* **Abstract**
* **Theoretical Background (literature review + network analysis theories + tools used)**

BACKGROUND

Network dynamics: – The network topology changes over times – Nodes and/or edges may come and go – Captures faults and reliability issue

These dynamic networks suffer an evolution ruled by the interactions between the individuals. One of the main assumptions when modelling these networks is that they tend to reach the equilibrium state, the states TESLA recovers.

A way to find an equilibrium state and measure how far the network is, so how unbalance the network is???

By duration networks can be – Transient: The dynamics occur for a short period, after which the system is static for an extended time period – Continuous: Changes are constantly occurring and the system has to constantly adapt to them

After all: It makes sense to model the financial market as a dynamic network: Network since its elements are linked and their behaviours can affect the whole system and dynamic to model the highly changing character of the stock market, that difficulties the ability to model it as a static system. In addition modelling a system as a evolving network gives the potential to study patterns or events’ effects that affect the network.

In many problems it is helpful and even necessary to study the relationships between entities of a system since we find them interconnected forming networks in which some small changes on an element can propagate and affect the entire system. To do so we use graph, and more precisely networks, theories which represent each individuals or actors by nodes and the connection or relationships as edges between the nodes.

We find a large list of examples in which it is necessary to apply network theory on different and wide subjects as social science, finance, biology or climatology.

Studying such networks can reveal lots of information like how an event can trigger a topological change of the entire network, how entities of the network can depend on each other’s states or share similar properties and organize themselves into groups or which individuals are the main agents that lead the evolution of the network.

Classically these analysis have been carried in the simplest way assuming static (non-varying) networks, however this is usually a simplification since in most of the cases the relationships between agents in a network evolve over time as the nodes’ states change. The internet, biological processes or the economy are some examples of networks evolving over time, shrinking or creating new links between the nodes.

Although there is wide literature and methods on modelling static networks it has not been until recent years that interest has grown on studying dynamic or time-varying networks ([1], [3], [5], [6], [7], [8], [11]).

Usually, only time series measurements, such as microarray, stock price, etc., of the activity of the nodal entities, but not their linkage status, are available. The goal is to recover the latent time-varying networks with temporal resolution up to every single time point based on time series measurements.

To be able to understand the current state of the research on the field and the strengths, weakness and challenges of the techniques proposed this work is going to analyze several algorithms, focusing mainly on TESLA (from the acronym TESLLOR, which stands for temporally smoothed *l*1*-*regularized logistic regression), built to recover the structure of time-varying networks over a fixed set of nodes from their time series nodal attributes [1].

# network main properties

Depending on the problem under consideration, it may be convenient to study the network from different points of view, focusing on relationships between the entities (edge-centric), one or more entities (vertex-centric) or the topological properties of the network as a whole (graph-centric) [5].

When studying edge-centric networks some important measures include transitivity (similar to the mathematical property: if *u* connected to *v* and *v* connected to *w*, then *u* connected to *w*) and reciprocity (when we find loops in a network with 2 edges length, so pairs of nodes that have edges running in both directions).

Meanwhile in vertex-centric networks centrality measures are used to quantify the relevance of the different nodes in a network. The most basic example is the degree of centrality of the nodes: the number of edges attached to them. Nodes with high degree are called hubs and obviously play an important role in the system.

Regarding graph-centric approach we are able to analyze the general shape by studying the clusters in which the vertices organize themselves, many techniques like the well-known hierarchical clustering are useful in this task showing important properties of the network like its tendency to follow an assortative or dissortative mixing pattern.

All these properties [9] give us important information to understand the behavior of the system.

# dynamic networks

Latest years the science of networks developed three important aspects that define this field nowadays: it is used to model real-world problems, it frequently assumes the networks are not static and it aims to understand networks not just as topological objects but also as the framework upon distributed dynamical systems are built [3].

Then studying networks as dynamical systems is the best way to understand evolving underlying processes in complex systems that can trigger critic events reshaping the network and changing its properties, one example of this is how a disease spreads: the structure of a network through which a contagious agent is transmitted can have a dramatic impact on outcomes at the level of entire populations [3].

# learning networks

Looking at their topology we find that networks are organized in two main classes: scale-free networks and random graphs.

