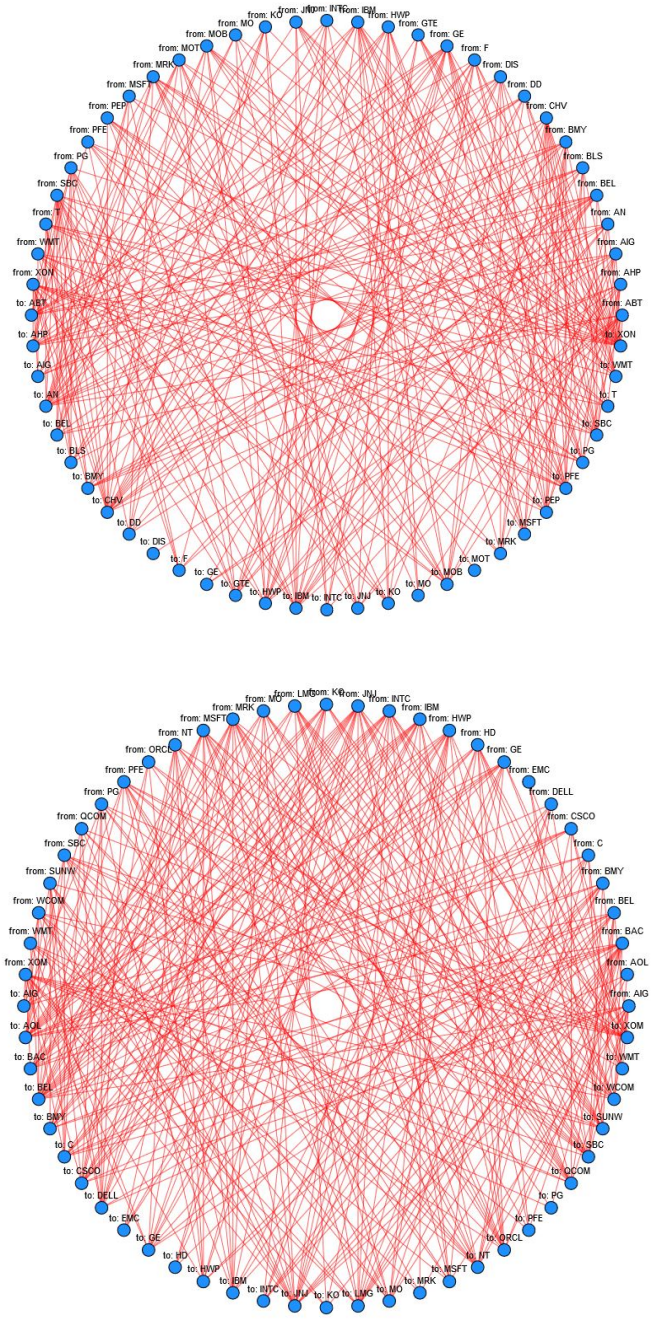


Figure 2.6: Empirical DIG for the periods 1990-1994 and 1995-1999.
Empirical DIG for the periods 1990-1994 (top) and 1995-1999 (bottom). Interactive graphs can be found at https://marcaureledivernois.github.io/firm-network/

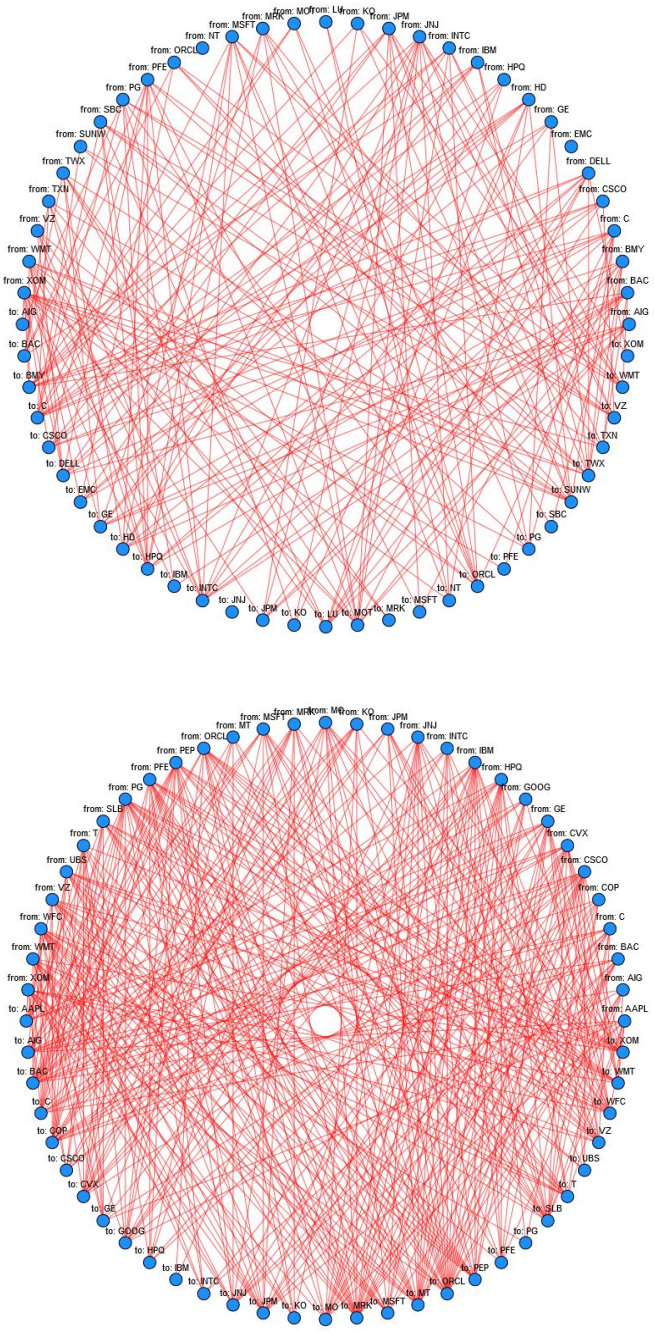


Figure 2.7: Empirical DIG for the periods 2000-2004 and 2005-2009.
Empirical DIG for the periods 2000-2004 (top) and 2005-2009 (bottom). Interactive graphs can be found at https://marcaureledivernois.github.io/firm-network/

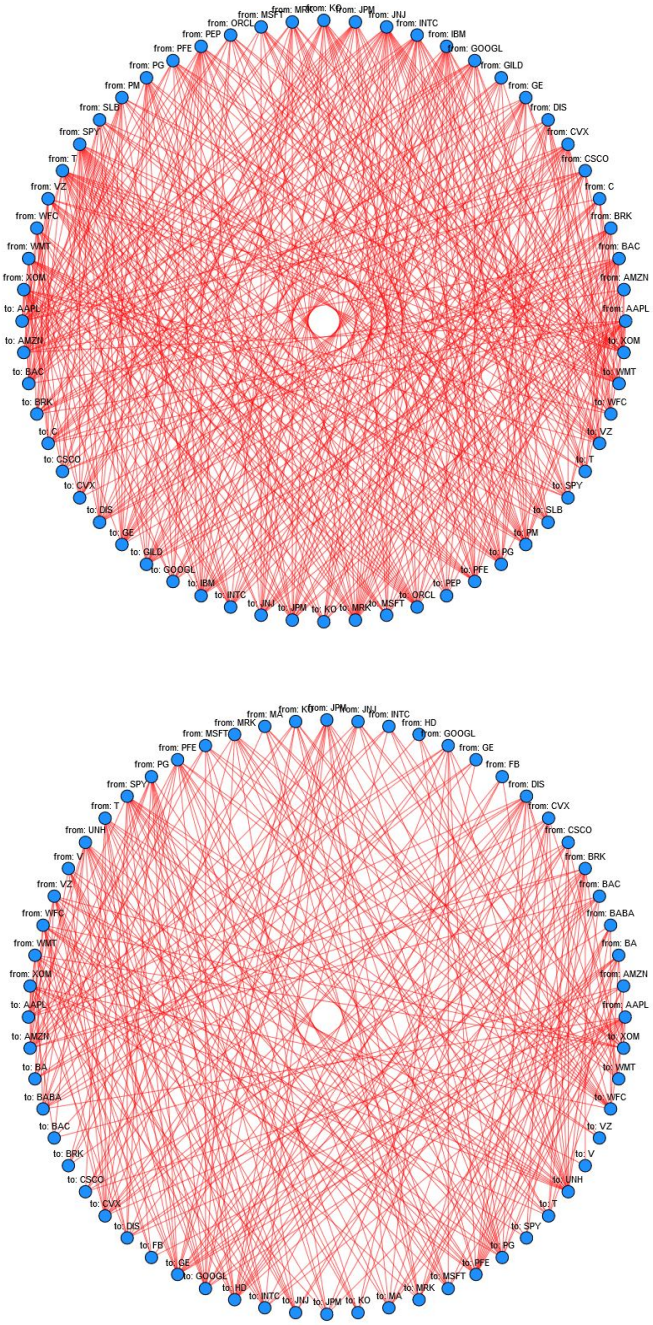


Figure 2.8: Empirical DIG for the periods 2010-2014 and 2015-2019.
Empirical DIG for the periods 2010-2014 (top) and 2015-2019 (bottom). Interactive graphs can be found at https://marcaureledivernois.github.io/firm-network/

1990-1994		1995-1999		2000-2004		2005-2009		2010-2014		2015-2019	
Ticker	Out	Ticker	Out	Ticker	Out	Ticker	Out	Ticker	Out	Ticker	Out
SBC	13	JNJ	12	XOM	12	PEP	15	SPY	17	SPY	14
CHV	11	HWP	12	JNJ	10	WFC	13	T	15	DIS	12
XON	10	INTC	12	MSFT	9	CSCO	12	JNJ	15	BRK	12
PG	10	BAC	12	DELL	9	WMT	11	CVX	13	PFE	11
GE	10	PFE	11	WMT	9	IBM	11	AAPL	12	UNH	11
AN	9	IBM	11	C	9	PG	11	INTC	12	JPM	11
BEL	9	MSFT	11	HD	8	CVX	11	CSCO	12	AAPL	10
MRK	8	SBC	11	JPM	8	HPQ	10	GE	12	GOOGL	10
IBM	8	GE	10	PFE	8	VZ	10	IBM	11	WMT	9
BMY	8	MRK	10	PG	7	GE	9	GOOGL	11	CSCO	9
ABT	7	LMG	9	INTC	7	UBS	9	XOM	11	VZ	9
AHP	7	MO	9	BAC	7	PFE	9	JPM	10	BA	9
T	7	NT	9	BMY	6	C	9	KO	10	CVX	8
BLS	7	XOM	9	SBC	6	SLB	9	MRK	10	WFC	8
AIG	7	KO	9	IBM	5	MSFT	9	BRK	10	V	8
F	7	HD	8	TWX	5	ORCL	9	SLB	10	PG	8
HWP	7	BEL	8	AIG	5	MRK	8	WMT	10	T	7
MOT	7	BMY	8	CSCO	4	KO	8	VZ	9	JNJ	7
DD	6	SUNW	8	GE	4	GOOG	7	DIS	9	MRK	7
DIS	6	AIG	7	VZ	4	XOM	7	PFE	9	KO	6
MOB	6	WCOM	7	MRK	4	JNJ	7	BAC	8	XOM	5
PEP	6	CSCO	6	MOT	4	MO	7	PM	8	MSFT	5
MSFT	6	QCOM	6	TXN	4	JPM	6	ORCL	7	MA	5
INTC	6	C	6	HPQ	3	COP	6	PG	7	BABA	5
WMT	6	PG	6	ORCL	3	T	6	PEP	6	FB	4
GTE	5	DELL	5	KO	2	INTC	6	MSFT	6	BAC	4
MO	5	ORCL	4	SUNW	1	AAPL	6	C	5	INTC	4
JNJ	4	WMT	4	LU	1	BAC	5	WFC	5	GE	3
PFE	4	EMC	2	EMC	0	MT	4	AMZN	5	HD	3
KO	2	AOL	2	NT	0	AIG	4	GILD	5	AMZN	2

Table 2.2: Outdegrees ranked for each sub-graph.
Outdegree (Out) is defined as the number of edges going out of a specific node.

1990-1994		1995-1999		2000-2004		2005-2009		2010-2014		2015-2019	
Tic	In	Tic	In	Tic	In	Tic	In	Tic	In	Tic	In
XON	22	BEL	15	MOT	12	MRK	19	MRK	18	UNH	17
CHV	17	JNJ	15	INTC	11	ORCL	19	PM	15	PFE	16
AN	13	AOL	14	C	11	BAC	16	JPM	15	HD	14
SBC	12	XOM	14	SUNW	11	SLB	14	ORCL	14	AMZN	14
HWP	11	NT	12	ORCL	10	WFC	13	IBM	14	GOOGL	12
PEP	10	QCOM	12	BMY	9	JPM	12	XOM	13	GE	12
AHP	10	SUNW	11	DELL	9	COP	12	PG	13	PG	12
MOB	10	ORCL	11	TWX	7	MT	11	JNJ	12	CVX	11
GTE	9	CSCO	11	HPQ	7	PEP	10	DIS	12	BABA	9
IBM	9	C	11	GE	7	C	10	INTC	11	MRK	9
BMY	8	DELL	10	CSCO	7	MO	10	WFC	10	DIS	9
ABT	8	SBC	10	LU	6	CVX	10	C	10	WFC	8
KO	7	BMY	9	HD	6	AIG	9	VZ	10	BA	7
BEL	7	WCOM	9	TXN	6	T	9	WMT	10	MA	7
MSFT	7	GE	9	EMC	6	JNJ	8	GE	10	AAPL	7
BLS	6	IBM	9	BAC	5	GOOG	7	KO	9	T	7
PFE	6	MO	9	PG	5	KO	7	BAC	9	WMT	6
DD	6	LMG	8	WMT	5	VZ	6	PFE	9	MSFT	6
JNJ	5	HWP	6	VZ	5	MSFT	6	AAPL	9	INTC	6
WMT	4	MSFT	6	JPM	4	INTC	6	SPY	8	FB	5
F	4	BAC	5	KO	3	AAPL	6	CVX	7	KO	5
MRK	4	INTC	5	NT	3	GE	6	AMZN	7	JPM	4
PG	4	WMT	5	XOM	2	WMT	6	GILD	7	XOM	4
AIG	4	PFE	4	MSFT	2	XOM	5	CSCO	7	JNJ	4
T	3	AIG	4	AIG	2	PFE	5	GOOGL	7	SPY	4
GE	2	KO	3	MRK	1	CSCO	4	MSFT	6	BRK	4
MO	2	MRK	2	PFE	1	HPQ	4	PEP	5	CSCO	3
INTC	2	PG	2	SBC	1	UBS	2	SLB	5	BAC	2
DIS	1	HD	2	IBM	0	IBM	1	T	4	V	1
MOT	1	EMC	1	JNJ	0	PG	1	BRK	4	VZ	1

Table 2.3: Indegrees ranked for each sub-graph.
Indegree (In) is defined as the number of edges going to a specific node.

3 StockTwits Classified Sentiment and Stock Returns

3.1 Introduction

Recent developments in artificial intelligence and the growing amount of alternative data have created new areas of research in finance. In particular, often coming from news, social media or annual reports, textual data is increasingly used in the literature. Nikfarjam et al. (2010) documents recent text mining approaches for stock market prediction. The pioneer work from Antweiler and Frank (2004) computes a bullishness measure out of 1.5 million messages posted on Yahoo! Finance and Raging Bull and finds that stock messages help predict market volatility. Their results clearly reject the hypothesis that all that talk is just noise. They show that there is financially relevant information present. Growing from that, there are now two main sides of sentiment analysis literature studying the forecasting abilities of natural language processing (NLP hereafter): text mining of annual reports and sentiment analysis using social media. On one side, Shirata et al. (2011) and Cecchini et al. (2010) report that extracting phrases from annual reports may be an effective predictor of corporate bankruptcy. Similarly, Loughran and McDonald (2011a) identify phrases that might be red flags indicating questionable behavior. On the other side, Renault (2017) builds an intraday investor sentiment indicator using messages and finds that the change in investor sentiment of the first half-hour of a trading day helps forecast the last half-hour market return of that trading day.

We conjecture that our investor sentiment measure can serve as a proxy of unobservable firm fundamentals. For instance, misaligned managerial and shareholder incentives are not easy to observe quantitatively in firm's annual reports (see Nikolov and Whited (2014)) but analysts may talk about them freely in bearish messages. This flow of data can be used throughout a period to improve forecasts (see *nowcasting* in Challet and Ayed (2013)). Tetlock et al. (2008) uses Wall Street Journal stories to examine whether the usage of language is able to predict individual firms' accounting earnings and stock returns. They find that some aspects of firms' fundamentals are hard-to-quantify, but investors may use linguistic media content to capture information and incorporate it into stock prices.

Most papers studying social media's predictive power use Twitter as their primary source of

data. Twitter has the advantage of being used by a wide range of people across the world and a few influencers can attract attention of many investors. In 2013, Carl Icahn tweeted that following a meeting with Tim Cook (Apple CEO), he bought a large position in Apple and believed that the company is extremely undervalued. This bullish tweet caused the market capitalization of Apple to jump by $12 billion. In 2019, JPMorgan has created the *Volfefe Index* to track Donald Trump's tweets impact on the stock market. However, it is more difficult to disentangle noise from relevant tweets in Twitter than in other more focused social media. Results from Ghoshal and Roberts (2016) show that StockTwits is significantly more informative than Twitter data. This is not surprising as StockTwits is a finance-only social media whereas Twitter also captures irrelevant opinions on a wide range of non-finance related matters.

To the best of our knowledge, our paper is the first that compares predictive power of messages in general and around specific events. Ranco et al. (2015) is likely the closest paper to our study. However, the finance-tailored data we use allow us to get higher contemporaneous correlations between stock returns and polarity. We also use much more messages (90 million versus 1 million) during a longer period (10 years versus 13 months). We also add contribution by looking at the predictive power of cumulative average abnormal polarity (CAAP hereafter). Our paper also contributes to the Efficient Market Hypothesis (EMH hereafter) literature by gauging how cumulative average abnormal returns (CAAR hereafter) and CAAP behave around sudden peaks of message activity. We collect 90 million messages published on StockTwits from mid-2010 to the end of March 2020. On this microblogging platform, users are invited to identify firms with their ticker when they share opinions. Another useful feature on the platform is the possibility to explicitly label their own messages as *bearish* or *bullish* when users post them. We believe that messages on the platform are reliable for several reasons. First, users have incentives to publish valuable information in order to maintain or increase mentions and/or retweets and thus have a greater share of voice in the forum (Sprenger et al. (2014)). In addition, market manipulations happen rarely because malicious users have incentives to post fake news only if they previously traded in the same direction than the news they are creating, which will only benefit them if they already have influence on the platform. Finally, SEC closely monitors large influencers to prevent any market abuse.

The challenge in this context is to create a classifier that understands the vocabulary of the messages posted by the investors. For instance, "bull" is an animal in everyday language but it is someone optimistic in the financial jargon. Work has already been done in this direction: Loughran and McDonald (2011b) creates a word list, which helps classify tone in a financial document. However, this might not be sufficient in the context of social media because messages posted present many typos, abbreviations and slang, so one needs to have an additional layer of data preprocessing. For instance, the word "goooooood" would not be recognized by the model if it is not corrected into "good" first.

In this paper, we use a logistic regression on Term Frequency-Inverse Document Frequency (TFIDF hereafter) vectorized labeled messages to classify unlabeled messages in either bearish, bullish or neutral class. TFIDF is a weighting scheme gauging the importance of a word in a

document (see Erdemlioglu et al. (2017)). As users are prone to post more bullish messages than bearish messages, a good classifier needs to take into account the unbalanced data. Without resampling, the classifier outputs classification scores biased towards the over-represented class. Because we are creating an artificial neutral class, we chose to not resample the data but rather optimally select classification score thresholds by maximizing F1 scores of two distinct classification algorithms: bullish versus non-bullish and bearish versus non-bearish. Then, we aggregate messages daily for every firm to compute sentiment polarity time series for individual firms and for the whole economy.

We then use the daily volume of messages on a given firm to identify sudden peak of activity, indicating a firm event. Using abnormal polarity on each event date, we are able to classify events into three classes: bullish, neutral and bearish. We then compute cumulative average abnormal return and cumulative average abnormal polarity in a 41 days window centered at the identified event. We show that abnormal polarities have significantly higher predictive power than abnormal returns. On average, changes of polarity are associated with changes of contemporaneous return of the same sign; but this result does not hold against next-day return. However, when we focus on specific events, polarity has strong predictive power. It is also interesting to note that polarity tends to be biased towards recent past events. Finally, as robustness check, our event study on CAAR is similar to previous literature and is consistent with Fama's theory.

The remainder of the paper is structured as follows. Section 3.2 presents the data. Section 3.3 develops the NLP logistic classifier on TFIDF vectorization. Section 3.4 explains our polarity measure. Section 3.5 contains the results of the event studies. Section 3.6 shows the portfolio construction and Section 3.7 concludes.

3.2 Data

Data is coming from two sources: Compustat/CRSP for the stock prices, StockTwits for the investor sentiment. From the CRSP database, we extract daily stock prices (closing prices), daily volume of transactions and number of shares outstanding from 2010 to 2020 for all US and Canadian firms. Stock prices and number of shares are adjusted to account for any distribution (i.e. dividends, stock splits) so that a comparison can be made on an equivalent basis before and after the distribution.

