UNIVERSIDADE D
COIMBRA

Rui Pedro Vilar Portela Seabra

Causal Inference of Networked Dynamical Systems under Partial Observability and Structured Noise: A Feature Based Approach

Dissertation in the context of the Master in Informatics Engineering, specialization in Intelligent Systems, advised by Dr. Augusto Santos co-advised Dr. Jorge Henriques, and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September of 2023

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

DEPARTMENT OF INFORMATICS ENGINEERING

Rui Pedro Vilar Portela Seabra

Causal Inference of Networked Dynamical Systems under Partial Observability and Structured Noise: A Feature Based Approach

Dissertation in the context of the Master in Informatics Engineering, specialization in Intelligent Systems, advised by Dr. Augusto Santos and co-advised by Dr. Jorge Henriques, and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September 2023

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Rui Pedro Vilar Portela Seabra

Inferência Causal de Sistemas Dinâmicos em Rede sob observabilidade parcial e ruído estruturado: Abordagem baseada em Features

Dissertação no âmbito do Mestrado em Engenharia Informática, especialização em Sistemas Inteligentes, orientada pelo Dr. Augusto Santos e co-orientada pelo Dr. Jorge Henriques, apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Setembro 2023

Acknowledgements

I would like to express my deepest gratitude to all those who took part in this research either in a more active or less present way.

To Dr. Augusto Santos firstly for the availability since the beginning of this project, always ready to help with questions and always open to discuss new ideas. The ease provided by him from the start was a big help to get through this project and the approach chosen. Because of that the group went along at all times and understood each other without any sort of stress during the progress made. The knowledge that he carries is so vast and to be able to transmit it to other person it is not easy and it requires discipline and desire which Augusto never failed to show. To Prof. José Moura for giving a scientific input and giving us feedback from the work done including the papers. For the opportunity to go to an internship in CMU - Carnegie Mellon University that will strengthen our knowledge and experiment usefull for our future success. To Pr. Jorge Henriques for the availability to be present in some meetings and giving his opinion on the results and ideas through out this thesis.

This endeavor would not have been possible without my parents and sister extraordinary support throughout my course, during the good and bad moments. Thank you for all the effort so that I didn't lack anything during these years. That effort was certainly not in vain.

With regard to personal and professional terms, the present research work was not only crucial to enhance my technical skills regarding the subjects of Deep Learning and Graph Learning, but also to strengthen my soft skills. This academic internship allowed me to put to the test my commitment and responsibility, to improve my communication ability, and most importantly, the development of critical thinking.

This thesis was partially supported by FCT/MCTES – Foundation for Sci- ence and Technology, Portugal under the project UIDB/50008/2020.

Abstract

Complex systems evolve over time driven by the interactions among its units or nodes. Examples are Brain activity, pandemics, social networks, Gene Regulatory Networks. These are applications whereby the underlying connectivity pattern between its comprising units fundamentally characterize the long term faith of the system or explains distinct emergent patterns, e.g., epileptic seizures, long term behavior of a pandemics, or aid in the design of mitigation policies in a pandemics. However, in all these applications, the causal geometry is not transparently available and should be inferred from observed data (time series) with technical guarantees of structural consistency. This Thesis studies the problem of identifying the causal structure of linear Networked Dynamical Systems. Owing to the intrinsic large scale nature of complex systems, we can only probe the time series activity at a subset of nodes. Further, in general, these Networked Dynamical Systems are excited by (possibly adversarial) noise or control input that exhibit nontrivial statistical structure. We offer two main contributions within the challenging scope of causal inference under the presence of latent nodes and structured excitation noise: i) A novel condition over the noise structure wherein the directed network can be consistently inferred from observed data (Chapter 4); ii) A novel causal inference algorithm with competitive performance (Chapter 4). In Chapter 5, we present a comprehensive collection of numerical results benchmarking our approach against popular state-of-the-art methods like Granger or Precision matrix (or Graphical Lasso) over directed networks, i.e., networks where a node i can influence node j, but not the other way around. The numerical experiments are performed across distinct regimes of connectivity, observability and noise correlation. The work developed has been submitted for publication [Santos et al., 2023].

Keywords

Causal inference, directed graphs, complex systems, statistical analysis, machine learning, Granger estimator, colored noise, networked dynamical systems, brain structural connectivity, real-data application.