Scale-free network term was coined to describe the networks that show a power law degree distribution (distribution of the connections over the whole graph) after Barabasi mapped a portion of the World Wide Web [2] finding that it followed this distribution of the links and that some nodes had many more links than others (hubs). Then the most notable characteristic in scale-free networks is that it is relatively common to find vertices with a much higher degree (links) than the average. They also have other important features like robustness due to its hierarchical organization or clustering coefficient that may help to understand networks following this distribution.

A random graph is obtained by starting with a set of isolated vertices and adding successive edges between them at random. The aim of the study in this field is to determine at what stage a particular property of the graph is likely to arise. Different random graph models produce different [probability distributions](https://en.wikipedia.org/wiki/Probability_distribution) on graphs [4].

## Exponential Random Graph Model

Based on this last one, the exponential random graph model (ERGM) has been extensively used to model static networks and also used as a base for two of the first algorithms proposed to study dynamic networks, the temporal exponential random graph model (tERGM) and the hidden tERGM for modelling a sequence of node attribute [7]. When tERGM assume that the sequence of networks is available, htERGM explores the possible dependencies of unobserved rewiring networks and leads to the algorithm that can reconstruct such networks from a snapshots’ sequence of nodal attributes.

Although this algorithms overcame approaches that recover single time-invariant networks they are not the most useful ones since they depend on unobserved network variables being unable to compute likelihood ratios, needing inference algorithms specifically for each problem.

## Hidden Markov Dynamic Bayesian Network

Another classical approach are the Bayesian networks, widely used to describe biological systems, risk analysis and even financial networks.

Bayesian networks are graphs whose edges represent conditional dependencies and nodes are random variables in a Bayesian sense (observable quantities, latent variables, unknown parameters or hypotheses) and are associated with probability functions that takes as input the node’s parent variables and gives the probability of the variable represented by the node.

Its standard assumption is ‘stationarity’, and therefore, several research efforts have been recently proposed to relax this restriction. However, those methods suffer from three challenges: long running time, low accuracy and reliance on parameter settings. [11] propose a non-stationary DBN model by extending each hidden node of Hidden Markov Model into a DBN (called HMDBN), which properly handles the underlying time-evolving networks resulting in a promising experimental evaluation of the method, demonstrating more stably high prediction accuracy and significantly improved computation efficiency (even with no prior knowledge and parameter settings) on both synthetic and real biological data.

Although this probabilistic model is more complex than TESLA algorithm it could be a good method to implement in order to compare results and performance.

## Bayesian non-parametric model

In this line there was proposed a Bayesian non-parametric model including time-varying predictors in dynamic network inference precisely for financial studies [6].

This model computes edge specific predictors where the link probabilities () are estimated via a logistic regression, with a baseline process quantifying the overall propensity to form links in the network across time, are vectors containing the latent coordinates favoring a higher link probability when units *i* and *j* have latent coordinates in the same direction and is a P-dimensional vector of time-varying edge-specific predictors for units *i* and *j* at time t and are the corresponding dynamic coefficients. This allows the proximity between units *i* and *j* at time *t* to depend on predictors in a manner that varies smoothly with time.

The main issue regarding this method is that it assumes time-constant smoothness while in finance and other network applications we expect smoothness to vary over time, however it had some promising results when analyzing the global stock market network in the 2008 crisis and this makes it a good candidate to test in this project.

## Temporally smoothed l1-regularized logistic regression

TESLA represents an extension of the lasso-style sparse structure recovery technique and is based on a key assumption that temporally adjacent networks are likely not to be dramatically different from each other in topology and therefore are more likely to share common edges than temporally distant networks.

Building on the *l*1-regularized logistic regression algorithm for estimating single sparse networks [10] it was developed a regression regularization scheme that connects multiple time-specific network inference functions via a first-order edge smoothness function that encourages edge retention between time-adjacent networks. An important property of this idea is that it fully integrates all available samples of the entire time series in a single inference procedure, what means an advantage in contrast of htERGM algorithm.

TESLA estimates , the correlation (or dependency strength) matrix between the nodes using a time-series of the observed stated of the nodes **xt**.

where,

Where the graph structure is given by the locations of the nonzero elements of the parameter vectors . Components of the vectors are indexed by distinct pairs of nodes and a component *j* of the vector is nonzero if and only if the corresponding edge (*i, j*) ∈ Eτ. denotes the observed states of all nodes but node *i* and is the log conditional likelihood of state under a logistic regression model.

First term of the algorithm are p logistic regressions, one for each node with respect the rest of them ( are the states of all nodes but node I in the dth sample in time epoch t).