StockTwits is a large social network similar to Twitter but designed for investors and traders. Users register online and can post messages about any listed firm through the prefix $ followed by the ticker of the firm. StockTwits was created in 2008 as an app built on the Twitter's API and later detached from Twitter to build a standalone social network. As of April 2019, it has over two million registered users and the number of messages posted is growing exponentially (see Figure 3.1). StockTwits describes itself as "the voice of social finance and the best way to find out what is happening right now in the markets and stocks you care about". In practice, it is effectively used by finance professionals to express their opinions on individual firms

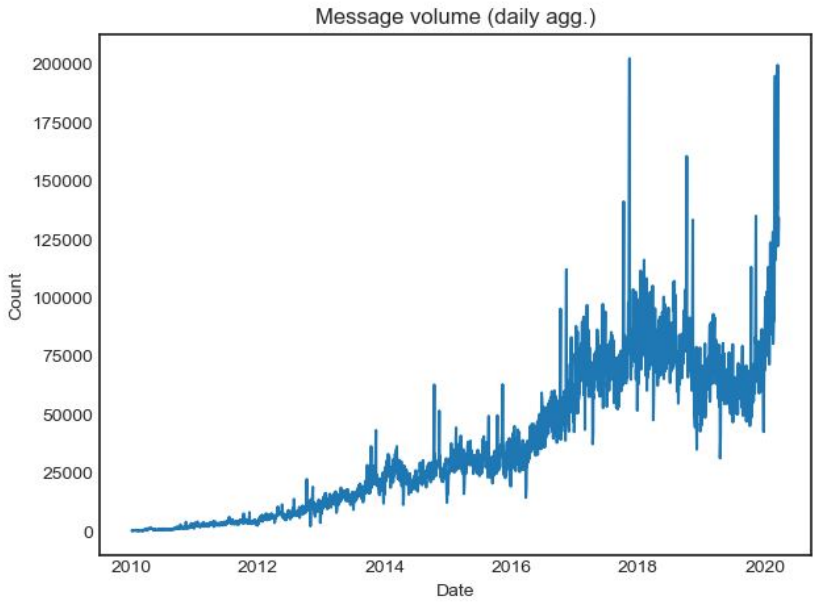


Figure 3.1: Number of messages posted daily on StockTwits.

and the market as a whole. Since mid-2010, users also have the possibility to label their own messages as either *bullish* or *bearish*. This feature is very useful for researchers as even though many messages are unlabeled, it allows for sentiment classification using NLP techniques. The reasons of using StockTwits and not another social media is threefold. First, one of the main challenge in Natural Language Processing is the creation of an appropriate labeled vocabulary. Loughran and McDonald (2011a)) shows that it is essential to have a specific vocabulary to interpret finance documents (i.e many words have a different meaning in finance than in traditional English (e.g "bear trap")). In addition to that, social media slang is an additional layer of language complexity. To this extent, the functionality to self-tag *bullish* and *bearish* messages that StockTwits implemented in 2012 is very valuable as it allows the creation of a specific labeled vocabulary out of labeled messages. We are not aware of any other social media platform in finance offering this functionality. Second, StockTwits is less noisy than Twitter because messages focus on finance and economics matters only. Third, extracting data out of StockTwits is easy because of its API. StockTwits' API is designed to query the database to download messages via JSON requests. Using a Python script, we extract all messages since the 10th of July 2009 until the 31st of March 2020 for a preset list of tickers coming from the Compustat database corresponding to all US and Canadian firms. This results in 90 million messages, which we download and store as JSON files. As one message may refer to several tickers, we consider a message with two or more tickers as one message for every ticker identified. We refer to the appendix for more information about this process.

Every message comes with the following eight features: (1) the ticker discussed (2) the exact timestamp of the message, (3) a unique message identifier, (4) the body of the message, (5)

the sentiment label (bearish, bullish, or none) entered by the user, (6) a unique identifier of the user who sent the message, (7) the number of messages published by the user who sent the message, and (8) the number of followers of the user who sent the message. Figure 3.2 shows a screenshot of the StockTwits website as of 3rd March 2020, for a query on the AAPL[1] ticker. The first message is labeled as bullish by the user “satkaru", the two next are unlabeled messages that will be classified using a machine learning classifier, and the last message is labeled as bearish by the user “Etrading”.

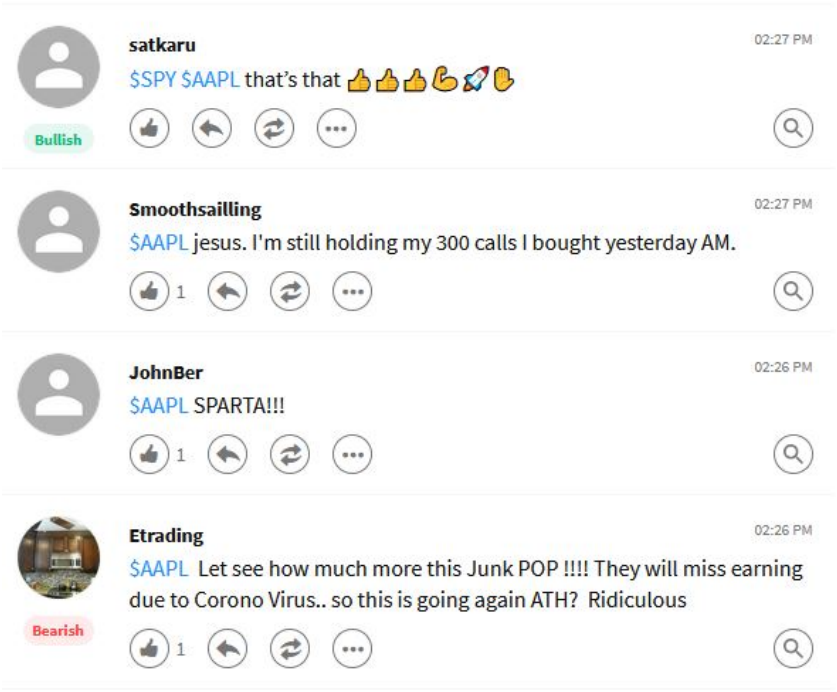


Figure 3.2: Screenshot of StockTwits as of 3rd March 2020.
The first message is labeled as bullish by the user “satkaru", the two next are unlabeled messages that will be classified using a machine learning classifier and the last message is labeled as bearish by the user “Etrading”.

Figure 3.3 shows the distribution of user-labeled and unlabeled messages. Overall, around 30 million messages are user-labeled and 60 million messages are unlabeled. Among the user-labeled messages we find five times more bullish than bearish ones. This ratio indicates that investors are on average optimistic about the market, which is consistent with findings in the literature, e.g., Renault (2017). Such class imbalance is a well-known issue in machine learning classification, which we address below. We will classify the unlabeled messages using a machine learning algorithm trained on the set of user-labeled messages. Since not every message contains substantial information, we believe that the sentiment classification should not be a bullish/bearish dichotomy. Hence, we allow for a neutral class to account for messages that do not take a clear stand.

Figure 3.4 shows a log-log histogram of the number of followers per user and a log-log his-

[1] AAPL is the ticker for Apple.

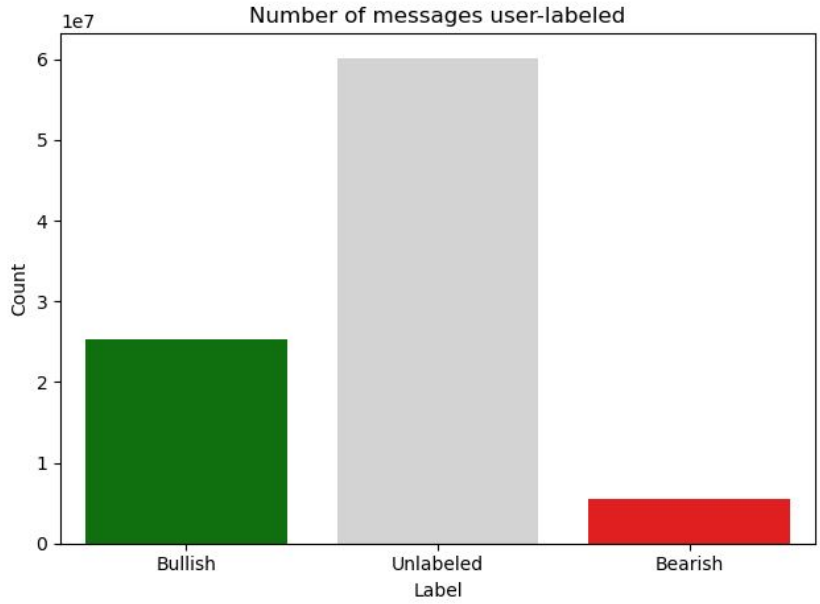


Figure 3.3: Number of user-labeled messages in each category.
There are around 60 million unlabeled messages that we classify using the 30 million user-labeled bullish and bearish messages.

togram of the number of messages posted by users. As expected, there are few users with many followers (they can be seen as “influencers"), and many users with a few followers. In addition, most users seem to post on average between 10 and 400 messages and a few post a lot more.

Figure 3.5 shows the top 30 most discussed tickers on the platform. Of all tickers extracted, around 75% are ordinary common share, 15% are ETFs and the remaining 10% are other types of securities. Not surprisingly, the S&P index is the most discussed, followed by Apple and other big tickers. The messages about the 15 (30) biggest tickers represent 20% (25%) of the total number of messages, which indicates that users talk about a wide panel of tickers and not only big firms. The bottom graph shows a histogram of the number of messages per ticker. The x-axis is log-scaled because of extreme values, the distribution is highly skewed.

Messages contain qualitative information, which needs to be transformed into quantitative data for the computer to understand. Thereto, we first apply the following preprocessing operations: an apostrophe handler, a contraction form handler (i.e. “aren’t" becomes “are not"), tickers removal, stop words (i.e. “a", “the", “of") removal[2], users removal, lemmatization, URLs removal and a simple spell corrector dealing with more than two repeated characters (i.e. “soooo goooood" becomes “soo good"). Table 3.1 shows five examples of messages before and after preprocessing.

One of the first steps in NLP is the tokenization : the way of slicing a piece of text in smaller units

[2]We follow Renault (2020) and Saif et al. (2014) and use a restrictive list of stopwords to avoid accuracy decrease.

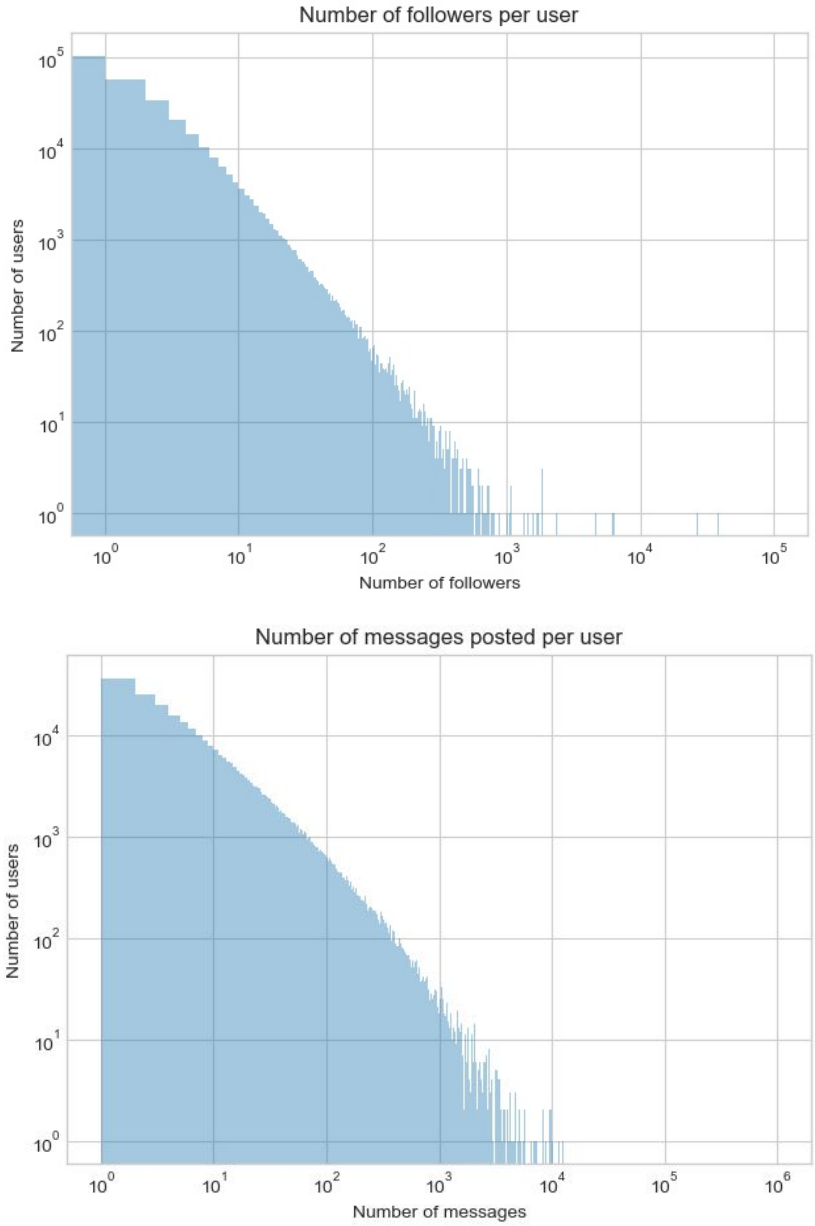


Figure 3.4: User summary statistics.
Top graph is a log-log histogram of the number of followers per user and the bottom graph shows the log-log histogram of the number of messages posted by users.

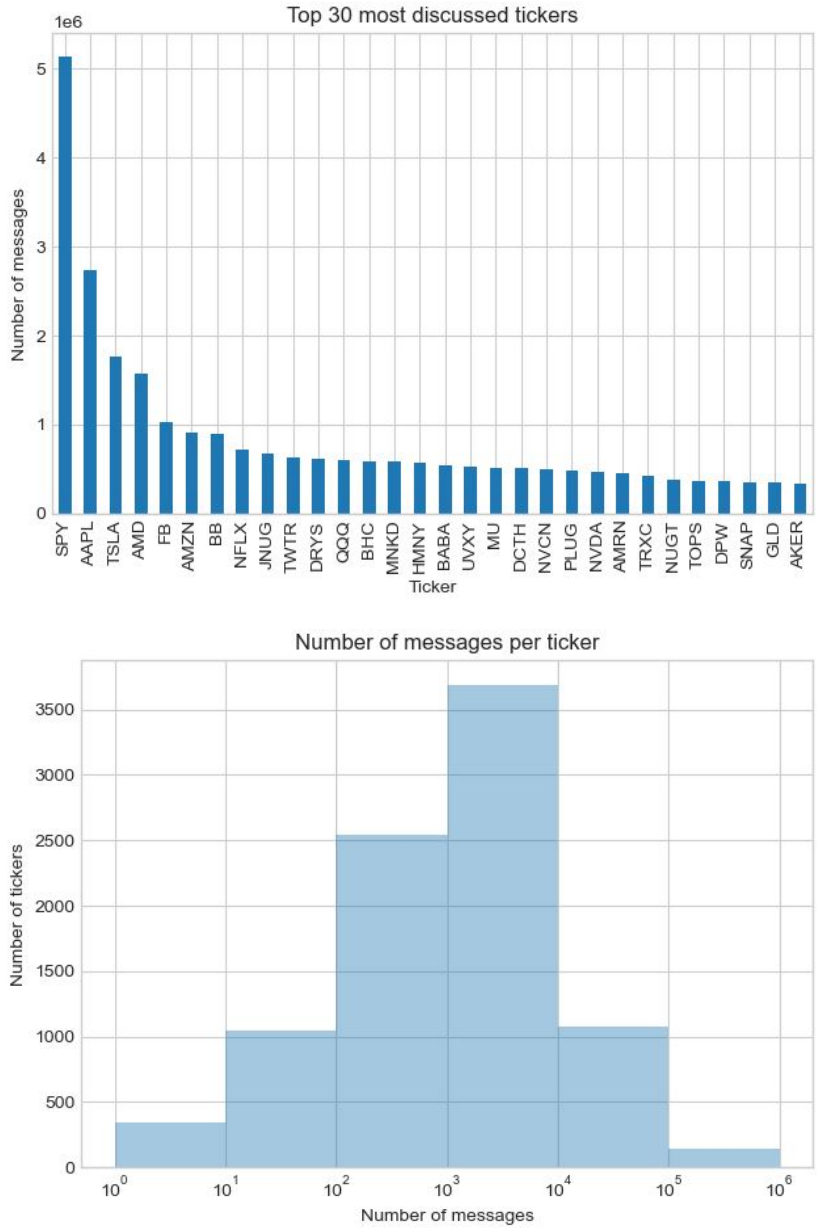


Figure 3.5: Firm summary statistics.
Top graph shows the top 30 most discussed tickers on the platform. SPY is the ticker of the S&P and AAPL is the ticker for Apple. Bottom graph shows the distribution of the number of messages across tickers.

called tokens or terms. In financial lingo, some words only have meaning when associated with other words (i.e “bad apple" or “bear flag"). N-gram models allow accounting for words frequently occurring together with other words. The main hyperparameter in N-gram models is the size of the group of words considered : unigram is a term with only one word, bigram is a term with two consecutive words, etc. Bigger N-grams models increase dramatically the size of the vocabulary (i.e : the collection of all terms considered). We are able to control the size of the vocabulary by tuning a hyperparameter keeping only a given number of most frequent terms. We chose to work with a vocabulary consisting of the one million most frequent unigrams, bigrams or trigrams.

Figure 3.6 represents the bullish and bearish word clouds. These correspond to the most frequent terms (up to 3-grams) in user-labeled bullish (bearish) messages relative to their total appearance. The size of the terms represent their importance in the cloud. In the bullish cloud, we see words such as “bullish divergence", “room to grow", “lot potential" which are clearly bullish signals. In the bearish cloud, we find words such as “recent resistance", “short setup", “bad apple" which indeed indicate bearish signals. These findings are reassuring in the sense that the content of the messages on the platform are consistent with their labels. The term “aldox" in the bullish cloud caught our attention. After some research, it is an abbreviation for Aldoxorubicin, a drug against tumors and is associated with pharmaceutical messages where investors were very enthusiastic about it (example of related message: “aldox is on the slide. have great faith this is truly world change"). That is why the term is appearing almost exclusively in bullish messages, hence in the bullish cloud.

On another note, most of the times messages are written by humans. It is however possible that some messages are generated by a robot (for instance to spread news or articles). In such cases, we define a neutral class that would absorb these messages if no substantial information is embedded in these messages. The term "long position open" present in the bearish cloud is an anomaly due to bearish user-labeled messages of intraday alerts such as “sell $labd close labd long position. open labd short position. time: 14:53 ny price: $13.64 zquant intraday alerts”. However, such anomalies are not an issue, a message “long position open” has a bullish probability of 0.91 and gets classify as bullish as it should. This is the case because when classifying a message, the score of the unigram “long” is much stronger than the score of the trigram “long position open”. From a linguistic point of view our approach is brute force. However, in turn, it works at a massive scale. Overall, the data quality is very good.

Figure 3.6: Bullish and bearish word clouds.
Bullish word cloud (top), bearish word cloud (bottom). These correspond to the most frequent terms (up to 3-grams) in user-labeled bullish (bearish) messages relative to their total appearance. The size of the terms represent their importance in the cloud.

Before preprocessing	
(1)	@CassandraTwit $uvxy contango 3.5%...still long. gooooooood
(2)	$FRPT Take profits while you still can.
(3)	$UVXY $tvix go time boys and girls. Holding overnight again
(4)	$dnr Nice upgrade as company goes into its quiet period!
(5)	$SPY market won't reverse again towards closing. Get put options.

After preprocessing	
(1)	contango still long good
(2)	take profit while you still can
(3)	go time boy and girl hold overnight again
(4)	nice upgrade as company go into its quiet period
(5)	market will not reverse again towards closing get put options

Table 3.1: Preprocessing of five sample messages.
Preprocessing operations include: punctuation removal, lower casing, apostrophe handling, contraction form handling (i.e. “won't" becomes “will not"), tickers removal, users removal, URLs removal, parsing and a simple spell corrector dealing with more than two repeated characters (i.e. “goooood" becomes “good")

3.3 Sentiment classification

Since mid-2010, StockTwits users have the choice to label their own messages as either *bearish* or *bullish* or to leave it unlabeled. Unlabeled messages are tricky to deal with because the user either deliberately chose to leave the message neutral by not labeling it or forgot to click on a label. Figure 3.7 shows the proportions of user-labeled messages in each category across time. In the early years of the platform, most messages are unlabeled, presumably because users were not familiar with the sentiment label yet. Albeit the proportion of unlabeled messages monotonically declines over the years, almost 60% of the more recent messages are still unlabeled. We believe that by far not all unlabeled messages reflect neutral opinions. Figure 3.9 shows that a lot of unlabeled messages get classified in either bullish or bearish messages, hence these unlabeled messages had indeed valuable information that we are now able to capture.

As one of the goal of this paper is to build an accurate time-series sentiment measure for individual firms, it motivates the use of Natural Language Processing to classify all unlabeled messages in either bullish, neutral or bearish class.