Resumo

Sistemas complexos evoluem ao longo do tempo impulsionados pelas interações entre os seus elementos ou nós. Exemplos incluem a atividade cerebral, pandemias, redes sociais e redes de regulação genética. Estas são aplicações em que o padrão de conectividade subjacente entre os elementos fundamentalmente caracteriza o destino de longo prazo do sistema ou explica padrões emergentes distintos, como convulsões epilépticas, comportamento de longo prazo de pandemias ou auxilia no desenvolvimento de políticas de mitigação em uma pandemia. No entanto, em todas essas aplicações, a geometria causal não está transparentemente disponível e deve ser inferida a partir de dados observados (séries temporais) com garantias técnicas de consistência estrutural. Esta tese estuda o problema de identificar a estrutura causal de Sistemas Dinâmicos em Rede lineares. Devido à natureza intrinseca de grande escala dos sistemas complexos, só podemos sondar a atividade de séries temporais em um subconjunto de nós. Além disso, em geral, esses Sistemas Dinâmicos em Rede são excitados por ruído (possivelmente adversarial) ou um *input* controlado que exibe uma estrutura estatística não trivial. Oferecemos duas contribuições principais no tema desafiador da inferência causal na presença de nós latentes e ruído de excitação estruturado: i) Uma nova condição sobre a estrutura de ruído na qual a rede pode ser consistentemente inferida a partir dos dados observados (Capítulo 4); ii) Um novo algoritmo de inferência causal com desempenho competitivo (Capítulo 4). No Capítulo 5, apresentamos uma coleção abrangente de resultados numéricos comparando a nossa abordagem com métodos populares de última geração, como Granger ou matriz de precisão (ou Grafo de Lasso). Os experimentos numéricos são realizados em regimes distintos de conectividade, observabilidade e correlação de ruído. O trabalho desenvolvido foi submetido para publicação [Santos et al., 2023].

Palavras-Chave

Inferência Causal, sistemas complexos, análise estatística, *machine learning*, *Granger* estimator, ruído colorido, sistems dinâmicos em rede, conectividade estrutural cerebral, aplicação em dados reais.

Contents

Acronyms

CNN Convolutional Neural Network.

FC Functional connectivity.

FFNN Feed-Forward Neural Network.

IdGap Identifiability Gap.

List of Figures

Chapter 1

Introduction

This Chapter presents a brief motivation for the main subject of this Thesis in Section 1.1 and the discussion of our main goals and contributions in Section 1.2. Lastly, we introduce the outline structure of this document in Section 1.3.

The work developed was submitted for publication [Santos et al., 2023].

1.1 Motivation

This thesis focuses on the application of machine learning techniques for the purpose of identifying the causal relationships underlying Networked Dynamical Systems, i.e., the directed network linking the distinct units of the system. Networked Dynamical Systems (NDS) can be defined as interacting systems, agents or nodes, each undergoing state changes over time as the overall system progresses. The collective dynamics are characterized by the coupling among the nodes and therefore, the pattern of connections linking these nodes holds paramount significance in comprehending the temporal evolution of Networked Dynamical Systems. Within the realm of Social Networks, the diffusion of *fake* news or information and beliefs critically relies on the underlying (social) network structure [A. Lalitha and Sarwate, 2018; Jadbabaie et al., 2012; Matta et al., 2020b]. Deeper comprehension of emergent patterns of the human brain remains still quite elusive; however, recent research has advocated that insights into its underlying connectivity pattern yield pivotal information concerning cognitive disorders [Huang and Ding, 2016; Liégeois et al., 2020; Morone et al., 2017; Stam et al., 2007; Wang et al., 2014]. In instances of pandemics, the comprehension of the connectivity among distinct communities conforms to a pivotal blueprint to formulating decisions regarding preventative measures [Ganesh et al., 2005; Ren et al., 2019], such as the protocols and quarantines instated during the global outbreak of COVID-19. Across all aforementioned examples, the time series entailing the state evolution of the nodes are observable, for example, through Functional Magnetic Resonance Imaging (FMRI) or Electroencephalography (EEG) in the brain activity case. Nevertheless, the underlying network that interconnects these nodes remains concealed. The configuration of this connectivity structure

profoundly influences the formulation of mitigation policies in scenarios involving pandemics, and it significantly informs the diagnose and therapeutic aspects of cerebral afflictions based on neural activity. This thesis endeavors to formally study the inference of this latent network from the time series data stemming from a linear Networked Dynamical System (NDS) under partial observability – the time series data of only a subset of the nodes can be feasibly probed, – specifically targeting directed networks (causal inference). As in general applications, the NDS is excited by *colored* noise, i.e., the noise is correlated across nodes. While the bulk of the causal inference literature assumes whiteness or noise independence (for analysis purposes), this conforms to a strong assumption, in general, and we consider the challenging noise correlation setting as this is a common characterizing property across real scenarios.