The second term is a lasso regularization that introduces sparsity and consistency in neighborhood selection, as showed in [10], converging to the true graph structure. Also a sparse graph effectively limits the degree of freedom of the model, which makes structure recovery possible given a small sample size [8].

Since in this problem we are estimating dynamic networks (so more than one single graph), and, as assumed before, temporary adjacent networks are going to be similar one to each other, we need to introduce the third term in the equation that penalizes the discrepancy between time-adjacent parameters .

Summarizing, the first term is the main one while the last two terms are the penalties () that are introduced to enforce sparsity and smoothness.

## TESLA updated

It is important to note that in this specific method we are assuming that are a piecewise constant function with abrupt changes in parameters (jumps).

However in some cases the changes in parameters are continuous, so further work [8] was carried out to adapt this algorithm including a weighting term that turns in continuous functions instead of the penalty in the structural changes () that assumes small “jumps” between time-adjacent values.

Where the weights are defined by

And is a symmetric nonnegative kernel function.

# financial application

When studying dynamic networks we aim to solve real world problems, so it is usual to look at meta-networks, this is multi-link (many types of links), multi-mode (many types of nodes) and multi-level (nodes can represent subnetworks) networks.

This added to the fact that not many work has been done to recover dynamic networks on finance and the promising algorithms reviewed on this paper sum up enough motivations to build an interesting topic: testing the performance of these different methods in the financial area to find the model that best defines it and extending the previous work done in this subject (analyze the behavior of financial networks against critic events [6]) by looking at meta-networks instead of simple networks.

ANALYSIS

Intro

When analysing a network we have to focus on different levels, going from the overall network as whole (in order to understand its distribution and measure the cohesion of the network), and later focusing on the individuals (to identify the central nodes of the system that support the network and study movements and interactions)

**Net topology**

To study the topology of the network first we need to find a suitable layout to build the network.

Many options are available, after testing and studying the designed layouts existing in R there were chosen some of them in order to try different approaches:

- Frutcherman-Reingold layout: based on a directed force algorithm which takes the nodes as charged particles that tend to repel while the edges are treated as springs with an [attraction force or constant???] equals to their weights (TESLA output). The algorithm iterates until it converges to a minimum energy state (equilibrium). Force directed algorithms seem an ideal choice in this case since heavily connected individuals will tend to group while keeping unrelated nodes separated.

- Kawai-Kamada layout: It is also based on a directed force algorithm. Edges:springs???

- other force directed algorithms????

- Multidimensional scaling layout: This performs the classic MDS algorithm that uses the chosen distance metric to compute nodes similarities and then plot them in a 2D space keeping these distances and showing the edges.

This way one can directly identify similar stocks and compare with the results given by the force directed algorithms. In addition it has the potential to be a useful layout to study flows and contagion effects.

- Random/Circular/By Sectors… FIXED OVER TIME: (define the used algorithm)

Keeping the nodes fixed over time is the best way to focus on the evolution of the relationships in the system. Also this one seems to be a good candite to study flows and compare with MDS.

- clustered by sectors (grid and starting coord for F-D

- using MDS as starting coord for force directed??

Nodes:charged particles?????

<https://en.wikipedia.org/wiki/Graph_drawing>

https://en.wikipedia.org/wiki/Preferential\_attachment

* Net Robustness (percolation) [https://en.wikipedia.org/wiki/Robustness\_of\_complex\_networks https://en.wikipedia.org/wiki/Percolation\_theory]
* Net connectivity https://en.wikipedia.org/wiki/Connectivity\_(graph\_theory)
* Diameter, net efficiency
* Distance + fastest paths
* Min spanning tree [https://en.wikipedia.org/wiki/Minimum\_spanning\_tree]
* Centrality (general net) https://en.wikipedia.org/wiki/Centrality#cite\_note-NewmanNetworks-1
  + Degree
  + Eig. Cent https://www.math.washington.edu/~morrow/336\_11/papers/leo.pdf
  + Betweeness
  + Closeness
* Crash and contagion study (gain/loss)

**Analysis Sectors**

* (EVOL OVER TIME OF:) Centrality, Degree, Weights, inter and intra connectivity

**Analysis Stocks IMPORTANCE**

* Centrality, Degree, Weights

- Software used:

Matlab: Used to compute TESLA algorithm, which is built and ready to retrieve online. This script has been modified according to our needs, simplifying it, what makes it easier a further implementation of the whole ‘preprocessing + computation + analysis’ pipeline in R to generate a function that automatically recovers dynamic networks and analyses them (TESLA algorithm just finds the dependency strengths).