To classify unlabeled messages, we use a logistic regression of the labels on TFIDF transformed user-labeled messages, as in Yildirim et al. (2018) and Qasem et al. (2015). TFIDF stands for Term Frequency-Inverse Document Frequency and is a widely used method to transform a text, in our case a message m, into a numerical vector, $TFIDF_m$. The dimension of this vector

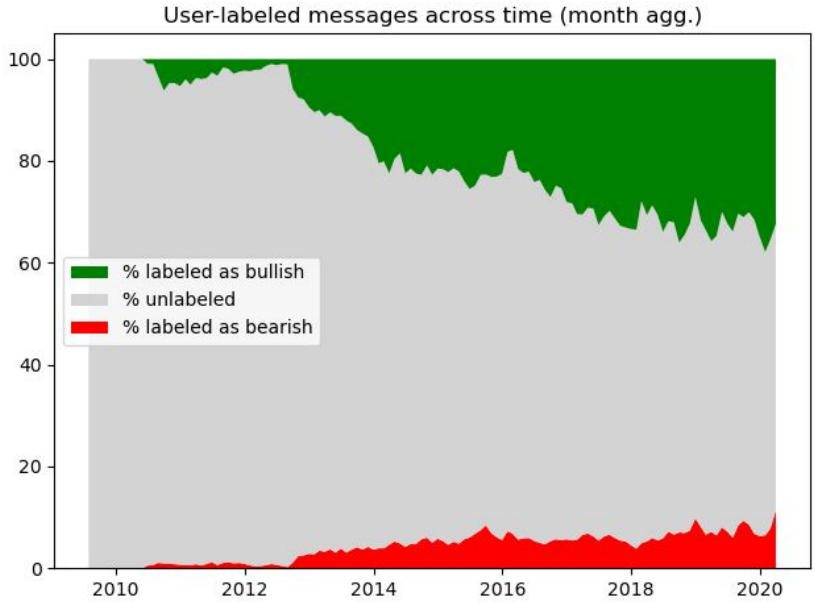


Figure 3.7: Proportions of user-labeled messages in each category
Proportions of user-labeled messages in each category: bullish (green), bearish (red), and unlabeled (gray). Proportions are aggregated monthly.

is equal to the size of the vocabulary (the collection of all terms across all messages). The components of the vector encode the importance of the corresponding terms t in the message m, as formally defined by $TFIDF_{m,t} = TF_{m,t} \cdot IDF_t$. The first factor measures how frequently term t appears in the message,

$$TF_{m,t} = \frac{\sum_{i=1}^{N_m} \mathbf{1}_{t=t_{m,i}}}{N_m},$$

where N_m denotes the number of terms $t_{m,i}$ in message m. The second factor measures how important term t is to the message,

$$IDF_t = \log\Big(\frac{V}{\sum_{j=1}^{V} \mathbf{1}_{t \in m_j}}\Big),$$

where V denotes the total number of messages m_j. A term t appearing in many documents (such as “the", “is", “of") is likely to have low information content, hence a low IDF_t.

As seen in Figure 3.3, user-labeled messages exhibit five times more bullish messages than bearish messages. Such an imbalance is a well-known issue in machine learning classification and needs to be dealt with to avoid biases towards the over-represented class (see Chawla et al. (2004)). To tackle class imbalance, the most common technique is to randomly oversample the minority class, which consists of repeating some samples of the minority class and balance the number of samples between classes in the data. We follow a different approach. In addition to bullish and bearish classes, we define an artificial neutral sentiment class as it is possible that some users are not expressing any opinion and sometimes post finance-irrelevant messages.

To do so, we optimally select classification score thresholds and eliminate the class imbalance bias at the same time.

Performance measures widely used in machine learning classification are precision, recall, F1 score and accuracy. The first step is to define a class as the positive class. Instances (messages) are then divided into true positives TP (predicted positive, actual positive), false positives FP (predicted positive, actual negative), true negatives TN (predicted negative, actual negative), and false negatives FN (predicted negative, actual positive). Precision $PRE = \frac{TP}{TP+FP}$ is the proportion of true positives among the predicted positives. Recall $REC = \frac{TP}{TP+FN}$ is the proportion of true positives among the actual positives. The precision-recall trade-off is captured by the F1 score, $\frac{2 \cdot PRE \cdot REC}{PRE+REC}$, the harmonic mean of precision and recall. Accuracy $ACC = \frac{TP+TN}{TP+TN+FP+FN}$ is the fraction of correct predictions regardless of the label.

We use 80% of the user-labeled (bearish and bullish) messages as a training set and keep 20% as a test set, then we run two binary classifiers. The first (second) classifier sets bullish (bearish) as positive and non-bullish (non-bearish) as negative class. Every message then falls into the following set of labels: {(non-bullish, bearish), (bullish, bearish), (non-bullish, non-bearish), (bullish, non-bearish)}. For the two outer cases the two algorithms agree and the final classification is defined to be bearish (non-bullish, bearish) or bullish (bullish, non-bearish), respectively. For the two inner cases, (bullish, bearish) and (non-bullish, non-bearish), the two algorithms disagree, so that the final classification is defined to be neutral. Formally, every message m is then mapped on either

$$m \mapsto \begin{cases} \text{(non-bullish, bearish)} & =: \text{bearish} \\ \text{(bullish, bearish)} & =: \text{neutral} \\ \text{(non-bullish, non-bearish)} & =: \text{neutral} \\ \text{(bullish, non-bearish)} & =: \text{bullish.} \end{cases}$$

Precision, recall, and F1 scores of the two algorithms differ because they depend on which class is defined as the positive one. To select optimal classification thresholds, we maximize the F1 score of either algorithm. The green (red) line in Figure 3.8 is the F1 score for the bullish versus non-bullish (bearish versus non-bearish) classification, respectively. Circles indicate the maximal F1 scores, along with the corresponding classification score thresholds, 0.50 and 0.72, respectively.

If the classification score of a message is bigger (smaller) than 0.72 (0.50), then both classifiers agree on the sentiment and the message is classified as bullish (bearish), respectively. If the classification score is between 0.50 and 0.72, the classifiers disagree, (bullish, bearish), and we consider the message as neutral. Finally, we overwrite the sentiment of a message predicted by the classifier by the user-labeled sentiment whenever the latter is available. Research in

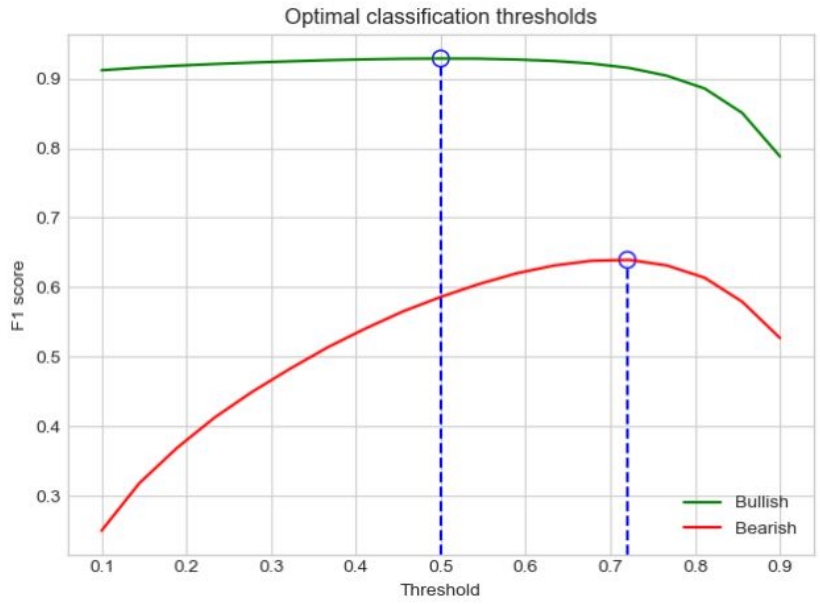


Figure 3.8: Optimal classification thresholds.
The green (red) line is the F1 score for the bullish versus non-bullish (bearish versus non-bearish) classification, respectively. Circles indicate the maximal F1 scores, along with the corresponding classification score thresholds.

sentiment classification shows that human annotators tend to agree about 80 to 85% of the time when evaluating the sentiment of a document (see e.g. Wilson et al. (2005) and Chen et al. (2020)). This usually represents the accuracy that a sentiment classifier should meet or beat. The accuracy in the test set of our combined classifier is 86%.

Figure 3.9 shows the proportions of classified messages in each category. Percentages of bearish (sum of labeled and classified as bearish) and bullish (sum of labeled and classified as bearish) messages are stable over time, suggesting that our classification method is robust. Even if most messages were not user-labeled in the early years of the platform, as seen in Figure 3.7, we are now able to classify the sentiment of most messages posted in this period. Consistent with the over-representation of bullish messages observed in the user-labeled messages in Figure 3.3, there are many more messages classified as bullish than bearish. Typical messages classified as bullish are messages such as “buy buy” or “hope the pump come soon” whereas typical bearish messages are messages such as “sell everything” or “start short position here”. Neutral messages are either empty, irrelevant to finance (e.g. “political posturing friend”[3]) or ambiguous (e.g. “lol wow”).

[3]This is a reply to the following message : "honestly, how dumb can you be to believe that china was going to buy significant amount of agricultural products after the breakdown in trade talks. even if they buy it will be just a little bit and not significant"

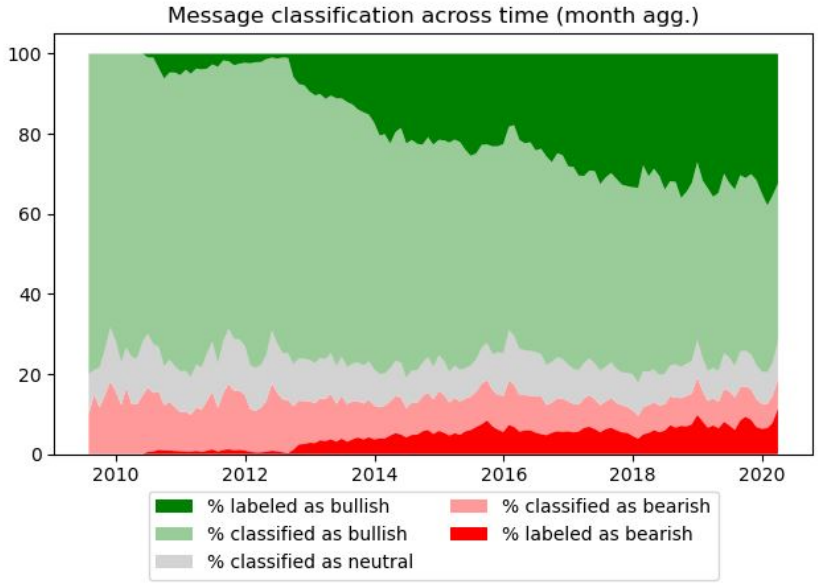


Figure 3.9: Proportions of classified messages in each category
Bullish (light green predicted, green user-labeled), bearish (light red predicted, red user-labeled), and neutral (gray). Proportions are aggregated monthly.

3.4 Polarity

To build sentiment measures for individual firms and the whole market on a given day, we count the sentiments in the messages. To match the timeline on which close-to-close stock returns are computed and to avoid forward-looking bias, we aggregate messages on a close-to-close manner. That is, polarity on day t is computed with messages posted from 4:00 pm on day $t-1$ to 4:00 pm on day t. As stock returns are not computed outside of business days, we shift messages posted outside of business days to the next business day available, then remove any non-business day from our sample. We denote by $C_{i,t,j}$ the sentiment of the jth message about firm i on day t. It is set to 1, 0, or -1 for bullish, neutral, or bearish, respectively. We follow Ranco et al. (2015) and define the polarity of firm i as

$$P_{i,t} = \frac{\sum_{j=1}^{V_{i,t}} (\mathbf{1}_{C_{i,t,j}=1} - \mathbf{1}_{C_{i,t,j}=-1})}{\sum_{j=1}^{V_{i,t}} (\mathbf{1}_{C_{i,t,j}=1} + \mathbf{1}_{C_{i,t,j}=-1})},$$

where $V_{i,t}$ denotes the number of messages about firm i on day t.[4]

As an aggregate measure, we define the polarity for the whole market as a weighted average over all firms

$$P_t^M = \frac{\sum_i V_{i,t} \cdot P_{i,t}}{V_t^M},$$

[4] If $V_{i,t} = 0$ then we set $P_{i,t} = 0$.

where $V_t^M = \sum_i V_{i,t}$ denotes the number of messages on day t.

Figure 3.10 shows a scatter plot of the market polarity P_t^M versus the polarity of the SPY[5]. We do not expect the market polarity to be the SPY polarity because the stock universe is not the same (market polarity contains stocks that are not necessarily in the S&P500 and vice versa). The slope coefficient of the regression line is statistically significantly positive and the contemporaneous Pearson correlation coefficient is 0.53, suggesting that the market polarity is an accurate measure of the aggregated sentiment of the market. Also, consistent with Figure 3.9, SPY and market polarities are bullish-biased.

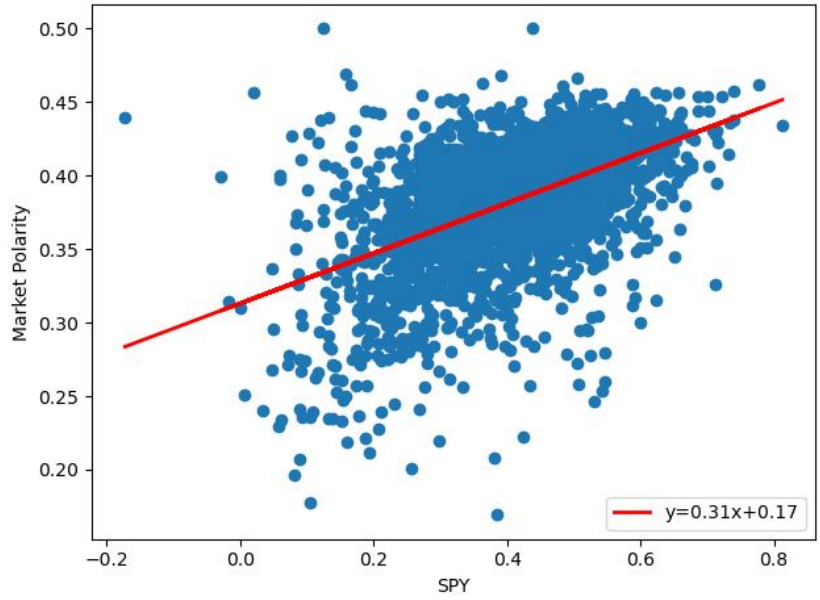


Figure 3.10: Market polarity versus the SPY polarity.
Market polarity on the y-axis versus the polarity of the SPY on the x-axis. The red line shows the linear regression line and coefficients.

At this point, for stationary reasons[6], we compute the median of daily message volume for each ticker and exclude from our sample tickers that have less than a median of 50 daily messages. Our final sample contains 19 tickers. We refer to the appendix for the list of tickers covered as well as more information about the trimming process.

To understand how polarity is related to investor sentiment, we run two linear regressions of contemporaneous daily returns on polarity and 5-days Cumulative Abnormal Polarity :

$$R_{i,t} = \alpha + \beta \cdot P_{i,t} + \epsilon_{i,t},$$
$$R_{i,t} = \alpha + \beta \cdot CAP_{i,5} + \epsilon_{i,t}.$$

Table 3.2 shows that β is positive and significant for both regressions. This indicates that

[5] SPY is an ETF tracking the S&P500 return. It is the largest ETF in the world.
[6] Computing the polarity ratio with few daily observations would lead to a spiky time series.

	$R_{i,t}$	$R_{i,t}$	$R_{i,t+1}$	$R_{i,t+1}$
Constant	-0.0047***	-7.92e-05	-0.0002	7e-06
	(0.000)	(9.81e-05)	(0.000)	(9e-05)
$P_{i,t}$	0.009***		0.0003	
	(0.000)		(0.000)	
$CAP_{i,5}$		0.0007***		-0.0002
		(0.000)		(0.000)
R^2	0.012	0.000	0.000	0.000
No. Obs.	34100	34024	34100	34024

Table 3.2: Linear regressions
Results from linear regressions of contemporaneous and next period stock returns on polarity. Stock returns are trimmed at the 5% percentile on both sides. Standard errors are reported in parentheses. Statistical significance at the 99%, 95%, and 90% level is indicated with ***, **, *, respectively.

polarity is a good proxy for the sentiment of investors. Further supporting evidence is given by the correlation between polarity and contemporaneous stock returns at the firm level. Figure 3.11 shows the time series during 2019 for the top 6 most discussed tickers. In our entire panel of firms, correlations are always positive and range between 0.1 and 0.3.

We also run two linear regressions of next day returns on polarity and 5-days Cumulative Abnormal Polarity :

$$R_{i,t+1} = \alpha + \beta \cdot P_{i,t} + \epsilon_{i,t},$$
$$R_{i,t+1} = \alpha + \beta \cdot CAP_{i,5} + \epsilon_{i,t}.$$

Table 3.2 reveals that polarity has no predictive power for next day stock returns unconditionally. However, the next section depicts how polarity still has embedded information around specific events.

3.5 Event studies

Event studies constitute a statistical method widely used in financial econometrics. In general, they are used to measure the effect of events on the market value of firms. Well known applications of event studies include the testing of various forms of the efficient market hypothesis (EMH) (see Fama et al. (1969) and Fama (1991)). What's more, as described in MacKinlay (1997), event studies can also be applied with little modification to other variables than stock returns.

To design an event study, the first step is to define events of interest and the *event window* over which a variable will be examined. Adhering to common practice, we choose the event window expanded from 20 business days before the event to 20 business days after the event. The

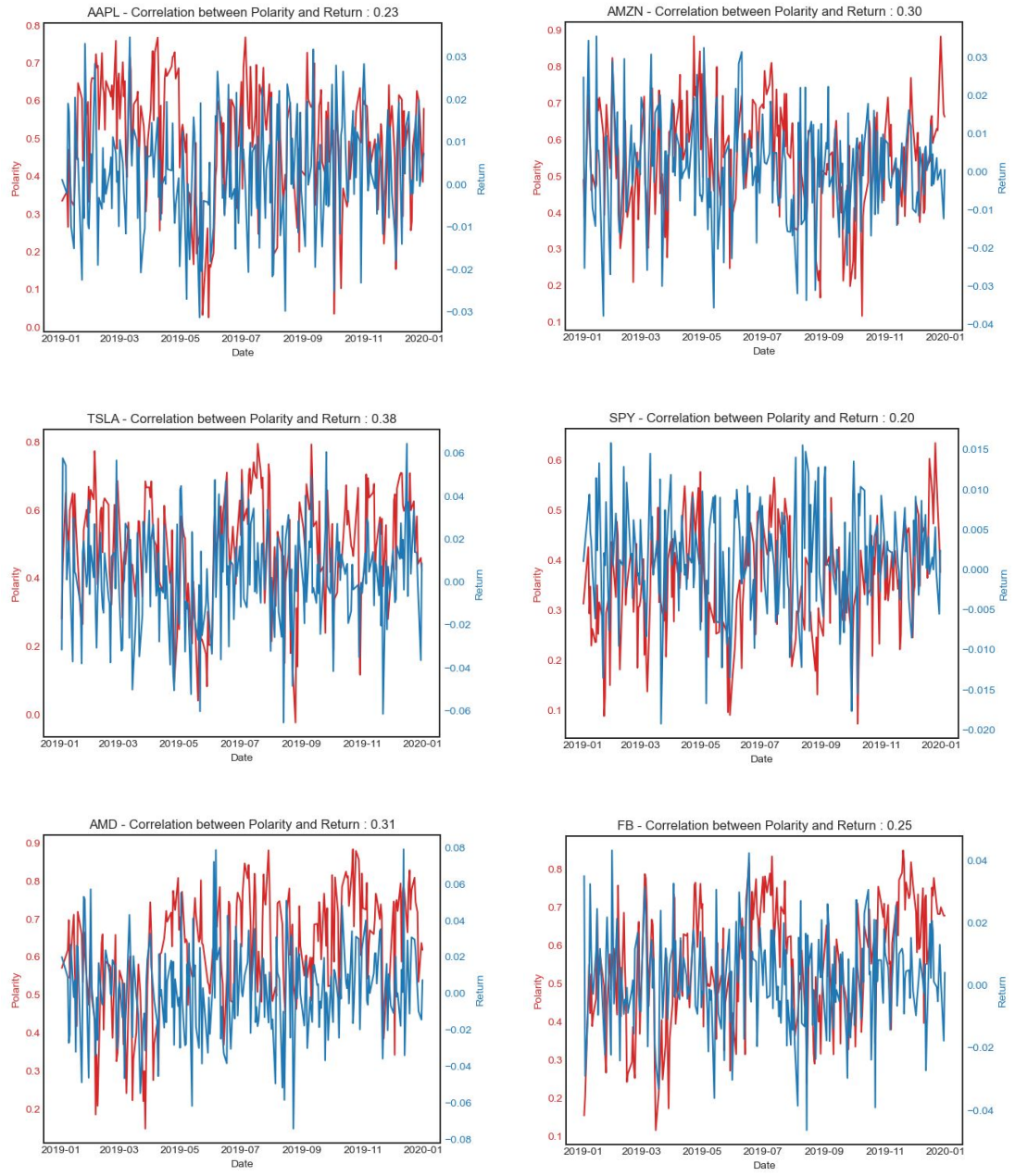


Figure 3.11: Time series of daily polarity
Time series of daily polarity (red - left axis) and daily stock returns (blue - right axis) since 1st of January 2019 for the top 6 most discussed tickers. Pearson correlation between the two time series is shown in the title.

estimation window is used to estimate the parameters of the market model. A common choice is a one-year window ending 20 days before the event. Figure 3.12 shows the corresponding timeline.

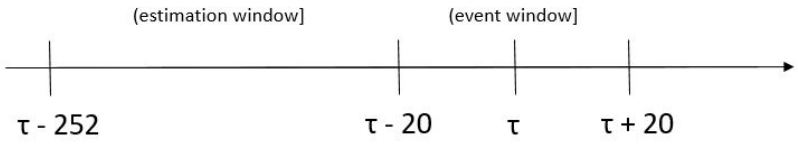


Figure 3.12: Timeline of our event studies.
τ is an event date, the event window has 41 business days centered around the event date and the estimation window is a one-year rolling period prior the event.

Then, event studies require the distinction between a normal and an abnormal measure. The normal measure is defined as the expected measure conditioning on the event not taking place. In other words, we need an expectation of the variable of interest that would have been measured if the event never existed. The abnormal measure is defined as the measure minus the normal measure. Formally, let $X_{i,t}$ be a measure of interest (i.e. stock return, polarity, ...), $AX_{i,t}$ be the abnormal measure and $E(X_{i,t}|Z_{i,t})$ the normal measure with $Z_{i,t}$ being the conditional information for the normal measure model. We have:

$$AX_{i,t} = X_{i,t} - E(X_{i,t}|Z_{i,t}).$$

The normal measure $E(X_{i,t}|Z_{i,t})$ is usually computed using either a constant mean model or a market model. Constant mean models use $E(X_{i,t}|Z_{i,t}) = \mu$ where μ is the mean of the measure during the estimation window. We chose to work with a market model but all following results hold with the constant mean model as well. As stated in MacKinlay (1997), the variance of the abnormal measure is not reduced a lot by choosing a more sophisticated model, hence the event study is not sensitive to the choice of the normal model. The market model we are using is defined as the following:

$$X_{i,t} = \alpha_i + \beta_i \cdot X_t^M + \epsilon_{i,t},$$

with $E(\epsilon_{i,t}) = 0$, $V(\epsilon_{i,t}) = \sigma_\epsilon^2$ and X_t^M the measure of the whole market. We estimate the parameters of the market model in a one-year rolling estimation window. To avoid overlaps between estimation windows and events, we remove any event day from the estimation windows. This ensures that large event returns do not influence the parameters of the normal measure.

3.5.1 Events

We define an event as an unusual high number of daily messages for a particular firm. We conjecture that a sudden peak in StockTwits message volume indicates that an important firm event is happening on the day of the peak. As a robustness check, we show in Figure 3.13 that

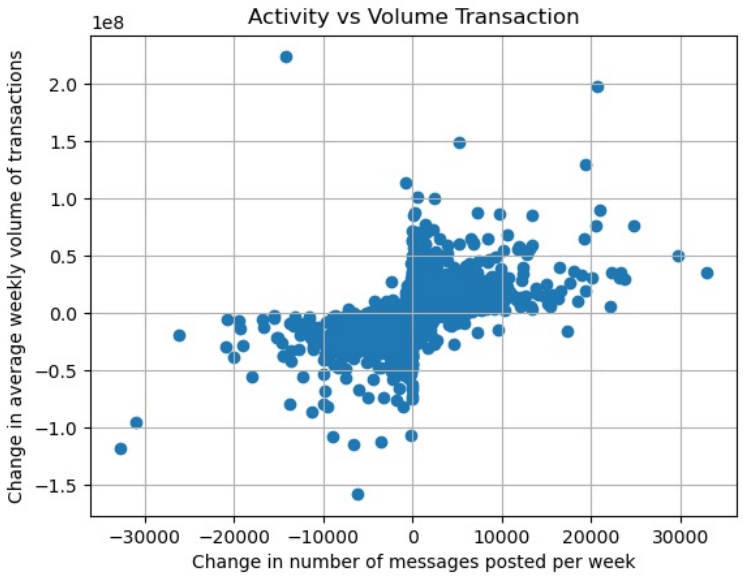


Figure 3.13: Volume of transactions and message activity.
Changes in weekly volume of transactions on the y-axis versus changes in message activity on the x-axis. Activity is measured in weekly messages posted per firm.

increases (decreases) in number of message volume are positively associated with increases (decreases) in contemporaneous weekly volume of stock transactions. These co-movements mean that investors not only post about these stocks but also trade them simultaneously, adjusting their portfolios. This indicates that the message volume peaks is a good proxy to identify event dates as there is no lag between message activity and trades.

To measure unusual activity peaks, we use the market model (3.5) on a one-year rolling estimation window and regress the daily relative change of message volume of individual firms $\frac{\Delta V_{i,t}}{V_{i,t-1}}$ on the daily relative change of total message volume $\frac{\Delta V_t^M}{V_{t-1}^M}$. Formally,

$$\frac{\Delta V_{i,t}}{V_{i,t-1}} = \alpha_i^V + \beta_i^V \cdot \frac{\Delta V_t^M}{V_{t-1}^M} + \epsilon_{i,t}.$$

We can then compute the abnormal volume of messages for firm i as :

$$AV_{i,t} = \frac{\Delta V_{i,t}}{V_{i,t-1}} - \hat{\alpha}_i^V - \hat{\beta}_i^V \cdot \frac{\Delta V_t^M}{V_{t-1}^M}.$$

We define an event for ticker i as a day t where the standardized abnormal volume exceeds 2,

$$\frac{AV_{i,t} - \mu_{AV_i}}{\sigma_{AV_i}} > 2.$$

Then, we define the type of the event as either bullish, neutral or bearish. We use the abnormal polarity $AP_{i,t}$ of the event date to assess how on average investors perceive the event. Figure 3.14 shows the distribution of abnormal polarities. We chose to use the one-third (-0.03) and two-third percentile (0.07) of the distribution of abnormal polarities as thresholds for the type of the event. Conditionally on the existence of an event for firm i at day t, we have:

$$Type_{i,t} = \begin{cases} Bullish & \text{if } AP_{i,t} > 0.07, \\ Neutral & \text{if } AP_{i,t} \in [-0.03, 0.07], \\ Bearish & \text{if } AP_{i,t} < -0.03. \end{cases}$$

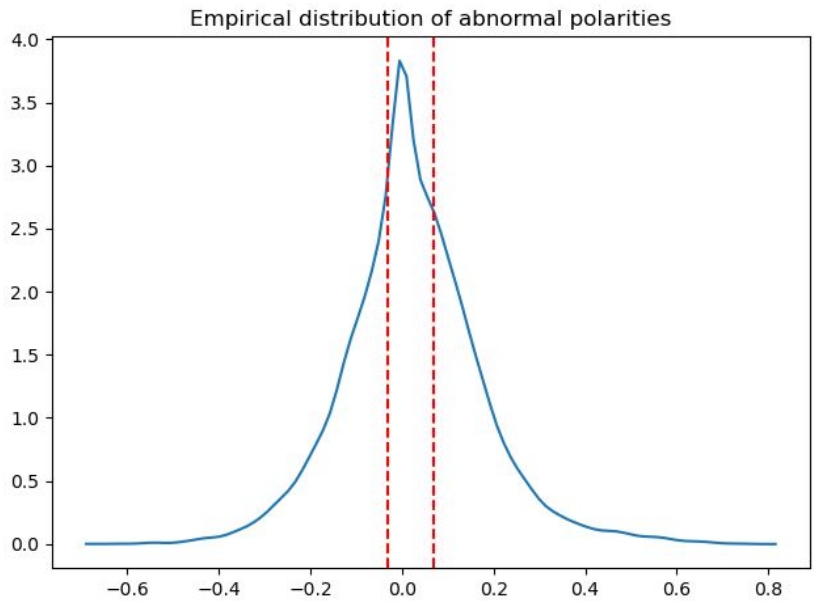


Figure 3.14: Empirical distribution of abnormal polarities.
Red dashed lines show the one-third and two-third percentiles.

As an illustration, Figure 3.15 shows for Apple the time-serie of message volume and the bullish, neutral and bearish events identified as green up-triangles, gray circles and red down-triangles, respectively. Between 2011 and 2020, our algorithm identified 73 events for Apple. Interestingly, our methodology allows us to capture more than earning announcements : about half of these events correspond to earning announcements, but some also correspond to Apple *Keynotes*[7] or even CEO letters addressed to investors. Across 19 tickers, we identify 1131 events distributed across the three categories : 454 bullish events, 294 neutral events and 383 bearish events. This coverage is on par with previous studies (i.e: MacKinlay (1997) has 30 firms and 600 events) Figure 3.16 shows the identified events and their types across the years.

[7] Keynotes are presentations that Apple gives to the press, often presenting new products.

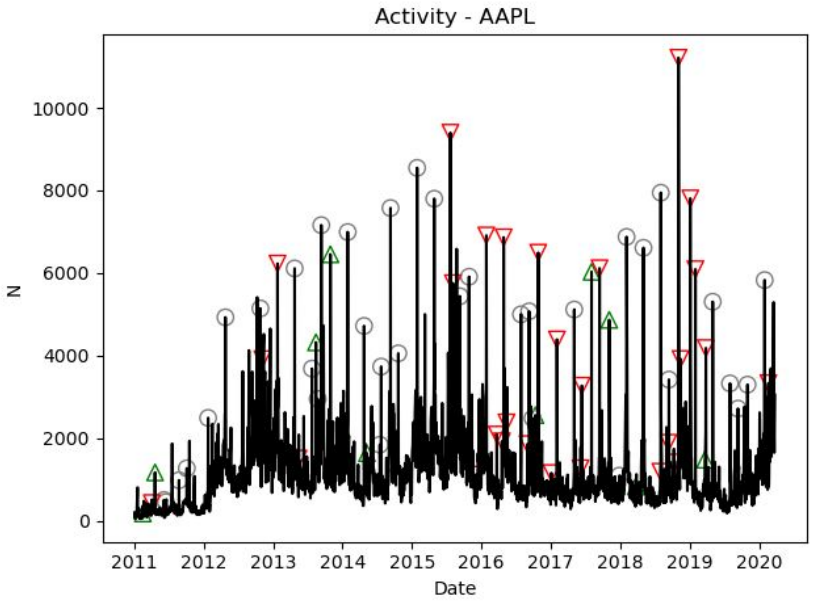


Figure 3.15: Daily message volume for Apple.
Events are days with an unusual high number of messages. Green upper-triangles show bullish events, gray circles are neutral events and red down-triangles represent bearish events.

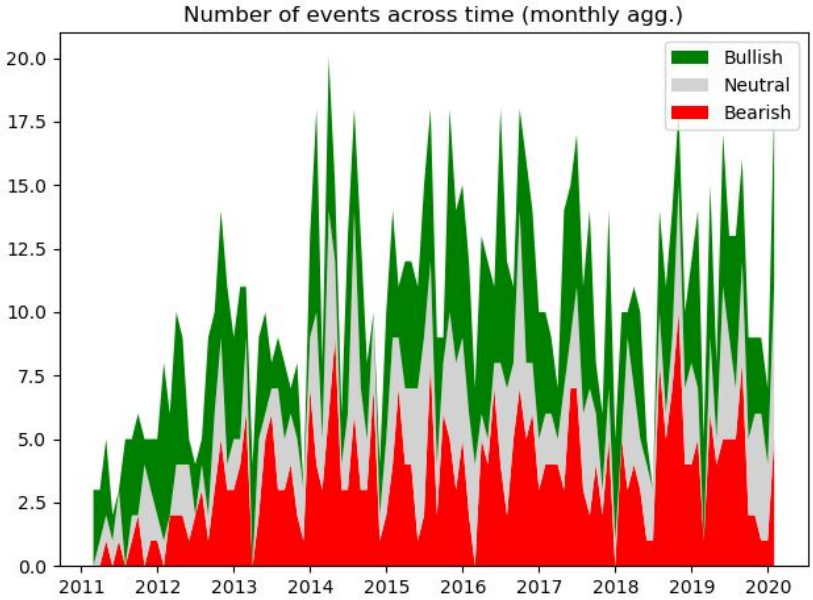


Figure 3.16: Number of events in each category across time. Numbers are aggregated monthly.

3.5.2 CAAR and CAAP

We estimate the parameters for the market models using OLS, as it is a consistent and efficient estimator under general conditions.[8]

Abnormal Returns

Given the market models parameters $\hat{\alpha_i}^R$ and $\hat{\beta_i}^R$ estimated in (3.5), the abnormal return is

$$AR_{i,t} = R_{i,t} - \hat{\alpha_i}^R - \hat{\beta_i}^R \cdot R_t^M,$$

with R_t^M being the market return. Then, we can compute the cumulative abnormal returns (CAR) around a firm i event τ as:

$$CAR_i(\tau, t) = \sum_{s=-20}^{t} AR_{i,\tau+s},$$

and finally get the cumulative average abnormal returns (CAAR) across the N events as:

$$CAAR(t) = \frac{1}{N} \sum_{j=1}^{N} CAR_{i_j}(\tau_j, t).$$

Variance of CAR is computed following MacKinlay (1997) as :

$$var(CAR(\tau, t)) = \frac{1}{N^2} \sum_{i=1}^{N} (CAR_i(\tau, t) - CAAR(t))^2.$$

Figure 3.17 shows the CAAR around the events identified. This plot is consistent with MacKinlay (1997). It shows that CAAR related to bearish (bullish) events displays a downward (upward) jump at the event date respectively, and then the jumps are followed by a stable CAAR during the 20 days period after an event. Interestingly, there is a systematic (small) shift in the CAAR already 1 day before an event. The CAAR related to the neutral events exhibits a slight upward shift around the event date but it fades away after a few days. The CAAR related to bearish events shifts already a few days before the event but this shift is not statistically significant. Additionally, as we see in the top-left plot of Figure 3.19, box plots are not shifted, indicating that the conditional distributions of the CAR are not statistically different from each other. Mann-Whitney U-test (Table 3.3) shows that 5 days before an event, the median of the CAR distribution of the bullish events is neither statistically different from the median of the neutral nor the bearish events.

[8] An interested reader can refer to MacKinlay (1997) for more details

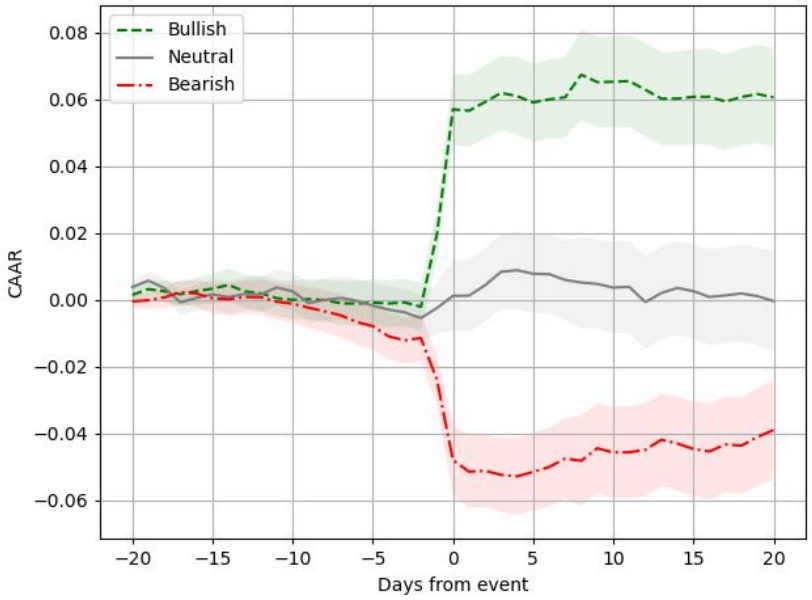


Figure 3.17: Cumulative average abnormal returns around identified events.
CAAR related to bearish, neutral and bullish events are displayed with the red, gray and green line, respectively. Areas around lines show confidence intervals at the 95% level.

Abnormal Polarity

Similar to the subsection above, given the market models parameters $\hat{\alpha_i}^P$ and $\hat{\beta_i}^P$ estimated in (3.5), the abnormal polarity is

$$AP_{i,t} = P_{i,t} - \hat{\alpha_i}^P - \hat{\beta_i}^P \cdot P_t^M,$$

with P_t^M being the market polarity computed in (3.4). Then, we can compute the cumulative abnormal polarity (CAP) and cumulative average abnormal polarity (CAAP) from $\tau - 20$ to $t \leq \tau + 20$ as:

$$CAP_i(\tau, t) = \sum_{s=-20}^{t} AP_{i,s+\tau},$$
$$CAAP(t) = \frac{1}{N} \sum_{j=1}^{N} CAP_{i_j}(\tau_j, t).$$

Figure 3.18 shows the CAAP around the events identified. The main findings about polarity are twofold. First, conversely to CAAR, CAAP for bullish and bearish events are not constant after the event date, suggesting that users' sentiment about a firm tend to be biased towards recent past events. This can be explained by the fact that users might still post bullish or bearish messages about an event even several days after the event happening, even though

the return had already adjusted. Second, and more interestingly, it looks like the CAAP for bullish and bearish events shift several days earlier than the CAAR. This would indicate that the users are on average able to anticipate a future bullish (bearish) event and post positively (negatively) about the firm days before the rise in CAAR. Figure 3.19 illustrates this striking result with box plots (see Dekking et al. (2005) and Tukey (1977)) showing the distribution of the CAR and CAP. The line inside a box shows the median while the edges of each box represent the 25% and 75% quantile of the distribution. From above the edges of a box, a distance of 1.5 times the interquartile range is measured and a whisker is drawn up to the largest and lowest observed point from the data that falls within this distance.[9] The three plots on the left (right) show the boxplots for the CAR (CAP) 5 days before an event, on event date, and 5 days after the event, respectively. To show statistical significance, we use the Mann-Whitney U-test (see Mann and Whitney (1947) and Sheskin (1998)) on every plot to assess whether the three samples (bullish, neutral and bearish) represent populations with different median values.[10] Under the null hypothesis, the three samples represent distributions with equal medians. Let θ_i be the median of the distribution i. Formally, we test $H_0 : \theta_{bullish} = \theta_{neutral}$ against $H_1 : \theta_{bullish} > \theta_{neutral}$ and $H_0 : \theta_{neutral} = \theta_{bearish}$ against $H_1 : \theta_{neutral} > \theta_{bearish}$ 5 days before an event, on event date and 5 days after an event. We define U as the Mann-Whitney test statistic, Z as the normal approximation of the Mann-Whitney test statistic for large sample sizes, n_1 and n_2 as the sample sizes. We refer to Sheskin (1998) for the test statistic computation. Table 3.3 shows U-test estimates for pairwise comparisons. The null is rejected in every case except for CAR at $t-5$. 5 days before the event, the boxes of the CAR between bullish, neutral and bearish events are on the same level, suggesting no predictive power of abnormal returns, consistent with the EMH. However, the boxes showing the CAP 5 days before the event are already shifted, meaning that conditionally on an event happening, investors are able on average to anticipate correctly the type of this event. On the event date, the boxes of the CAR shift as the abnormal returns jump for both bullish and bearish events, consistent again with the EMH. Finally, 5 days after the event, the distributions of the CAR are very similar to the distributions on the event dates. Again, this is consistent with the EMH as the abnormal returns adjusted quickly on event date and all information is now embedded in the prices. The distribution of the CAP 5 days after the event continued to shift compared to the event date, as people continue to post about recent past events.

3.6 Portfolios

For the construction of a portfolio, in practice, we cannot condition on an event happening over the holding period. But we can repeat an experiment many times by supposing there is an event during the holding period and invest according to this hypothesis (long or short according to CAP). We would hope that repeating this experiment many times, we will be correct a few times and make large gain (or loss).

[9] Interquartile range is equal to the third quartile minus the first quartile.

[10] This interpretation only holds under stringent assumptions on the populations, namely that the two population distributions are equal up to a shift.

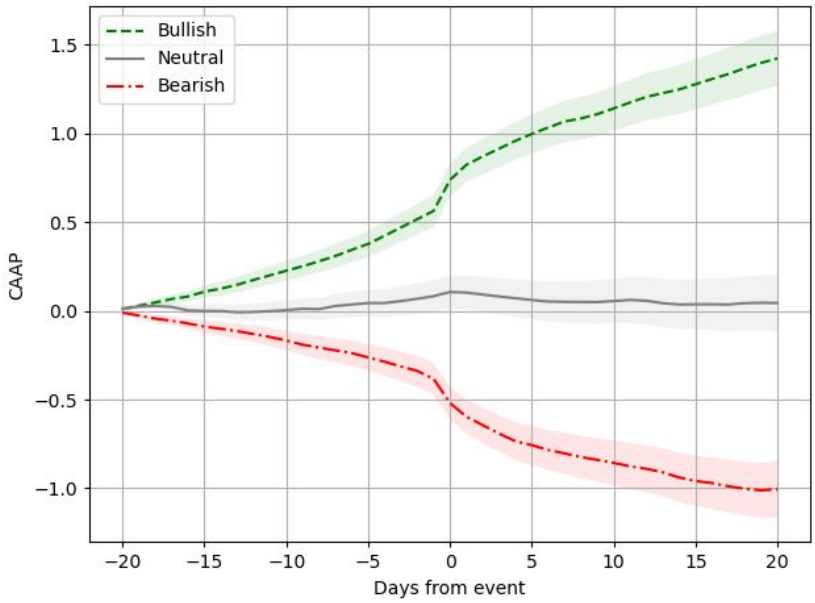
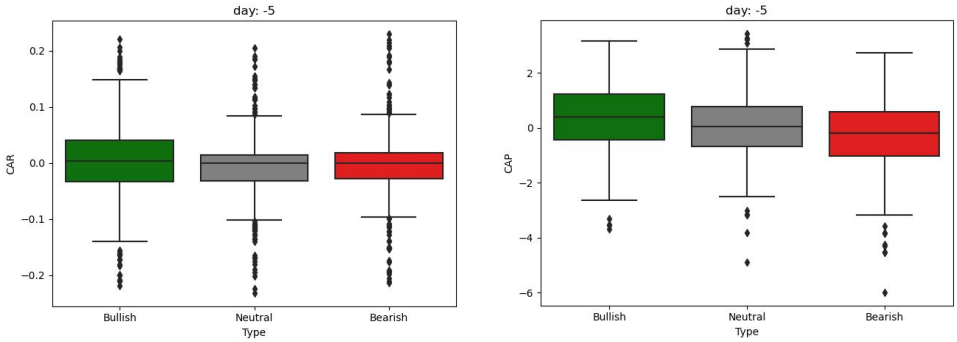


Figure 3.18: Cumulative average abnormal polarity around identified events.
CAAP related to bearish, neutral and bullish events are displayed with the red, gray and green line, respectively. Areas around lines show confidence intervals at the 95% level.