R: Used in the exploration/preprocessing stage and graph plotting and analysis thanks to the packages *igraph, network, sna* and *ggplot2*.

* **Experimental results**

- Data (characteristics, exploration and preprocessing). Different kinds of data and many cases in finance!! -> different approaches (changing parameters, analysing TESLA/correlations

As said before, TESLA estimates the dependency strength between nodes using time-series of the observed states of the nodes, hence the data used in this work correspond to time series of the stock values for the most important companies forming the British index: FTSE100.

In order to study and analyse critical events we want to have some reference cases. To do so the data used in this work comprises different epochs.

* 2008 crisis (2006 – 2010)
* Brexit (2015 - ?)
* Different intervals: references? (2009-2010)

…

The FTSE100 consists of the largest 100 UK companies by full market value, given that the problem here is that the companies forming the FTSE100 change quarterly resulting in not homogenous data for large enough intervals, to solve this potential problem, data from all the stocks taking part on the FTSE100 in any time of the interval studied are retrieved (when data available during the whole interval). In a similar way, we noticed a couple of cases in which not all stocks had valuation probably corresponding to a vacation, this is solved by performing inner join (using only observations with values for all stocks).

Before moving on, it is important to notice that a large slice of these stocks are international companies, however, so the index's movements are a fairly weak indicator of the UK economy

A better indication of the UK economy is the FTSE250 since it contains bigger proportion of national companies. (study??) or FTSE350?

Since we assume that stocks with similar trend are likely to be influenced by the same underlying event, translating in the existence of a link between them, it is needed to calculate the daily returns. In addition to this, just using the stock values it is very unlikely to find links between companies with different order of values even if their trends are similar.

Once data is prepared we input it to TESLA.

First attempts it didn’t work at all, after simplifying some complex and useless features (for this problem) it started working.

After exploring and plotting first the results, I found out that the parameters for the indices i,i (node relation with itself) gave extremely high values (in this very case we are not interested in loops on the same node and it obscures the possible existing interactions with other stocks). Given the problem formulation on the TESLA paper [ref] we expect these values to be 0. In order to fulfil this condition some further modification of the algorithm was necessary.

- (updated)Algorithm + Analysis: Different analysis carried on after some exploration of the data and networks (few high degree nodes: highlight them, study them with analysis variables and last compare with news/historical data about those companies or general behaviour of market in those years (time series of stocks!)

The analysis performed to these networks focus on studying how the topology of the system changes on different situations, finding and studying the effects of the most important agents and understanding the relations between these stocks.

General topology

The first networks computed showed a star distribution in which there is one or few central nodes (hubs) that connects to (almost) all rest of the ‘weak’ nodes, usually only connected to the hubs, setting the background net on which the whole system is built. An advantage of the star topology is the simplicity of adding additional nodes as well as its reliability since because if one node or its connection breaks it doesn’t affect the other nodes and their connections. On the other hand the hub represents a single point of failure, the less hubs the system have the less robust the system is.

First thing we can notice after having a look to the financial networks is that it has similar characteristics than a scale-free distribution but doesn’t follow a power law, instead there is usually a small number of high-degree nodes (hubs) while the rest follow quasi uniform values.

Robustness???

Pathways??

Weak or strong nets when facing contagion?? (notes)

GainLoss matrix: contagion??? :::::: use different shape for financials!!!!!! (try to study if they represent riskier nodes):::::::: study this mathematically (so find the proportion of adjacent nodes that gets the same state [+/-] that the source node had the previous epoch taking into account the sign of the interaction (so if the interaction is neg: the nodes would need to have different sign), and measure the average abs(weight) of the nodes that joined them (esto es porque las interacciones negativas ya las hemos tenido en cuenta) so we can check the viability of being the contagion source)-> maybe more useful in high freq analysis)

Also: sum incoming weights of node i (taking into account the state (sign) of the sources): sum(weightsIn(j)\*GainLoss(j)) (normalize??) j represent every node but the one we are studying. After this we can compare with the Gain/Loss value of the node i (in the same period? The next period??) and try to find a correlation: or a model. If we can model and predict accurately enough (error, boxplots…) the gain or loss of a stock given the incoming weights of its connected nodes this could be extremely helpful for companies to prevent situations and react before to minimise the loss or maximise the gains. \*\*\*try different approaches: maybe with the averaged weight, or taking into account other factors as the previous Gain/Loss value of the node I, or its value on the stock market, or weighting the weights by the companies’ value (CP)??(4M, 2.1B, 856M…) of the source nodes: sum(weightsIn(j)\*GainLoss(j)\*CP(j))/sum(CP(j))

And maybe multiplying or using CP(i).