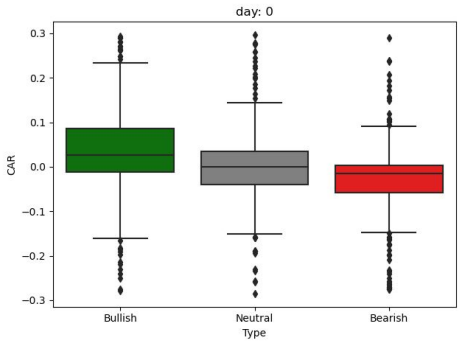
CAR					
	Alternative Hypothesis	U	Z	n_1	n_2
τ-5	$H_1 : \theta_{bullish} > \theta_{neutral}$	61109	-1.70	452	292
	$H_1 : \theta_{neutral} > \theta_{bearish}$	54168	-0.52	292	380
τ	$H_1 : \theta_{bullish} > \theta_{neutral}$	50967	-5.25***	452	292
	$H_1 : \theta_{neutral} > \theta_{bearish}$	43100	-4.96***	292	380
τ+5	$H_1 : \theta_{bullish} > \theta_{neutral}$	52738	-4.63***	452	292
	$H_1 : \theta_{neutral} > \theta_{bearish}$	43239	-4.91***	292	380
CAP					
	Alternative Hypothesis	U	Z	n_1	n_2
τ-5	$H_1 : \theta_{bullish} > \theta_{neutral}$	55408	-3.70***	452	292
	$H_1 : \theta_{neutral} > \theta_{bearish}$	47998	-3.00***	292	380
τ	$H_1 : \theta_{bullish} > \theta_{neutral}$	49385	-8.98***	452	292
	$H_1 : \theta_{neutral} > \theta_{bearish}$	42101	-5.36***	292	380
τ+5	$H_1 : \theta_{bullish} > \theta_{neutral}$	44364	-7.55***	452	292
	$H_1 : \theta_{neutral} > \theta_{bearish}$	40515	-6.00***	292	380

Table 3.3: Mann-Whitney U-test estimates
Mann-Whitney U-test estimates for pairwise significant differences between distribution medians. Under the null hypothesis, the two samples represent two distributions with equal median values. Statistical significance at the 99%, 95%, and 90% level is indicated with ***, **, *, respectively.

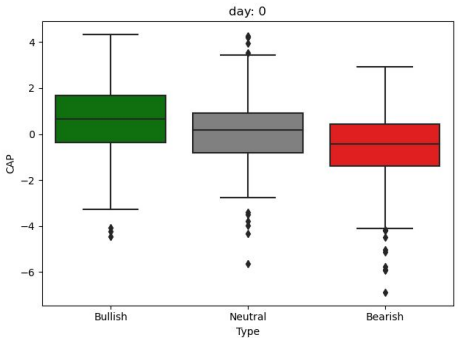


CAR 5 days before the event

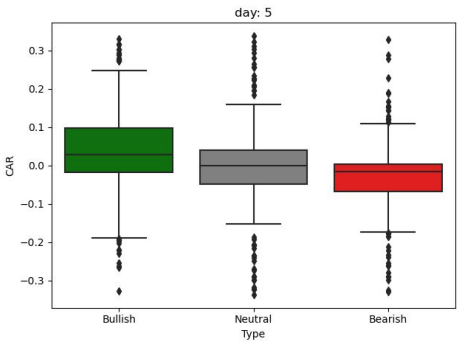
CAP 5 days before the event



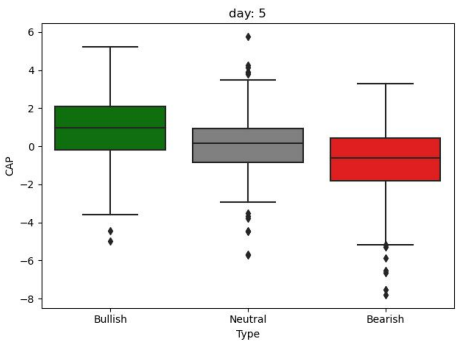
CAR on the event date



CAP on the event date



CAR 5 days after the event



CAP 5 days after the event

Figure 3.19: Distributions of CAP and CAR
Distributions of CAP and CAR 5 days before an event, on event date and 5 days after an event. The line inside a box shows the median while the edges of each box represent the 25% and 75% quantile of the distribution. From the edges of a box, a distance of 1.5 times the interquartile range is measured and a whisker is drawn up to the largest and lowest observed point from the data that falls within this distance.

Our further intuition is that the larger the magnitude of |CAP|, the more likely there will be an event in the holding period. We have no statistical evidence for this, but intuitively, it makes sense to assume that the magnitude |CAP| is a "proxy" for predicting whether an event will happen or not. In that sense, we should better calibrate the thresholds such that the portfolio is zero quite often. Only if we have reason to believe that there will be an event (large magnitude |CAP|) then we invest, according to the sign of CAP. Otherwise we stay out of the market.

The data consists of ticker-date instances (i,t), for the universe of tickers $i = 1,\dots,19$ and dates t ranging through all business days of the sample period, excluding the first 14 days (for the CAP) and the last day (for the last holding period). Every ticker-date instance (i,t) comes with the feature $CAP_{i,t} = \sum_{s=t-14}^{t} AP_{i,s}$, which is the running CAP over the last 14 days plus current day t (we do end of day t rebalancing). Note this is different from the definition of "$CAP_i(\tau,t)$" above.

3.6.1 CAP reset after an event

As CAP continues to shift after an event, for our portfolio construction, we chose to reset it to 0 after every event. This ensures that we are not exposed to short-term reversals. Let $CAP^{(R)}_{i,t}$ be the feature time-serie of the CAP reset to 0 after every event. Formally, let $\tau_{i,t} \le t$ denote the most recent past event date by t of ticker i. Then we define

$$CAP^{(R)}_{i,t} = \sum_{s=\max\{t-14,\tau_{i,t}+1\}}^{t} AP_{i,s} = \begin{cases} CAP_{i,t}, & \text{if } \tau_{i,t} < t-14, \\ \sum_{s=\tau_{i,t}+1}^{t} AP_{i,s}, & \text{if } t-14 \le \tau_{i,t} < t, \\ 0, & \text{if } \tau_{i,t} = t, \end{cases}$$

where we used the convention that $\sum_{s=t+1}^{t} \cdot = 0$.

3.6.2 Cross-sectional thresholds

We use time-varying thresholds to build a long and a short portfolio. For every day, we compute the cross-sectional mean and standard deviations of $CAP^{(R)}_{i,t}$ across the 19 tickers. We will use the mean $\pm$ x standard deviations as and up and down thresholds in the portfolio construction, with x as hyperparameter. We define $U_t(x) = \mu_t + x \cdot \sigma_t$ the upper threshold, and $L_t(x) = \mu_t - x \cdot \sigma_t$ the lower threshold, with μ_t and σ_t being the cross-sectional mean and standard deviations, respectively. Figure 3.20 shows the cross-sectional mean and its 99% confidence interval ($x = 2.58$). The mean is well centered at zero, which speaks for our method of computing CAP. It also shows that there are essentially two regimes, with a switch in early 2015. In the first regime the standard deviation is much larger (and more volatile) than in the second regime. The Appendix contains the results for 95% ($x = 1.96$) and 99.5% confidence intervals ($x = 2.81$).

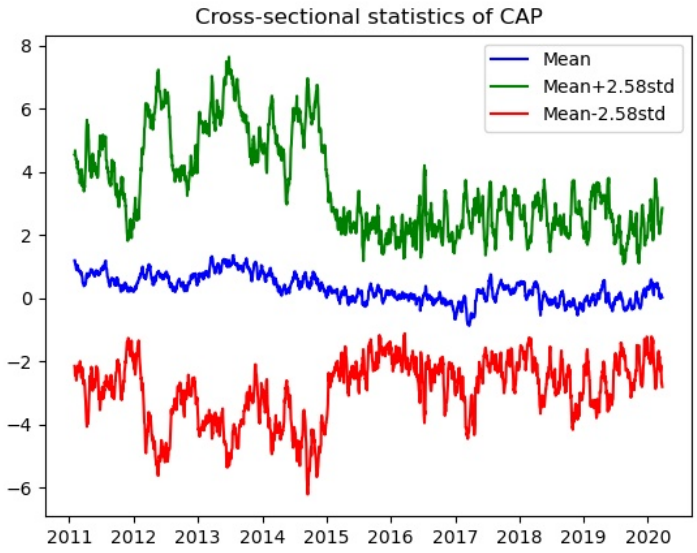


Figure 3.20: Cross-sectional statistics of $CAP^{(R)}$.
The blue lines shows the cross-sectional mean of the CAP across the 19 tickers over the year. The green and red lines show the up and down 99% confidence band. We use them as up and down thresholds for the portfolio construction.

3.6.3 Portfolio and returns

We look at two types of portfolios: long only, and short only that we hold for one day.

Denote by $R_{i,t+1} = (S_{i,t+1} - S_{i,t})/S_{i,t} - R_{f,t}$ the excess return of ticker i over $[t, t+1]$.

Long only: At the end of any business day t, we go long all tickers i with $CAP^{(R)}_{i,t} > U_t(x)$. Denote by $I^{long}_t = \{i \mid CAP^{(R)}_{i,t} > U_t(x)\}$ the corresponding index set, which could be empty. We then form an equally weighted portfolio and consider the 1-day excess return

$$R^{long}_{t,t+1} = \begin{cases} \frac{1}{|I^{long}_t|} \sum_{i \in I^{long}_t} R_{i,t,t+1}, & \text{if } I^{long}_t \neq \emptyset. \\ 0, & \text{if } I^{long}_t = \emptyset. \end{cases}$$

Short only: At the end of any business day t, we go long all tickers i with $CAP^{(R)}_{i,t} < L_t(x)$. Denote by $I^{short}_t = \{i \mid CAP^{(R)}_{i,t} < L_t(x)\}$ the corresponding index set, which could be empty. We then form an equally weighted portfolio and consider the 1-day excess return

$$R^{short}_{t,t+1} = \begin{cases} \frac{1}{|I^{short}_t|} \sum_{i \in I^{short}_t} R_{i,t,t+1}, & \text{if } I^{short}_t \neq \emptyset. \\ 0, & \text{if } I^{short}_t = \emptyset. \end{cases}$$

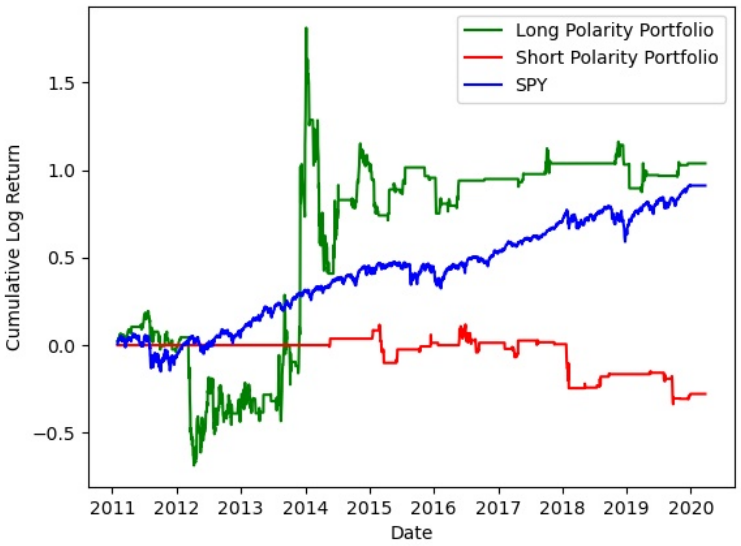
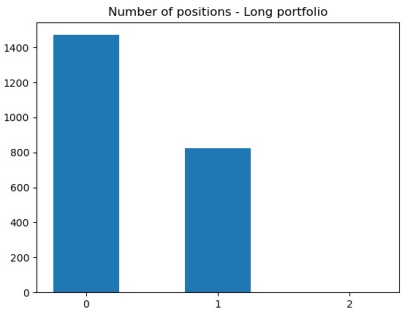


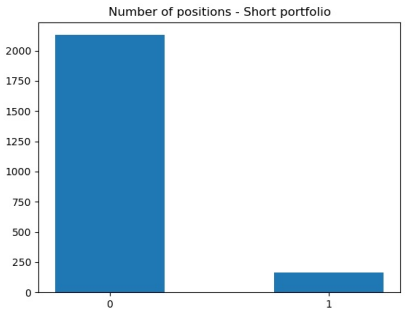
Figure 3.21: Cumulative log returns of both portfolios (x=2.58) and the S&P500.

Now, we compute these returns for all t and plot a time series. What we would like to see, of course, is significantly positive excess returns $R^{long}_{t,t+1}$ and negative excess returns $R^{short}_{t,t+1}$.

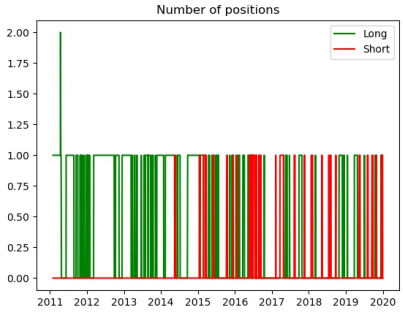
Figure 3.21 shows the cumulative log returns of long and short portfolios as well as the S&P500. Overall, the strategies are correct in both direction : the long portfolio shows positive returns and the short portfolio shows negative returns. Remarkably, cumulative log returns are most of the time shaped as step functions, suggesting that the strategy is able to take positions before an event, and close the position after the return is earned due to the CAP reset. Figure 3.22 shows the number of positions across time of our portfolios. Most of the returns are earned with portfolios consisting of very few tickers. This is a result of our stock picking strategy : we only invest in the top/bottom percentile of CAP, whenever possible.



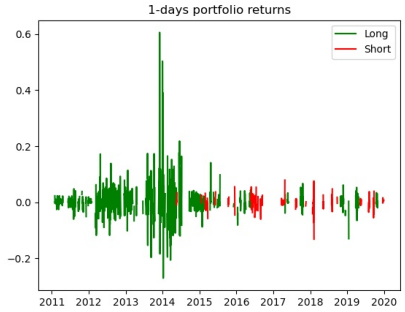
Number of positions - Long portfolio



Number of positions - Short portfolio



Number of positions over time



Daily returns

Figure 3.22: Long and Short portfolios for x=2.58.

3.7 Conclusion

We believe that an accurate and timely estimation of investor sentiment on both firms and aggregate market is an excellent proxy of unobservable firm fundamentals. In particular, recent studies on *nowcasting* shows that alternative sources of data can enhance traditional models of stock return and accounting earnings predictions (Challet and Ayed (2013)). In this paper, we scrape 90 million messages out of StockTwits during 2010 to 2020. Messages are either user-labeled as bullish or bearish or left unlabeled. Using the labeled messages as training set, we build a logistic regression on TFIDF vectorized messages to classify the unlabeled messages in either bullish, neutral or bearish class. We observe a 5-for-1 bullish-to-bearish ratio, indicating that investors are on average optimistic. Then, we build daily time-series of polarity for both individual firms and the aggregate market. We show that changes in daily polarity are strongly associated to changes of the same sign in contemporaneous stock returns, but this result loses its significance against next-day returns. However, focused around specific firm events (defined as sudden peak of message volume on a firm), we show that cumulative abnormal polarity has much more predictive power than cumulative abnormal returns. We also note that user's sentiment about a firm tend to be biased towards recent past events. Finally, as robustness check, we show that event studies on CAAR are consistent with previous literature on EMH.

A Appendix

A.1 Appendix to Chapter 1

A.1.1 Relation between $g_{it}(\tau)$ and $\psi_{it}(\tau)$

The relation between $g_{it}(\tau)$ and $\psi_{it}(\tau)$ is given by :

$$\begin{aligned} g_{it}(\tau) &= \frac{F'_{it}(\tau)}{1-F_{it}(\tau)} \\ &= \psi_{it}(\tau) + \psi'_{it}(\tau)\tau. \end{aligned} \tag{A.1}$$

Proof. The last equation above can be computed with the first derivative of $\psi_{it}(\tau)$ with respect to time. Using definition 1.8 we have :

$$\begin{aligned} \psi'_{it}(\tau) = \frac{u'v - uv'}{v^2} &= \frac{\dfrac{F'_{it}(\tau)}{1-F_{it}(\tau)}\tau + \ln(1-F_{it}(\tau))}{\tau^2} \\ &= \frac{\dfrac{F'_{it}(\tau)}{1-F_{it}(\tau)}}{\tau} + \frac{\ln(1-F_{it}(\tau))}{\tau^2} \\ \Rightarrow \psi'_{it}(\tau)\tau &= \frac{F'_{it}(\tau)}{1-F_{it}(\tau)} + \frac{\ln(1-F_{it}(\tau))}{\tau} \\ \Rightarrow \frac{F'_{it}(\tau)}{1-F_{it}(\tau)} &= \psi_{it}(\tau) + \psi'_{it}(\tau)\tau. \end{aligned}$$

□

A.1.2 Computation of $\psi_{it}(\tau)\tau$

The quantity $\psi_{it}(\tau)\tau$ that we are looking for is :

$$\psi_{it}(\tau)\tau = \int_0^\tau g_{it}(s)ds. \tag{A.2}$$

Proof. Integrating by parts $\psi'_{it}(s)s$ between 0 and τ, we have :

$$\begin{aligned}\int_0^\tau \psi'_{it}(s)\cdot s\cdot ds &= \psi_{it}(s)\cdot s|_{s=0}^{s=\tau} - \int_0^\tau \psi_{it}(s)\cdot 1\cdot ds \\ &= \psi_{it}(\tau)\tau - \int_0^\tau \psi_{it}(s)ds.\end{aligned}$$

Integrating both sides of equation A.1 leads to :

$$\begin{aligned}\int_0^\tau g_{it}(s)ds &= \int_0^\tau \psi_{it}(s)ds + \int_0^\tau \psi'_{it}(s)\cdot s\cdot ds \\ &= \int_0^\tau \psi_{it}(s)ds + \psi_{it}(\tau)\tau - \int_0^\tau \psi_{it}(s)ds \\ &= \psi_{it}(\tau)\tau.\end{aligned}$$

□

A.1.3 Likelihood function

The likelihood function has been developed in Duan et al. (2012). However, the likelihoods have to be slightly updated to be compatible with the neural network framework. That is, (λ, μ) is the set of parameters in the neural network for the forward intensity of default f_{it} and one for h_{it} with the combined exit forward intensity being $g_{it} = f_{it} + h_{it}$. I impose non-negativity on both f_{it} and h_{it} to ensure that the combined exit intensity is at least bigger than the forward default intensity. The overall likelihood function for horizon of prediction τ is by definition given by

$$\mathscr{L}_\tau(\lambda,\mu;\tau_C,\tau_D,X) = \prod_{i=1}^{N}\prod_{t=0}^{T-1}\mathscr{L}_{\tau,i,t}(\lambda,\mu;\tau_{C_i},\tau_{D_i},X_{it}), \tag{A.3}$$

where

$$\begin{aligned}\mathscr{L}_{\tau,i,t}(\lambda,\mu;\tau_{C_i},\tau_{D_i},X_{it}) &= \mathbf{1}_{t_{0_i}\le t,\tau_{C_i}>t+\tau+1}\cdot\mathbb{P}_t(\tau_{C_i}>t+\tau+1) \\ &+ \mathbf{1}_{t_{0_i}\le t,\tau_{D_i}=\tau_{C_i}\le t+\tau}\cdot\mathbb{P}_t(t+\tau<\tau_{D_i}=\tau_{C_i}\le t+\tau+1) \\ &+ \mathbf{1}_{t_{0_i}\le t,\tau_{C_i}\le t+\tau,\tau_{D_i}\neq\tau_{C_i}}\cdot\mathbb{P}_t(t+\tau<\tau_{C_i}\neq\tau_{C_i}\le t+\tau+1) \\ &+ \mathbf{1}_{t_{0_i}>t} + \mathbf{1}_{\tau_{C_i}\le t},\end{aligned} \tag{A.4}$$

with

$$\mathbb{P}_t(\tau_{C_i} > t+\tau+1) = \exp(-\sum_{s=0}^{\tau} g_{it}(s)\Delta t), \tag{A.5}$$

$$\mathbb{P}_t(t+\tau < \tau_{D_i} = \tau_{C_i} \leq t+\tau+1) = \begin{cases} 1-\exp(-f_{it}(0)\Delta t), & \text{if } \tau_{C_i} = t+1 \\ \exp(-\sum_{s=0}^{\tau_{C_i}-t-2} g_{it}(s)\Delta t)\times & \\ \quad (1-\exp[-f_{it}(\tau_{C_i}-t-1)\Delta t], & \text{otherwise} \end{cases} \tag{A.6}$$

$$\mathbb{P}_t(t+\tau < \tau_{C_i} \neq \tau_{C_i} \leq t+\tau+1) = \begin{cases} 1-\exp(-g_{it}(0)\Delta t)- & \\ \quad (1-\exp(-f_{it}(0)\Delta t)), & \text{if } \tau_{C_i} = t+1 \\ \exp(-\sum_{s=0}^{\tau_{C_i}-t-2} g_{it}(s)\Delta t)\times & \\ \quad (\exp[-f_{it}(\tau_{C_i}-t-1)\Delta t- & \\ \quad \exp[-g_{it}(\tau_{C_i}-t-1)\Delta t], & \text{otherwise} \end{cases} \tag{A.7}$$

Since the indicator functions are all mutually exclusive, taking the log of the likelihood function is very helpful. After the log-linearization, the product terms become summation terms, and the two last indicator functions drop. We are left with the summation of the indicator functions times the logarithm of each probability defined above. Similar to the proposition 2 in Duffie et al. (2007) and subsection 3.2 in Duan et al. (2012), the pseudo log-likelihood is the product of seperate terms which are function of f and g. The first decomposition consists of separating terms involving f and terms involving g. The second decomposition consists on separating terms corresponding to different τ. In the end, we get two likelihood functions to estimate for each horizon. For each horizon the parameters of the likelihood function involving f are estimated in a neural network with output $N_{it}^{(\lambda)}$. g is assumed to be of the form $f+h$ as in previous literature. The parameters of the likelihood function involving h are estimated in a neural network with output $N_{it}^{(\mu)}$. Since $\exp(-f)-\exp(-g) = \exp(-f)-\exp(-f-h) = \exp(-f)*(1-\exp(-h))$, the first decomposition is the following :

$$\begin{aligned} \mathscr{L}_{\tau,i,t}(\lambda;\tau_{C_i},\tau_{D_i},X_{it}) = {} & \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}>t+\tau+1}\cdot\exp(-\sum_{s=0}^{\tau} f_{it}(s)\Delta t) \\ & + \mathbf{1}_{t_{0_i}\leq t,\tau_{D_i}=\tau_{C_i}\leq t+\tau}\cdot\exp(-\sum_{s=0}^{\tau_{C_i}-t-2} f_{it}(s)\Delta t)\cdot(1-\exp(-f_{it}(\tau_{C_i}-t-1)\Delta t) \\ & + \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}\leq t+\tau,\tau_{D_i}\neq\tau_{C_i}}\cdot\exp(-\sum_{s=0}^{\tau_{C_i}-t-2} f_{it}(s)\Delta t)\cdot\exp(-f_{it}(\tau_{C_i}-t-1)\Delta t) \\ & + \mathbf{1}_{t_{0_i}>t} + \mathbf{1}_{\tau_{C_i}\leq t}, \end{aligned} \tag{A.8}$$

$$\begin{aligned}\mathscr{L}_{\tau,i,t}(\mu;\tau_{C_i},\tau_{D_i},X_{it}) = {} & \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}>t+\tau+1}\cdot\exp(-\sum_{s=0}^{\tau}h_{it}(s)\Delta t) \\ & + \mathbf{1}_{t_{0_i}\leq t,\tau_{D_i}=\tau_{C_i}\leq t+\tau}\cdot\exp(-\sum_{s=0}^{\tau_{C_i}-t-2}h_{it}(s)\Delta t) \\ & + \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}\leq t+\tau,\tau_{D_i}\neq\tau_{C_i}}\cdot\exp(-\sum_{s=0}^{\tau_{C_i}-t-2}h_{it}(s)\Delta t)\cdot(1-\exp(-h_{it}(\tau_{C_i}-t-1)) \\ & + \mathbf{1}_{t_{0_i}>t}+\mathbf{1}_{\tau_{C_i}\leq t}.\end{aligned} \tag{A.9}$$

Similar to previous literature, we can still decompose the likelihoods into terms involving different horizon of prediction τ. For each τ, we can consider as constant all terms involving previous horizons forward intensities. After log-linearization, the second decomposition is the following :

$$\begin{aligned}\mathscr{L}_{i,t}(\lambda;\tau_{C_i},\tau_{D_i},X_{it}) = {} & \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}>t+\tau+1}\cdot(-f_{it}(s)\Delta t)) \\ & + \mathbf{1}_{t_{0_i}\leq t,\tau_{D_i}=\tau_{C_i}\leq t+\tau}\cdot\ln(1-\exp(-f_{it}(s)\Delta t)) \\ & + \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}\leq t+\tau,\tau_{D_i}\neq\tau_{C_i}}\cdot(-f_{it}(s)\Delta t),\end{aligned} \tag{A.10}$$

$$\begin{aligned}\mathscr{L}_{i,t}(\mu;\tau_{C_i},\tau_{D_i},X_{it}) = {} & \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}>t+\tau+1}\cdot(-h_{it}(s)\Delta t) \\ & + \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}\leq t+\tau,\tau_{D_i}\neq\tau_{C_i}}\cdot\ln(1-\exp(-h_{it}(s)\Delta t)).\end{aligned} \tag{A.11}$$

For all horizon of prediction s. We are now left with many small likelihoods that we can maximize separately instead of the huge maximum likelihood A.3. Ultimately, we can design two neural networks NN_λ and NN_μ with outputs N_{it}^λ and N_{it}^μ respectively to maximize the two above loss functions.

A.1.4 Distance-to-Default estimation

Distance-to-default

Let V_T be the firm value at time T, L the amount of debt to be repaid and E_T the equity value at time T. In case of bankruptcy debt holders receive money before shareholders in case of bankruptcy. The payoff to shareholders is then given by $E_T = \max(V_T - L, 0)$. This payoff is the same as a call option E_T on the underlying V_T with the strike being L. Merton (1974) model considers V_t is following a geometric Brownian motion :

$$dV_t = \mu V_t d_t + \sigma V_t dB_t.$$

We can then use Black-Scholes formula to get the option price and firm's equity value E_t at any time t :

$$E_t = V_t N(d_t) - e^{-r(T-t)} \cdot L \cdot N(d_t - \sigma\sqrt{T-t}), \tag{A.12}$$
$$d_t = \frac{\ln(V_t/L) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}},$$

with r being the risk-free rate and N(x) the normal cumulative distribution function. The distance-to-default is defined as the difference between the expected value of the asset and the default point. After substitution into a normal cumulative distribution function, we get :

$$DtD_t = \frac{\ln(\frac{V_A}{L}) + (\mu - \frac{\sigma_A^2}{2})(T-t)}{\sigma_A\sqrt{T-t}}. \tag{A.13}$$

Note that similarly to the variance restriction method, μ cannot be estimated precisely. Thus, we compute DtD with the following formula :

$$DtD = \frac{\ln(\frac{V_A}{L})}{\sigma_A\sqrt{T-t}}. \tag{A.14}$$

In the figure A.1 taken from Crosbie and Bohn (2003), the distance to default is denoted by "DD". The higher the asset value at horizon H the greater the distance-to-default will get, which lowers the default probability since the firm is more likely to be able to repay the debt owed. Similarly, the more debt the firm has, the smaller the DtD will be for a given level of asset value.

Variance restriction method to estimate DtD

The variance restriction method is used in Duffie et al. (2007) to estimate the distance-to-default (DtD). This method is based on Merton (1974) model which states that firm's equity value can be seen as a call option on the underlying asset and the strike being the amount of debt. This holds because stockholders receive money only once debt holders are fully paid. Applying the Black-Scholes call option formula to equity value, we get the following :

$$\begin{cases} V_E &= V_A N(d_1) - e^{-r(T-t)} \cdot D \cdot N(d_2) \\ d_1 &= \dfrac{\frac{V_A}{D} + (r - \frac{\sigma_A^2}{2}(T-t)}{\sigma_A\sqrt{T-t}} \\ d_2 &= d_1 - \sigma_A\sqrt{T-t}. \end{cases}$$

Where V_E is the market equity value, V_A is the market asset value, D is the default point and N the normal cumulative distribution function. Following KMV assumption, the default point D in the variance restriction method is specified as short-term debt plus one half of long-term debt.

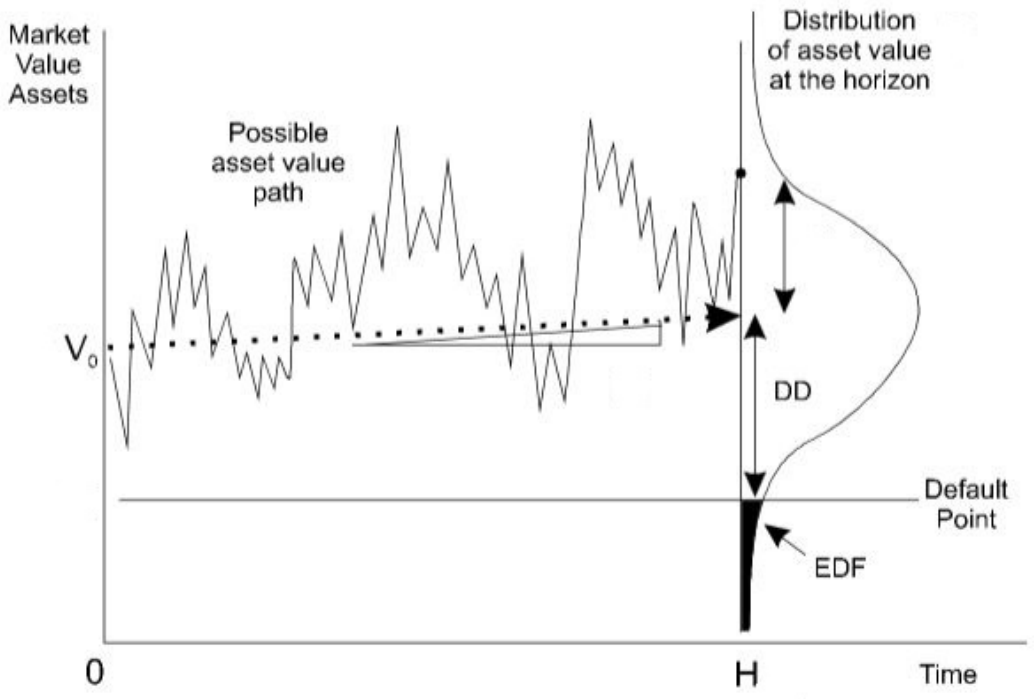


Figure A.1: Distance-to-default

Using Itô, we can show that

$$\sigma_E = \frac{V_A}{V_E} \cdot \frac{\partial V_E}{\partial V_A} \cdot \sigma_A.$$

Hence, the method consists of solving the following system with two equations and two unknowns V_A and σ_A :

$$\begin{cases} V_E &= V_A N(d_1) - e^{-r(T-t)} \cdot D \cdot N(d_2) \\ \sigma_E &= \frac{V_A}{V_E} \cdot \frac{\partial V_E}{\partial V_A} \cdot \sigma_A, \end{cases}$$

Once V_A and σ_A are estimated, the DtD is defined as the distance between the expected value of the asset and the default point. After substitution in a normal CDF, we get

$$DtD = \frac{\ln(\frac{V_A}{L}) + (\mu - \frac{\sigma_A^2}{2})(T-t)}{\sigma_A \sqrt{T-t}}. \tag{A.15}$$

However, many papers agree to say that μ is very tedious to estimate. Hence, DtD is often computed as

$$DtD = \frac{\ln(\frac{V_A}{L})}{\sigma_A \sqrt{T-t}}. \tag{A.16}$$

The major drawback of the variance restriction method used in Duffie et al. (2007) is the definition of the default point. The default point in the variance restriction method is following the so-called KMV assumption. This assumption states that for every firm the default point is exactly equal to short term debt plus one half of long-term debt. However, many financial firms do not account debt as short or long-term debt but as "other liabilities". This causes the default point to be abnormally low for financial firms when using KMV assumption. Therefore, to take financial firms into account, we need to adjust the default point by taking into account other liabilities. The method proposed by Duan et al. (2012) employs a maximum likelihood to estimate the optimal fraction δ of other liabilities to include in the model.

Maximum likelihood estimation to estimate DtD

Duan et al. (2012) and Duan and Wang (2012) presented a method to estimate distance-to-defaults without having to exclude financial firms. The method accounts for other liabilities using a maximum likelihood estimation including a parameter δ to take into account other liabilities. The default point in this method becomes :

$$L = \text{short-term debt} + 0.5 \times \text{long-term debt} + \delta \times \text{other liabilities}. \tag{A.17}$$

Duan et al. (2012) and Duan and Wang (2012) usually estimate δ for many firms altogether (i.e. δ for a whole industry). However, we will improve the methodology by computing δ for each firm individually. We should obtain higher deltas for financial firms than for non-financial firms. Estimating δ for each firm is highly time consuming because of greater computation time but it should highly improve the granularity and precision of the model. The log-likelihood function is given in Duan et al. (2012) and Duan and Wang (2012) by :

$$\begin{aligned}\mathscr{L}_i(\mu,\sigma,\delta) = &-\frac{n-1}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=2}^{n}\ln(\sigma^2 h_t) - \sum_{t=2}^{n}\ln(\frac{\hat{V}_t(\sigma,\delta)}{A_t}) \\ &- \sum_{t=2}^{n}\ln(N(\hat{d}_t(\sigma,\delta))) - \sum_{t=2}^{n}\frac{1}{2\sigma^2 h_t} \times (\ln(\frac{\hat{V}_t(\sigma,\delta)}{\hat{V}_{t-1}(\sigma,\delta)} \cdot \frac{A_{t-1}}{A_t} - (\mu - \frac{\sigma^2}{2}))^2.\end{aligned}$$

where n is the number of period observations for each firm i. The likelihood above differs from Duan et al. (2012) and Duan and Wang (2012) because of the index i since we estimate the likelihood for each of the 2099 firms in our sample to get 2099 estimations of δ. Using this kind of likelihood is complex and very time consuming because we have to solve many inverse Black-Scholes formulas to get the time values of implied asset value $\hat{V}_t(\sigma,\delta)$ for each firm by solving equation A.12. However, inverse Black-Scholes formula doesn't have any closed form solution. The optimization is even more tedious since the implied asset value $\hat{V}_t(\sigma,\delta)$ depends on the final output of the likelihood δ. A_t is the book asset value and h_t is the "length of time between two consecutive equity values measured in trading days as a fraction of a year". h_t in the model is set to be 0.25 since we performed a linear interpolation when we had a missing value in the sample (see "Missing information"). To get rid of the inverse Black-Scholes formula problem, I use a dichotomic algorithm to compute the time series of

asset values.

A.2 Appendix to Chapter 2

A.2.1 Technical proofs

Proof of Lemma 1

From the definition of DIG, we know that for any $R_k \notin \mathscr{PA}_j$, $I(R_k \to R_j || \mathscr{R}_{-\{k,j\}}) = 0$. This implies that for all t,

$$p(R_{j,t}|\mathscr{R}^{t-1}) = p(R_{j,t}|\mathscr{R}^{t-1}_{-\{k\}}). \tag{A.18}$$

On the other hand, by the assumption of the Lemma, $R_i \notin \mathscr{PA}_j$, we have

$$I(R_i \to R_j || \mathscr{R}_{-\{i,j\}}) = 0, \tag{A.19}$$

or equivalently, for all t,

$$p(R_{j,t}|\mathscr{R}^{t-1}) = p(R_{j,t}|\mathscr{R}^{t-1}_{-\{i\}}). \tag{A.20}$$

Combining (A.18) and (A.20) imply that for any pair $\{R_i, R_k\}$ that are not in the parent set of R_j, we have

$$p(R_{j,t}|\mathscr{R}^{t-1}_{-\{k\}}) = p(R_{j,t}|\mathscr{R}^{t-1}_{-\{i\}}). \tag{A.21}$$

To prove the claim of this lemma, we use (A.21) to show that all the time series in $\mathscr{R}_{-\{i,j\}} \setminus \mathscr{C}$ can be removed from the conditioning in (A.19). Let $R_k \in \mathscr{R}_{-\{i,j\}} \setminus \mathscr{C}$, by multiplying the above equality with $p(R_i^{t-1}|\mathscr{R}^{t-1}_{-\{i,k\}})$ and marginalizing over R_i^{t-1}, we obtain

$$\begin{aligned}
&\int p(R_{j,t}|\mathscr{R}^{t-1}_{-\{k\}}) p(R_i^{t-1}|\mathscr{R}^{t-1}_{-\{i,k\}}) dR_i^{t-1} = p(R_{j,t}|\mathscr{R}^{t-1}_{-\{i,k\}}) \\
&= \int p(R_{j,t}|\mathscr{R}^{t-1}_{-\{i\}}) p(R_i^{t-1}|\mathscr{R}^{t-1}_{-\{i,k\}}) dR_i^{t-1} = p(R_{j,t}|\mathscr{R}^{t-1}_{-\{i\}}).
\end{aligned}$$

The above equalities and (A.20) imply that for all t,

$$p(R_{j,t}|\mathscr{R}^{t-1}) = p(R_{j,t}|\mathscr{R}^{t-1}_{-\{i,k\}}),$$

or equivalently,

$$I(R_i \to R_j || \mathscr{R}_{-\{i,j,k\}}) = 0.$$

By repeating the above procedure, we obtain

$$I(R_i \to R_j || \mathscr{C}) = 0.$$

Proof of Lemma 2

Consider the VAR model in (2.14). First, we assume that X_i has no influence on X_j, i.e., $I(X_i \to X_j || \mathscr{X}_{-\{i,j\}}) = 0$ or equivalently $a_{j,i} = 0$ and show that (2.17) holds. Given this assumption, we have that for all t,

$$p(X_{j,t}|\mathscr{X}^{t-1}_{-\{i\}}) = p(X_{j,t}|\mathscr{X}^{t-1}).$$

Using the equations in (2.14) and the assumption that $a_{j,i} = 0$, we obtain

$$\begin{aligned}
&p(X_{j,t}|\mathscr{X}^{t-1}) = p(N_{j,t} + \sum_k a_{j,k} X_{k,t-1}|\mathscr{X}^{t-1}) = p(N_{j,t} + \sum_{k\neq i} a_{j,k} X_{k,t-1}|\mathscr{X}^{t-1}) \\
&= p(N_{j,t} + \sum_{k\neq i} a_{j,k} X_{k,t-1}| \sum_{k\neq i} a_{j,k} X_{k,t-1}, X_i^{t-1}, X_j^{t-1}) \\
&= p(N_{j,t} + \sum_{k\neq i} a_{j,k} X_{k,t-1}| \sum_{k\neq i} a_{j,k} X_{k,t-1}, X_j^{t-1}).
\end{aligned}$$

Note that we could replace $\mathscr{X}^{t-1}$ by $\{\sum_{k\neq i} a_{j,k} X_{k,t-1}, X_i^{t-1}, X_j^{t-1}\}$ or $\{\sum_{k\neq i} a_{j,k} X_{k,t-1}, X_j^{t-1}\}$ in the above equations, because given either of them $\sum_{k\neq i} a_{j,k} X_{k,t-1}$ becomes a constant and independent of $N_{j,t}$. By defining $Q_{t-1} := \sum_{k\neq i} a_{j,k} X_{k,t-1}$, the above equations can be rewritten as follows

$$p(X_{j,t}|\mathscr{X}^{t-1}) = p(X_{j,t}|Q_{t-1}, X_i^{t-1}, X_j^{t-1}) = p(X_{j,t}|Q_{t-1}, X_j^{t-1}), \forall\, t,$$

or equivalently,

$$\mathbb{E}\left[\log \frac{p(X_{j,t}|Q_{t-1}, X_i^{t-1}, X_j^{t-1})}{p(X_{j,t}|Q_{t-1}, X_j^{t-1})}\right] = 0, \forall\, t.$$

Using the definition of DI, the above equalities can be written in terms of DI as follows

$$I(X_i \to X_j || Q) = 0.$$

On the other hand, we have

$$[a_{j,1}, ..., a_{j,i-1}, a_{j,i+1}, ..., a_{j,m}] = \arg\min_{\mathbf{w}\in\mathbb{R}^{m-1}} \mathbb{E}\left[||X_{j,t} - \mathbf{w}^T \mathbf{X}_{-\{i\},t-1}||_2^2\right] := \mathbf{u}_t.$$

where $\mathbf{X}_{-\{i\},t-1} := [X_{1,t-1}, ..., X_{i-1,t-1}, X_{i+1,t-1}, ..., X_{m,t-1}]^T$. This means that $Q_{t-1} = \mathbf{u}_t^T \mathbf{X}_{-\{i\},t-1}$.

Next, we show the reverse direction, i.e., we assume (2.17) holds, then we show $I(X_i \to$

$X_j||\mathscr{X}_{-\{i,j\}}) = 0$. To do so, it suffices to show $a_{j,i} = 0$. Since (2.17) holds, we have

$$p(X_{j,t}|\mathbf{u}_t^T\mathbf{X}_{-\{i\},t-1}, X_i^{t-1}, X_j^{t-1}) = p(X_{j,t}|\mathbf{u}_t^T\mathbf{X}_{-\{i\},t-1}, X_j^{t-1}), \forall\, t.$$

Using the j-th equation of (2.14) and the above equalities, for any instances $(\mathbf{u}_t^T\mathbf{x}_{-\{i\},t-1}, x_i^{t-1}, x_j^{t-1})$ of $(\mathbf{u}_t^T\mathbf{X}_{-\{i\},t-1}, X_i^{t-1}, X_j^{t-1})$, we obtain $\forall\, t$,

$$\mathbb{E}\left[X_{j,t}|\mathbf{u}_t^T\mathbf{x}_{-\{i\},t-1}, x_i^{t-1}, x_j^{t-1}\right] = \mathbb{E}\left[X_{j,t}|\mathbf{u}_t^T\mathbf{x}_{-\{i\},t-1}, x_j^{t-1}\right],$$

which implies

$$\begin{aligned}&\mathbb{E}\left[N_{j,t}\right] + \sum_{k\neq i} a_{j,k}x_{k,t-1} + a_{j,i}x_{i,t-1} = \\ &\mathbb{E}\left[N_{j,t}\right] + \sum_{k\neq i} a_{j,k}x_{k,t-1} + a_{j,i}\mathbb{E}\left[X_{i,t-1}|\mathbf{u}_t^T\mathbf{x}_{-\{i\},t-1}, x_j^{t-1}\right].\end{aligned}$$

This simplifies to

$$a_{j,i}x_{i,t-1} = a_{j,i}\mathbb{E}\left[X_{i,t-1}|\mathbf{u}_t^T\mathbf{x}_{-\{i\},t-1}, x_j^{t-1}\right], \forall\, t.$$

This equation should hold for any $x_{i,t-1}$. This is only possible if $a_{j,i} = 0$.

Proof of Lemma 3

The proof is similar to the linear version and uses the fact that exogenous noises $\{\varepsilon_{j,t}\}$ are independent. More precisely, we have

$$p(X_{j,t}|\mathscr{X}^{t-1}) = p(F_j(\mathscr{X}^{t-1}) + \varepsilon_{j,t}|F_j(\mathscr{X}^{t-1})).$$

Since there is no influence from X_i to X_j, we can eliminate it from the conditioning and the argument of function F_j and obtain

$$p\big(F_j(\mathscr{X}^{t-1}) + \varepsilon_{j,t}|\mathscr{X}^{t-1}\big) = p\big(F_j(\mathscr{X}_{-\{i\}}^{t-1}) + \varepsilon_{j,t}|\mathscr{X}_{-\{i\}}^{t-1}\big).$$

On the other hand, because given either $\{F_j(\mathscr{X}_{-\{i\}}^{t-1}), X_i^{t-1}, X_j^{t-1}\}$ or $\{F_j(\mathscr{X}_{-\{i\}}^{t-1}), X_j^{t-1}\}$, the value of $F_j(\mathscr{X}_{-\{i\}}^{t-1})$ is no longer a random variable. Using this relationship and the fact that $\varepsilon_{j,t}$ is independent of $\mathscr{X}^{t-1}$, we obtain

$$p\big(F_j(\mathscr{X}_{-\{i\}}^{t-1}) + \varepsilon_{j,t}|F_j(\mathscr{X}_{-\{i\}}^{t-1}), X_i^{t-1}, X_j^{t-1}\big) = p\big(F_j(\mathscr{X}_{-\{i\}}^{t-1}) + \varepsilon_{j,t}|F_j(\mathscr{X}_{-\{i\}}^{t-1}), X_j^{t-1}\big).$$

By defining $Q_{t-1} := F_j(\mathscr{X}_{-\{i\}}^{t-1})$, the above equations can be rewritten in terms of DI as follows,

$$I(X_i \to X_j||Q) = 0.$$

To show the reverse, we need to prove that $I(X_i \to X_j||\mathscr{X}_{-\{i,j\}}) = 0$ if Equation (2.19) holds.