CENTRALITY MEASURES

Information about the relative importance of nodes and edges in a graph can be obtained through [centrality](https://en.wikipedia.org/wiki/Centrality) measures

"Importance" can be conceived in relation to a type of flow or transfer across the network. This allows centralities to be classified by the type of flow they consider important[ Borgatti, Stephen P. (2005). "Centrality and Network Flow".Social Networks. Elsevier.**27**: 55–71.[*doi*](https://en.wikipedia.org/wiki/Digital_object_identifier):[*10.1016/j.socnet.2004.11.008*](https://dx.doi.org/10.1016%2Fj.socnet.2004.11.008).] "Importance" can alternately be conceived as involvement in the cohesiveness of the network.

Degrees: total, in out????

One important characteristic of the recovered networks is that they are directed given that the adjacency matrices computed by the TESLA algorithm returns the relation of each one of the nodes (independently) with the rest on a single inference procedure. This results on not (necessary) reciprocal interactions. In this case it is necessary then to study out (connections of the computed node with the rest) and in (connections of the other computed nodes relating this one) degrees of the nodes.

Degree out: Number of computed connections that emerge from the studied node when applying TESLA to find its relation with the rest.

Having a look to Out Degree tables we can observe that they are quite homogenous in general (excluding some outliers we will study further as possible hubs), this is given by TESLA since the time-series used for each iteration (stocks) are almost the same (every node but the one we are studying) and they share constraints. Although the Out Degree is usually similar in all nodes in a given network, when comparing these values in different time stamps we find that they vary [buscar context cuando desconectada], pointing the potential of this variable to measure market’s global cohesion.

[[First of all we would compare this value with the degree in, is it a hub.in?, then the time series of the stock….]]

Degree in: Number of computed connections that reach a given stock.

On these histograms we can see a completely opposed distribution of the Out Degrees. Since this degree gives us the number of times a given stock appears as related to the rest when applying TESLA to them we can just use it as a relevance/centrality measure. The higher the degree, the most relevant is the stock for the market so it will have more effect on it (leading the system, spreading a contagion, connecting the individuals and setting a background net over which the market structures…) finish

(degree variation over time: sectors and general) In the example case we can observe how the degree of all sectors decrease at the same time indicating a reaction to ?????

Also here we can check how although the financial sector has bigger degree, the stocks belonging to the utilities sector represent stronger hubs as they have more average number of edges attached. Probably the Financial sector is more important as a whole (maybe highly intraconnected approaching the behaviour a supernode itself?) but the average stock in the Utilities sector is stronger than the ones in Financials. This could be interesting: degree of a sector vs avg degree of stocks from sector

Eigenvector centrality

Betweenness centrality

Katz centrality???

Weights

Values and distributions by individuals and sectors

Intra and inter sector connectivity

In order to study importance of different sectors over time we can compute sector degrees and weights.

Also to know how much a sector is isolated or interacts with the market we should study its intra and interconnectivity with other sectors by measuring the proportion of connections a given sector have between its stocks and with different sectors.

The bigger the intraconnectivity compared to the interconnectivity the more isolated a given sector will be while the bigger the interconnectivity the more active it is.

Layouts (physic models, MDS, random fixed…)

Different information from different approaches.

Main ones: kk layout (particles and springs) to study how the general system evolves and random/circular fixed to focus on how the links changes over time and study flows or contagions.

MDS gives us the similarities between the stocks (using Euclidean distance and computed for each of the epochs to compare with the network)

Explain layouts

TODO: Network grouped by sector. Kk-means. Network coordinates = pca???

Show plots and explain encodings: size = degOut+degIn/10 (so we can see the hubs double than the rest but still can see them), vertex colors=sectors, edge colors: edge in, clockwise flow….

Basically enumerate all analysis carried on and why (based on network analysis, on other papers or made up after some exploration) and how they relate to each other, which results we can extract from all data analysed together…)

* **Conclusions**
  + Comparison with other works
  + Further work (next month, next 20 years)
  + Conclusions