Because $I(X_i \to X_j || Q) = 0$ and using Equation (2.18), for all t, we have

$$p\big(F_j(\mathscr{X}^{t-1}) + \varepsilon_{j,t} | Q_{t-1}, X_i^{t-1}, X_j^{t-1}\big) = p\big(F_j(\mathscr{X}^{t-1}) + \varepsilon_{j,t} | Q_{t-1}, X_j^{t-1}\big),$$

where $Q_{t-1} = F_j(\mathscr{X}_{-\{i\}}^{t-1})$. Note that the conditioning on the right-hand-side distribution is independent of X_i^{t-1}. This implies that function F_j does not depend on X_i. Therefore, we can remove X_i^{t-1} from the argument of F_j, i.e.,

$$X_{j,t} = F_j(\mathscr{X}^{t-1}) + \varepsilon_{j,t} = F_j(\mathscr{X}_{-\{i\}}^{t-1}) + \varepsilon_{j,t},$$

which further implies

$$p(F_j(\mathscr{X}_{-\{i\}}^{t-1}) + \varepsilon_{j,t} | \mathscr{X}_{-\{i,j\}}^{t-1}, X_i^{t-1}, X_j^{t-1}) = p(F_j(\mathscr{X}_{-\{i\}}^{t-1}) + \varepsilon_{j,t} | \mathscr{X}_{-\{i,j\}}^{t-1}, X_j^{t-1}).$$

This is equivalent to

$$I(X_i \to X_j || \mathscr{X}_{-\{i,j\}}) = 0.$$

A.2.2 k-Nearest Neighbors estimator of mutual information

Suppose that $N+M$ i.i.d. realizations $\{\mathbf{x}_1, ..., \mathbf{x}_{N+M}\}$ are available from $\mathbb{P}_{X,Y,Z}$, where $\mathbf{x}_i$ denotes the i-th realization of (X, Y, Z). The data is randomly divided into two subsets $\mathscr{S}_1$ and $\mathscr{S}_2$ of N and M points, respectively. The estimator has two main stages: In the first stage, a k-nearest density estimator $\widehat{\mathbb{P}}_{X,Y,Z}$ at the N points of $\mathscr{S}_1$ is estimated using the M realizations of $\mathscr{S}_2$ as follows:

Let $d(\mathbf{x}, \mathbf{y}) \in \mathbb{R}$ denote the Euclidean distance between points $\mathbf{x}$ and $\mathbf{y}$ and $d_k(\mathbf{x}) \in \mathbb{R}$ denotes the Euclidean distance between a point $\mathbf{x}$ and its k-th nearest neighbor among $\mathscr{S}_2$. The k-nearest region is $\mathscr{S}_k(\mathbf{x}) := \{\mathbf{y} : d(\mathbf{x}, \mathbf{y}) \leq d_k(\mathbf{x})\}$ and the volume of this region is $V_k(\mathbf{x}) := \int_{\mathscr{S}_k(\mathbf{x})} 1 \, dv$. The standard k-nearest density estimator of Sricharan et al. (2011) is defined as

$$\widehat{\mathbb{P}}_{X,Y,Z}(\mathbf{x}) := \frac{k-1}{M \cdot V_k(\mathbf{x})}.$$

Similarly, we obtain k-nearest density estimators $\widehat{\mathbb{P}}_{X,Z}, \widehat{\mathbb{P}}_{Y,Z}$, and $\widehat{\mathbb{P}}_Z$. Subsequently, the N samples of $\mathscr{S}_1$ is used to approximate the conditional mutual information:

$$\widehat{I}(X; Y | Z) := \frac{1}{N} \sum_{i \in \mathscr{S}_1} \log \widehat{\mathbb{P}}_{X,Y,Z}(\mathbf{x}_i) + \log \widehat{\mathbb{P}}_Z(\mathbf{x}_i) - \log \widehat{\mathbb{P}}_{X,Z}(\mathbf{x}_i) - \log \widehat{\mathbb{P}}_{Y,Z}(\mathbf{x}_i).$$

For more details corresponding this estimator including its bias, variance, and confidence, please see the works by Sricharan et al. (2011) and Loftsgaarden et al. (1965).

A.2.3 Koopman-based Lifting Method

Let $\mathbf{X}_t := \{X_{1,t}, ..., X_{m,t}\}$ denote a network of m time series such that

$$\dot{\mathbf{X}}_t = F(\mathbf{X}_t), \tag{A.22}$$

where the vector field $F(\mathbf{X}) = (F_1(\mathbf{X}), ..., F_m(\mathbf{X}))$ is of the form

$$F_j(\mathbf{X}) = \sum_{k=1}^{K} w_{j,k} h_k(\mathbf{X}). \tag{A.23}$$

In the above equation, $w_{j,k} \in \mathbb{R}$ are unknown weights and $\{h_k(\mathbf{X})\}$ denote a set of known library functions, e.g., monomials. Furthermore, let $\varphi^t(\mathbf{X}_0)$ denote the solution to (A.22) associated with the initial condition $\mathbf{X}_0$.

Now, suppose that we have N noisy observations $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_N, \mathbf{y}_N)\}$ of the system trajectory, where $\mathbf{x}_i$ is the initial point and $\mathbf{y}_i$ is the final point after T_s steps, i.e.,

$$\mathbf{y}_i - \epsilon_i = \varphi^{T_s}(\mathbf{x}_i - \varepsilon_i), \quad i = 1, ..., N,$$

where ϵ_i and ε_i are the measurement noises. The goal is to estimate the weights $\{w_{j,k}\}$ using these observations and consequently infer the causal network among the time series. To do so, we use the Koopman approach Mauroy and Goncalves (2019) that lifts the observation space to another space in which the relationships are linear. More precisely, the steps are as follows:

- Select a set of M basis lifting functions $\{p_1(\mathbf{x}), ..., p_M(\mathbf{x})\}$, and lift the observations,

$$\mathbf{P}_x := \begin{pmatrix} p_1(\mathbf{x}_1) & \cdots & p_M(\mathbf{x}_1) \\ p_1(\mathbf{x}_2) & \cdots & p_M(\mathbf{x}_2) \\ \vdots & \ddots & \vdots \\ p_1(\mathbf{x}_N) & \cdots & p_M(\mathbf{x}_N) \end{pmatrix}, \mathbf{P}_y := \begin{pmatrix} p_1(\mathbf{y}_1) & \cdots & p_M(\mathbf{y}_1) \\ p_1(\mathbf{y}_2) & \cdots & p_M(\mathbf{y}_2) \\ \vdots & \ddots & \vdots \\ p_1(\mathbf{y}_N) & \cdots & p_M(\mathbf{y}_N) \end{pmatrix}. \tag{A.24}$$

- Identify the Koopman operator $\mathbf{L} := \frac{1}{T_s} \log(\mathbf{P}_x^\dagger \mathbf{P}_y)$, where $\mathbf{P}_x^\dagger$ denotes the pseudo-inverse of $\mathbf{P}_x$ and the function log denotes the (principal) matrix logarithm.

- Identify the weights using the following equations: $\widehat{w}_{k,j} := [\mathbf{L}]_{k,l}$, with l such that $p_l(\mathbf{x}) = x_j$, where $\mathbf{x} = (x_1, ..., x_m)$.

An alternative approach to obtain the weights is the dual lifting method which executes the following steps instead of the above last step. At first, it finds matrix $\widehat{\mathbf{F}}$ using the following

equation,

$$\widehat{\mathbf{F}}_{N\times m} := \mathbf{L}_{N\times N} \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_N^T \end{pmatrix}_{N\times m}.$$

Next, it constructs

$$\mathbf{H}_x := \begin{pmatrix} p_1(\mathbf{x}_1) & \cdots & p_M(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ p_1(\mathbf{x}_N) & \cdots & p_M(\mathbf{x}_N) \end{pmatrix}_{N\times M},$$

and for each j, solve the following regression problem to get the weights

$$\widehat{\mathbf{w}}_j := \arg\min_{\mathbf{w}\in\mathbb{R}^M} \left|\left|\mathbf{H}_x\mathbf{w} - \widehat{\mathbf{F}}_{:,j}\right|\right|_2^2 + \rho||\mathbf{w}||_1,$$

where $\widehat{\mathbf{F}}_{:,j}$ denotes the j-th column of matrix $\widehat{\mathbf{F}}$ and $\widehat{\mathbf{w}}_j = [\hat{w}_{j,1}, ..., \hat{w}_{j,M}]^T$.

A.2.4 Ideal portfolio

In this section, we show how the ideal portfolio is related to the coefficients of the linear system in (2.14). Recall the optimization problem in Lemma 2.

$$\mathbf{u}_t := \arg\min_{\mathbf{w}\in\mathbb{R}^{m-1}} \mathbb{E}\left[||X_{j,t} - \mathbf{w}^T\mathbf{X}_{-\{i\},t-1}||_2^2\right],$$
$$\mathbf{X}_{-\{i\},t-1} := [X_{1,t-1}, \cdots, X_{i-1,t-1}, X_{i+1,t-1}, \cdots, X_{m,t-1}]^T.$$

Consider the j-th Equation in (2.14), i.e.,

$$X_{j,t} = \sum_{k=1}^{m} a_{j,k}X_{k,t-1} + N_{j,t}.$$

If $a_{j,i} = 0$, by substituting the above equation into the optimization, we obtain

$$\begin{aligned}
&\min_{\mathbf{w}\in\mathbb{R}^{m-1}} \mathbb{E}\left[||\sum_{k\neq i}(a_{j,k} - w_k)X_{k,t-1} + N_{j,t}||_2^2\right] = \min_{\mathbf{w}\in\mathbb{R}^{m-1}} \mathbb{E}\left[||N_{j,t}||_2^2\right] \\
&+ \mathbb{E}\left[||\sum_{k\neq i}(a_{j,k} - w_k)X_{k,t-1}||_2^2\right] + 2\mathbb{E}\left[||\left(\sum_{k\neq i}(a_{j,k} - w_k)X_{k,t-1}\right)N_{j,t}||_2^2\right] \\
&= \min_{\mathbf{w}\in\mathbb{R}^{m-1}} \mathbb{E}\left[||\sum_{k\neq i}(a_{j,k} - w_k)X_{k,t-1}||_2^2\right] + \mathbb{E}\left[||N_{j,t}||_2^2\right]
\end{aligned}$$

The last equality is due to the fact $N_{j,t}$ is independent of $\{X_{k,t-1}\}$ and have zero mean. This implies that the solution is $w_k = a_{j,k}$ for $k \in \{1, ..., i-1, i+1, ..., m\}$.

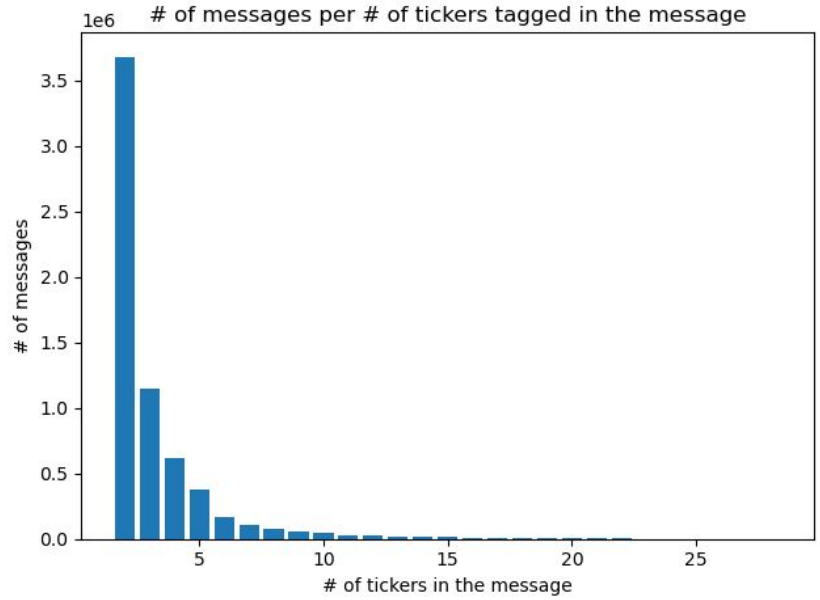


Figure A.2: Number of tickers per message
Histogram of the number of tickers per message, across all messages referring to more than one ticker. The maximum number of tickers per message amounts to 28 and corresponds to 11 messages in the sample.

A.3 Appendix to Chapter 3

A.3.1 Number of messages

As the same information sometimes refer to several companies, users are allowed to identify more than one ticker per message. Figure A.2 shows the histogram of the number of tickers tagged per message. As the vast majority of message includes only one ticker, we only show on this plot messages referring to more than one ticker. However, many messages refer to several tickers and this creates duplicates in the database because we consider the same message for all tickers tagged in the message. Figure A.3 shows the number of messages with and without double counting. In our sample, the number of messages without double counting is 76 million, as opposed to 90 million messages with double counting. Figure A.4 shows the ratio between the number of messages with double counting divided by the number of messages without double counting. Throughout this paper, we only refer to the number of messages with double counting.

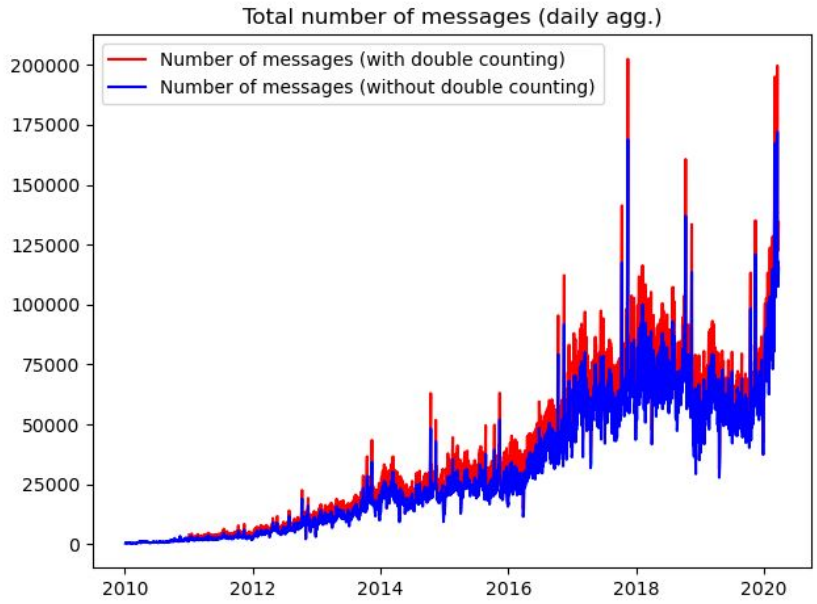


Figure A.3: Number of messages
Total number of messages with double counting (red) compared to the total number of messages without double counting (blue). Numbers are aggregated daily.

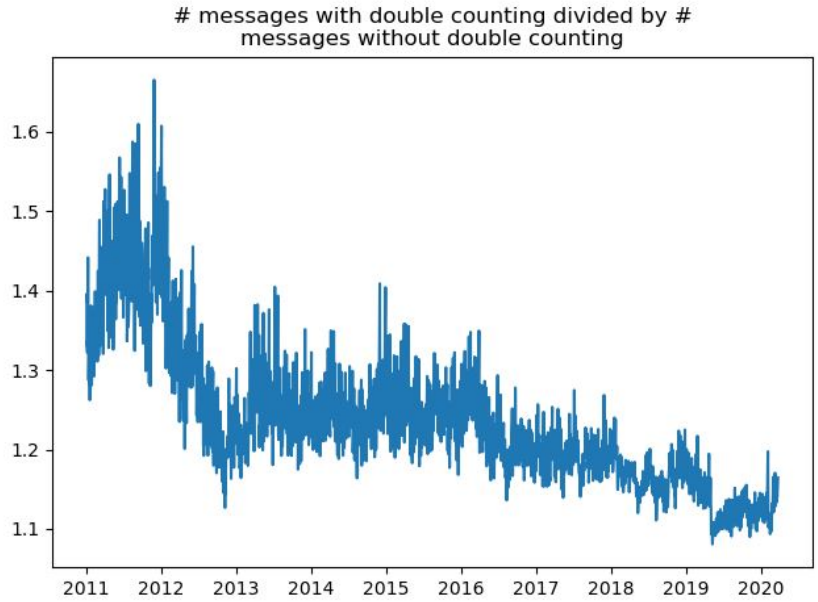


Figure A.4: Number of messages II
Ratio between the number of messages with double counting and the number of messages without double counting. Numbers are aggreaged daily.

A.3.2 Coverage

Stocktwits is neither regulated nor moderated, so one needs to filter the information that we use. Even if Stocktwits has valuable information from respected contributors, a blog[1] describes the concerns that may rise when using Stocktwits as a financial information provider, namely self-promotion, lack of credibility and other noise. To diversify noise and better extract information, we exclude from our sample tickers that are rarely discussed. Thereto, we compute the median of daily message volume for each ticker and exclude from our sample tickers with a median of less than 50. Decreasing the median threshold increases the coverage at the expense of more noise in the daily polarity. Figure A.5 shows the coverage as a function of the median threshold. To increase the coverage we need to decrease the threshold a lot (e.g., decreasing the median threshold to 40 from 50 would increase the number of tickers covered to merely 22 from 19). We chose a median threshold of 50 as a balanced trade-off between noise and coverage. Table A.1 shows the list of tickers covered and associated market capitalization as of 31st of December 2019. To avoid bias in the polarity, we restrict the universe and focus on the most discussed tickers. On top of that, it appears that the most discussed tickers are not always big firms, so it is has the advantage of covering all sizes of firm and we avoid big-firm bias. Also, we do not consider only firms but also ETFs on alternative investments. Finally, we cover several sectors so even if we have a restrictive universe, it is well diversified.

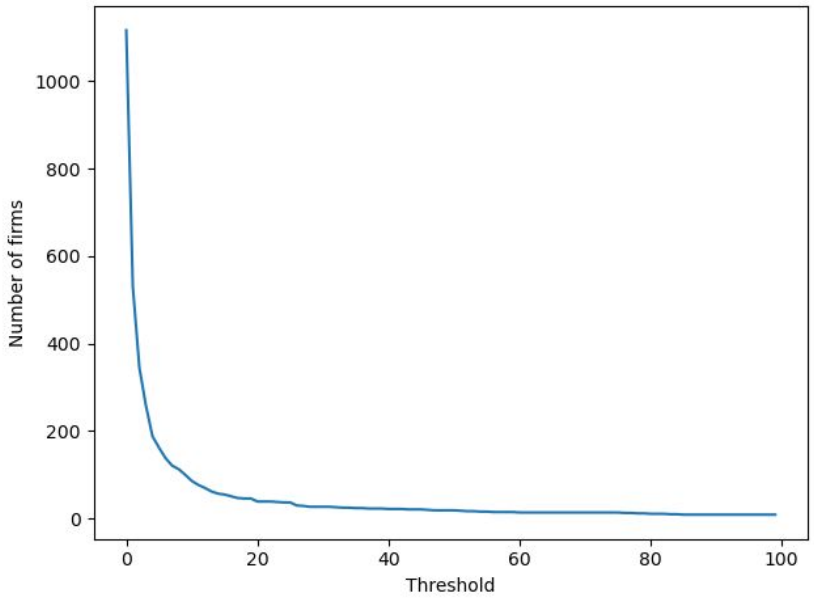


Figure A.5: Coverage
Coverage as a function of the median threshold. A lower threshold increases the coverage at the expense of a bigger bias in the polarity.

[1] https://www.warriortrading.com/stocktwits-review

Ticker	Name	Market capitalization
AAPL	Apple	1287
AMD	Advanced Micro Devices	53
AMRN	Amarin	7
AMZN	Amazon	920
BABA	Alibaba	571
BAC	Bank of America	311
BB	BlackBerry	4
FB	Facebook	585
GLD	Gold ETF	59
IWM	Small-Cap ETF	55
JNUG	Direxion	0.5
MNKD	MannKind Corporation	0.2
NFLX	Netflix	142
PLUG	Plug Power	1
QQQ	Nasdaq100 ETF	134
SPY	S&P500 ETF	391
TSLA	Tesla	76
TWTR	Twitter	25
UVXY	VIX ETF	0.8

Table A.1: Coverage
Coverage after the trimming process. List of tickers and corresponding market capitalization as of 31st of December 2019.

A.3.3 Tutorial for StockTwits messages extraction

We use CRSP to get stock prices of all US and Canadian listed firms from 1990 to 2020. Out of this dataset, we create the list of unique tickers for which we will extract messages. We will later be able to merge the two datasets using the date and ticker for every observation. We use the StockTwits Application Programming Interface (API) to collect messages from StockTwits. One query on StockTwits API is called a JavaScript Object Notation (JSON) request. Every message on StockTwits has a unique identifier ("msg_id") posted by a user with a unique identifier ("user_id"). JSON requests allow to query the database by ticker (called "symbol method") or by user (called "user method"). We use the query by ticker. One query only outputs the latest 30 messages concerning that ticker. However, it is possible to set a parameter ("max") to output the latest 30 messages up to this particular message identifier. This parameter allows us to crawl the message history of a ticker by recursively changing the "max" parameter to the oldest message identifier in the query. To perform a JSON request for Apple (AAPL) up to the message identifier 30'000'000, simply enter the following URL in a browser : https://api.stocktwits.com/api/2/streams/symbol/AAPL.json?&max=30000000. The page we get looks unreadable but it has always the same structure : several pairs of keys and values. The structure of JSON can easily be interpreted by modern programming languages. We create a Python script to query the API and extract the message history of every ticker in the ticker list. We store the output of every JSON request in .txt files in dedicated ticker folders.

A.3.4 Portfolio construction - various x

As the thresholds $U_t(x)$ and $L_t(x)$ are functions of the hyperparameter x, we provide for robustness check the results of our portfolio construction with x = 1.96 (95% confidence band) and x = 2.81 (99.5% confidence band).

x = 1.96

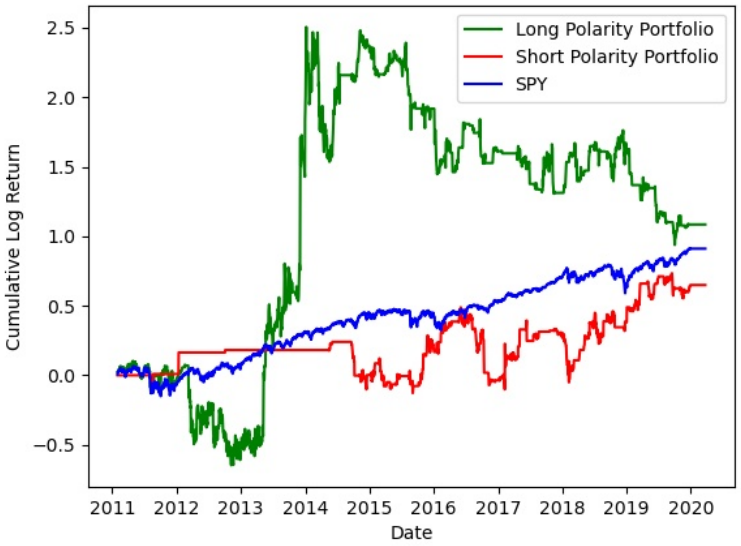
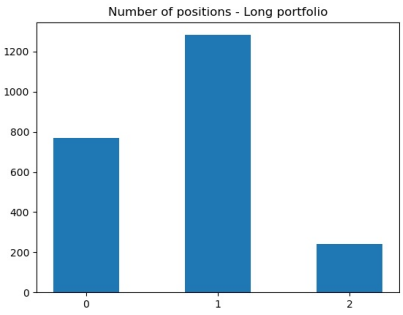
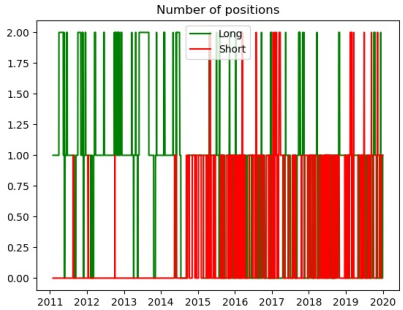


Figure A.6: Cumulative log returns of both portfolios (x=1.96) and the S&P500.

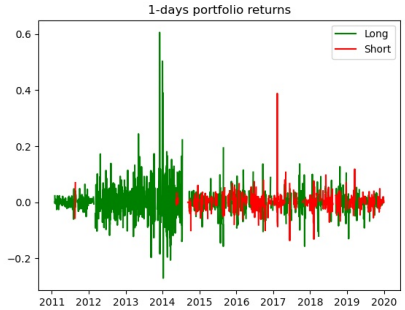


Number of positions - Long portfolio

Number of positions - Short portfolio



Number of positions over time



Daily returns

Figure A.7: Long and Short portfolios for x = 1.96.

x = 2.81

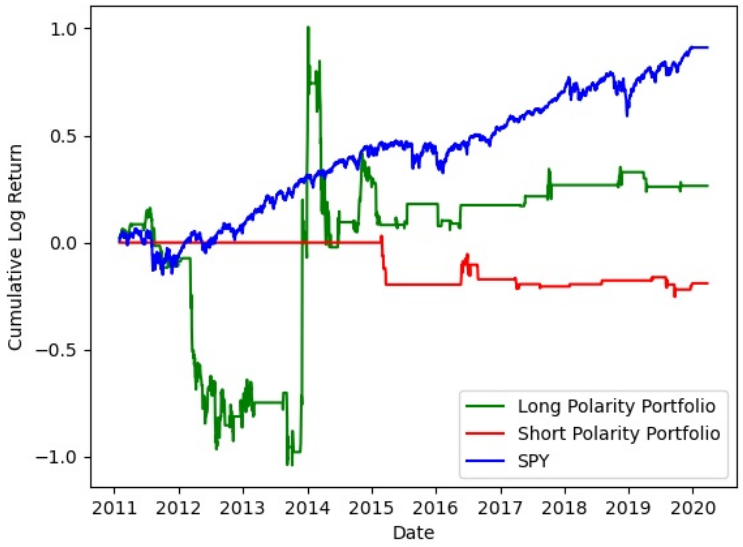
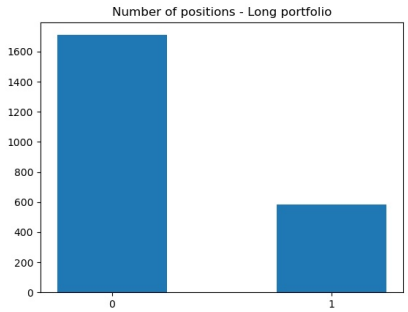
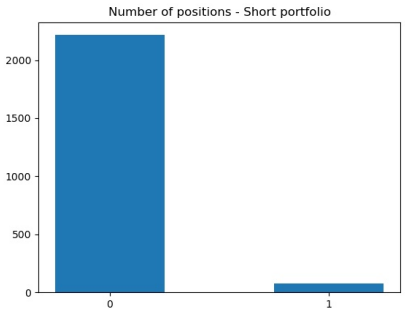


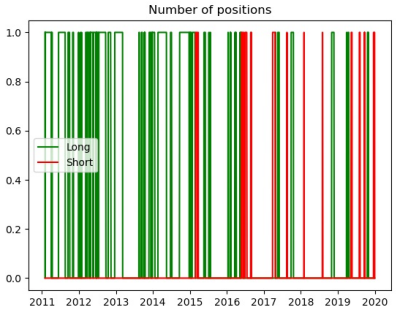
Figure A.8: Cumulative log returns of both portfolios (x=2.81) and the S&P500.



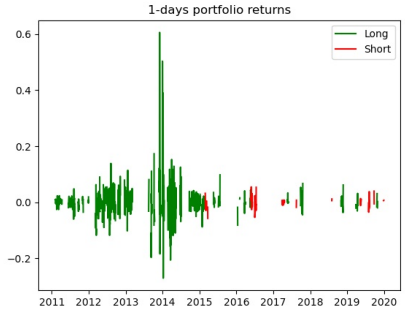
Number of positions - Long portfolio



Number of positions - Short portfolio



Number of positions over time



Daily returns

Figure A.9: Long and Short portfolio for x=2.81.

Bibliography

Allen, F., Babus, A., and Carletti, E. (2010). Financial connections and systemic risk. Technical report, National Bureau of Economic Research.

Altinbas, H. and Biskin, O. T. (2015). Selecting macroeconomic influencers on stock markets by using feature selection algorithms. *Procedia Economics and Finance*.

Altman, E., Kimura, H., and Barboza, F. (2017). Machine learning models and bankruptcy prediction. *Expert Systems with Applications*.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*.

Antweiler, W. and Frank, M. Z. (2004). Is all that talk just noise? the information content of internet stock message boards. *The Journal of finance*.

Barigozzi, M. and Hallin, M. (2016). A network analysis of the volatility of high dimensional financial series. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*.

Bernardi, M. and Costola, M. (2019). High-dimensional sparse financial networks through a regularised regression model. *SAFE Working Paper*.

Bianchi, D., Billio, M., Casarin, R., and Guidolin, M. (2019). Modeling systemic risk with markov switching graphical sur models. *Journal of Econometrics*.

Billio, M., Casarin, R., and Rossini, L. (2019). Bayesian nonparametric sparse var models. *Journal of Econometrics*.

Billio, M., Getmansky, M., Lo, A. W., and Pelizzon, L. (2010). Measuring systemic risk in the finance and insurance sectors. *MIT Sloan School of Management Working Paper*.

Billio, M., Getmansky, M., Lo, A. W., and Pelizzon, L. (2012). Econometric measures of connectedness and systemic risk in the finance and insurance sectors. *Journal of Financial Economics*.

Bonaccolto, G., Caporin, M., and Panzica, R. (2019). Estimation and model-based combination of causality networks among large us banks and insurance companies. *Journal of Empirical Finance*.

Cecchini, M., Aytug, H., Koehler, G. J., and Pathak, P. (2010). Making words work: Using financial text as a predictor of financial events. *Decision Support Systems*.

Challet, D. and Ayed, A. B. H. (2013). Predicting financial markets with google trends and not so random keywords. *arXiv preprint arXiv:1307.4643*.

Chawla, N. V., Japkowicz, N., and Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, 6(1):1–6.

Chen, E., Lu, Z., Xu, H., Cao, L., Zhang, Y., and Fan, J. (2020). A large scale speech sentiment corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6549–6555.

Corsi, F., Lillo, F., Pirino, D., and Trapin, L. (2018). Measuring the propagation of financial distress with granger-causality tail risk networks. *Journal of Financial Stability*.

Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

Crosbie, P. and Bohn, J. (2003). Modeling default risk.

Culver, W. J. (1966). On the existence and uniqueness of the real logarithm of a matrix. *Proceedings of the American Mathematical Society*.

Dekking, F., Kraaikamp, C., Lopuhaa, H., and Meester, L. (2005). *A Modern Introduction toProbability and Statistics*.

Diebold, F. X. and Yılmaz, K. (2014). On the network topology of variance decompositions: Measuring the connectedness of financial firms. *Journal of Econometrics*.

Duan, J.-C., Sun, J., and Wang, T. (2012). Multiperiod corporate default prediction — a forward intensity approach. *Journal of Econometrics*.

Duan, J.-C. and Wang, T. (2012). Measuring distance-to-default for financial and non-financial firms. *Global Credit Review*.

Duffie, D., Saita, L., and Wang, K. (2007). Multi-period corporate default prediction with stochastic covariates. *Journal of Econometrics*.

Erdemlioglu, D., Gillet, R. L., and Renault, T. (2017). Market reaction to news and investor attention in real time. *Available at SSRN 3010847*.

Etesami, J., Habibnia, A., and Kiyavash, N. (Unpublished results). Econometric modeling of systemic risk: A time series approach.

Etesami, J. and Kiyavash, N. (2014). Directed information graphs: A generalization of linear dynamical graphs. In *American Control Conference*. IEEE.

Fama, E. F. (1991). Efficient capital markets: Ii. *The journal of finance*, 46(5):1575–1617.

Fama, E. F., Fisher, L., Jensen, M. C., and Roll, R. (1969). The adjustment of stock prices to new information. *International Economic Review*, 10(1):1–21.

Ghoshal, S. and Roberts, S. (2016). Extracting predictive information from heterogeneous data streams using gaussian processes. *Algorithmic Finance*, 5(1-2):21–30.

Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*.

Granger, C. W. J. (1963). Economic processes involving feedback. *Information and control*.

Hong, Y., Liu, Y., and Wang, S. (2009). Granger causality in risk and detection of extreme risk spillover between financial markets. *Journal of Econometrics*.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*.

Huang, C.-L. and Wang, C.-J. (2006). A ga-based feature selection and parameters optimization for support vector machines. *Expert Systems with Applications*.

Iacopini, M. and Rossini, L. (Unpublished results). Bayesian nonparametric graphical models for time-varying parameters var.

Jiao, J., Permuter, H. H., Zhao, L., Kim, Y.-H., and Weissman, T. (2013). Universal estimation of directed information. *Information Theory*.

Kalli, M. and Griffin, J. E. (2018). Bayesian nonparametric vector autoregressive models. *Journal of econometrics*.

Kim, S., Putrino, D., Ghosh, S., and Brown, E. N. (2011). A granger causality measure for point process models of ensemble neural spiking activity. *PLoS computational biology*.

Koopman, B. O. (1931). Hamiltonian systems and transformation in hilbert space. *Proceedings of the national academy of sciences of the united states of america*.

Leippold, M., Maire, B., and Blochlinger, A. (2012). Are ratings the worst form of credit assessment apart from all the others? *Swiss Finance Institute Research Paper*.

Loftsgaarden, D. O., Quesenberry, C. P., et al. (1965). A nonparametric estimate of a multivariate density function. *The Annals of Mathematical Statistics*.

Longin, F. and Solnik, B. (2001). Extreme correlation of international equity markets. *The Journal of Finance*.

Lorenz, M. O. (1905). Methods of measuring the concentration of wealth. *Publications of the American Statistical Association*.

Loughran, T. and McDonald, B. (2011a). Barron's red flags: Do they actually work? *Journal of Behavioral Finance*.

Loughran, T. and McDonald, B. (2011b). When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*.

MacKinlay, A. C. (1997). Event studies in economics and finance. *Journal of economic literature*, 35(1):13–39.

Mann, H. B. and Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60.

Massey, J. (1990). Causality, feedback and directed information. In *Proc. Int. Symp. Inf. Theory Applic.(ISITA-90)*.

Mauroy, A. and Goncalves, J. (2019). Koopman-based lifting techniques for nonlinear systems identification. *IEEE Transactions on Automatic Control*.

Merton, R. C. (1974). On the pricing of corporate debt: the risk structure of interest rates. *Journal of Finance*.

Nikfarjam, A., Emadzadeh, E., and Muthaiyah, S. (2010). Text mining approaches for stock market prediction.

Nikolov, B. and Whited, T. M. (2014). Agency conflicts and cash: Estimates from a dynamic model. *The Journal of Finance*, 69(5):1883–1921.

Noshad, M., Zeng, Y., and Hero, A. O. (2019). Scalable mutual information estimation using dependence graphs. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Paninski, L. (2003). Estimation of entropy and mutual information. *Neural computation*.

Petrova, K. (2019). A quasi-bayesian local likelihood approach to time varying parameter var models. *Journal of Econometrics*.

Piramuthu, S. (2004). Evaluating feature selection methods for learning in data mining applications. *European Journal of Operational Research*.

Psaradakis, Z., Ravn, M. O., and Sola, M. (2005). Markov switching causality and the money–output relationship. *Journal of Applied Econometrics*.

Qasem, M., Thulasiram, R., and Thulasiram, P. (2015). Twitter sentiment classification using machine learning techniques for stock markets. *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*.

Quinn, C., Kiyavash, N., and Coleman, T. P. (2015). Directed information graphs. *Transactions on Information Theory*.

Quinn, C. J., Coleman, T. P., Kiyavash, N., and Hatsopoulos, N. G. (2011). Estimating the directed information to infer causal relationships in ensemble neural spike train recordings. *Journal of computational neuroscience*.

Quinn, C. J., Kiyavash, N., and Coleman, T. P. (2013). Efficient methods to compute optimal tree approximations of directed information graphs. *IEEE Transactions on Signal Processing*.

Quinn, C. J., Pinar, A., and Kiyavash, N. (2017). Bounded degree approximations of stochastic networks. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*.

Ranco, G., Aleksovski, D., Caldarelli, G., Grčar, M., and Mozetič, I. (2015). The effects of twitter sentiment on stock price returns. *PloS one*, 10(9):e0138441.

Renault, T. (2017). Intraday online investor sentiment and return patterns in the us stock market. *Journal of Banking & Finance*.

Renault, T. (2020). Sentiment analysis and machine learning in finance: a comparison of methods and models on one million messages. *Digital Finance*, 2(1):1–13.

Saif, H., Fernández, M., He, Y., and Alani, H. (2014). On stopwords, filtering and data sparsity for sentiment analysis of twitter.

Schönbucher, P. (2003). *Credit Derivatives Pricing Models: Models, Pricing and Implementation*. The Wiley Finance Series. Wiley.

Sheskin, D. J. (1998). *Handbook of Parametric and Nonparametric Statistical Procedures*.

Shirata, C. Y., Takeuchi, H., Ogino, S., and Watanabe, H. (2011). Extracting key phrases as predictors of corporate bankruptcy: Empirical analysis of annual reports by text mining. *Journal of emerging technologies in accounting*.

Sprenger, T. O., Tumasjan, A., Sandner, P. G., and Welpe, I. M. (2014). Tweets and trades: The information content of stock microblogs. *European Financial Management*.

Sricharan, K., Raich, R., and Hero, A. O. (2011). k-nearest neighbor estimation of entropies with confidence. In *Information Theory Proceedings*.

Tetlock, P. C., Saar-Tsechansky, M., and Macskassy, S. (2008). More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*.

Tsai, C.-F. (2009). Feature selection in bankruptcy prediction. *Knowledge-Based Systems*.

Tukey, J. W. (1977). *Explanatory Data Analysis*.

Wiener, N. (1956). The theory of prediction. *Modern mathematics for engineers*.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 347–354.

Yildirim, S., Jothimani, D., Kavaklioglu, C., and Basar, A. (2018). Classification of hot news for financial forecast using nlp techniques. *2018 IEEE International Conference on Big Data*.

Yuqinq, H., Kamaladdin, F., and Lipo, W. (2013). Feature selection for stock market analysis. *Lecture Notes in Computer Science*.

Divernois Marc-Aurèle

Rue du Midi 34, CH-1800 Vevey
27.02.1992, Swiss nationality
Email: divernois@gmail.com
Mobile: +41 79 756 47 53
Git: marcaureledivernois

Education

EPFL — Lausanne, CH
Ph.D., Advisor: Damir Filipovic — 2017–Current
- Area of research: "Machine Learning applied to Risk Management."

HEC Lausanne — Lausanne, CH
M.S. in Finance, GPA: 5.60/6.00 — 2013–2015
- Thesis: "Estimation of a forward intensity model for corporate default prediction".

HEC Lausanne — Lausanne, CH
B.S. in Economics, GPA: 5.30/6.00, ranked 6th in graduating class. — 2010–2013

Experience

Lombard Odier Asset Management — Geneva, CH
Risk Analyst — 2015–2017
- Responsible for the daily monitoring of performance and risk of >200 Multi-Asset and Fixed Income portfolios and mandates. Analyzed the risk metrics and exposures for key portfolios to track any material changes in the portfolio positioning.

Lombard Odier Asset Management — Geneva, CH
Analyst — 2015
- Creation of a VBA platform in order to automate all the reporting of the fund (CHF 7.5 billion AuM).

Northlight Group — London, UK
Intern — Summer 2014
- Analyst for the investment team of a credit hedge fund, participating in various tasks such as high yield loan modeling, equity research and firm valuation.

Technical skills

- **Data Science/Machine Learning with Python**
  - **Supervised & Unsupervised:** Regressions, PCA, Clustering, Numpy, Pandas, Scikit-Learn, Statsmodels.
  - **Web Data Scraping:** Requests, BeautifulSoup, Selenium, JSON.
  - **Natural Language Processing :** Sentiment Analysis, NLTK, SpaCy, TFIDF, Word embeddings.
  - **Deep Learning :** Artificial Neural Networks, CNN, RNN, TensorFlow, Keras, PyTorch.
- **Programming Languages**
  - Python, Matlab, R, SQL, VBA, Stata, HTML, CSS, Git, Office, LaTeX.

Publications

[1] M.-A. Divernois and D. Filipovic, "StockTwits Classified Sentiment and Stock Returns", 2022.

[2] M.-A. Divernois, J. Etesami, D. Filipovic, and N. Kiyavash, "Firm Networks Using Granger Causality", submitted to Journal of Econometrics, 2021.

[3] M.-A. Divernois, "A Deep Learning Approach to Estimate Forward Default Intensities", in *Swiss Finance Institute Research Paper No. 20-79*, 2019.

Conferences talks

- **Applied Machine Learning Days** 2022
  Speaker of the track of Advances of Machine Learning Approaches for Financial Decision Making & Time Series Analysis.
- **SIAM Conference on Financial Mathematics and Engineering** 2021
  Presented my paper "StockTwits Classified Sentiment and Stock Returns".
- **SFI Research Days** 2020
  Presented my paper "A Deep Learning Approach to Estimate Forward Default Intensities".
- **Swissquote Conference on Artificial Intelligence in Finance** 2019
  Speaker at the EPFL Finance and Technology Program.

Teaching

- **Head Teaching Assistant** at EPFL 2017-2022
  Financial Big Data (M.Sc. class, Prof. D. Challet) ∼50 students.
  Advanced Derivatives (M.Sc. class, Prof. E. Perazzi) ∼40 students.
- **Teaching Assistant** at HEC 2011-2014
  Economics I : microeconomics (B.Sc. class, Prof. T. von Ungern) ∼900 students.
  Economics II : macroeconomics (B.Sc. class, Prof. C. Sfreddo) ∼600 students.
  Principles of Finance (B.Sc. class, Prof M. Rockinger) ∼300 students.

Awards

- **Best Teaching Assistant 2021** 2021
  Elected Best Teaching Assistant by the students of the Master in Financial Engineering at EPFL. Prize awarded during the graduation ceremony.
- **Winner of Saxo Bank's portfolio management simulation** 2014
  Highest portfolio value out of ∼750 participants at the end of a six weeks stock market game held by SaxoBank.
- **Winner of HEC Business Game** 2013
  Team ranked first out of 64 participants on a real-time business strategy simulation held by several professors using ERPSim.

Languages

- **French:** Mother tongue.
- **English:** Fluent.
- **German:** Intermediate B1.

Extracurricular Activities

- **Chess club Teacher** 2018
  Giving chess lessons to children up to 1500ELO.
- **Head of Junior Entreprise** 2014
  Student club offering consulting services to the market.