In the causal inference literature, the main approach is to estimate a scalar value that describes the interaction strength between two nodes from the time series data. This is done through various methods, e.g., the standard correlation, Precision matrix [Loh and Wainwright, 2013], or mutual-information [Chow and Liu, 1968]. Fig. 1.1 illustrates the main idea. If the value estimated is *high* enough, the nodes are considered connected, otherwise, it is considered disconnected. The method can be described as a thresholding or hypothesis testing problem where above the threshold are the linked pair of nodes, and under it are the disconnected ones.

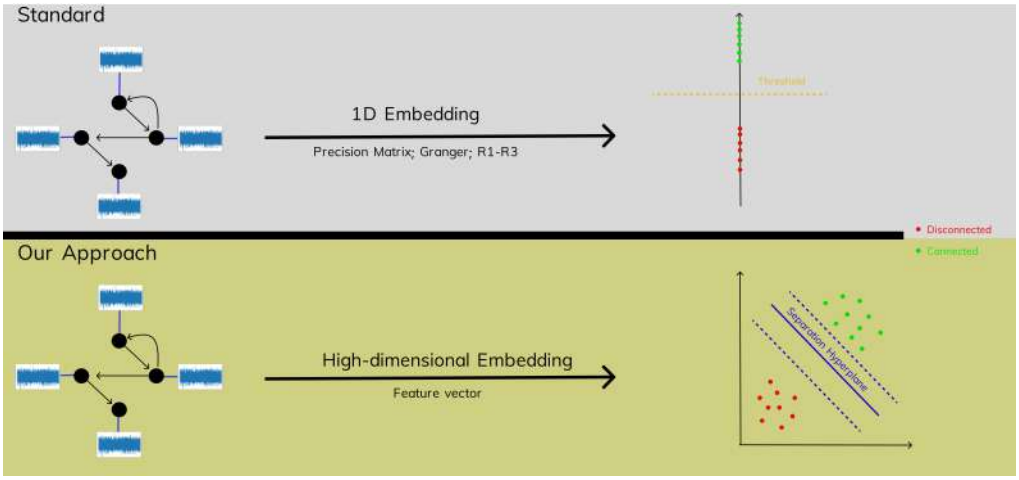


Figure 1.1: Our approach with the feature vector embedding vs the main scalar based methods in the literature. Our approach builds on the previous work [Machado et al., 2023] and relies on certain identifiability properties (namely, linear separability) in comparison with the literature main approach.

In reference [Machado et al., 2023], the pertinent information concerning the network connectivity is encapsulated in the separability of the set of features: A hyperplane partitions the set of features, i.e., features of connected pairs lie on one side of the hyperplane and features of disconnected pairs lie on the other side. Notably, the features's inherent separability properties have been rigorously established, engendering the prospect of harnessing machine learning methodologies, namely, supervised methods for the inference of the network's underlying

structure. It is worth noting, however, that the scope of the investigation carried out in [Machado et al., 2023] is restricted to Networked Dynamical Systems excited by diagonal noise, i.e., the covariance matrix of the noise is a multiple of the identity matrix (noise is independent across nodes). We greatly extend the approach in [Machado et al., 2023] to the case of colored noise under partial observability. Namely, the present thesis explores a more comprehensive setting, encompassing systems characterized by colored noise—wherein the noise covariance matrix contains off-diagonal elements distinct from zero. This extends the analytical framework to account for a broader spectrum of real-world scenarios, thereby enhancing the model's applicability and fostering deeper insights into the inherent structure of Networked Dynamical Systems. Further, instead of Convulational Neural Networks (CNNs) our method consists on Feed Forward Neural Networks (FFNNs) with the input of novel features. All-in-all our main goal is to develop tools to consistently recover the connectivity pattern underlying a Networked Dynamical System (NDS) from partially observed time series under colored noise excitation.

1.2 Proposed Approach

This thesis focus on identifying the network structure underlying linear NDS excited by colored noise and under partial observability. The qualitative behavior of these dynamical systems strongly depends on the underlying causal network linking its constituent units. In this thesis, we focus on directed graphs to establish a novel feasibility condition on the noise structure whereby the network can be consistently inferred from the observed time series. The networked system is generated as a directed graph where only few nodes are observable. From the time-series stemming from those nodes, we compute the new feature vector and through our reconstruction module the structure is recovered. These time series datasets stem from a linear NDS excited by colored noise. Fig. 1.2 summarizes the paradigm.

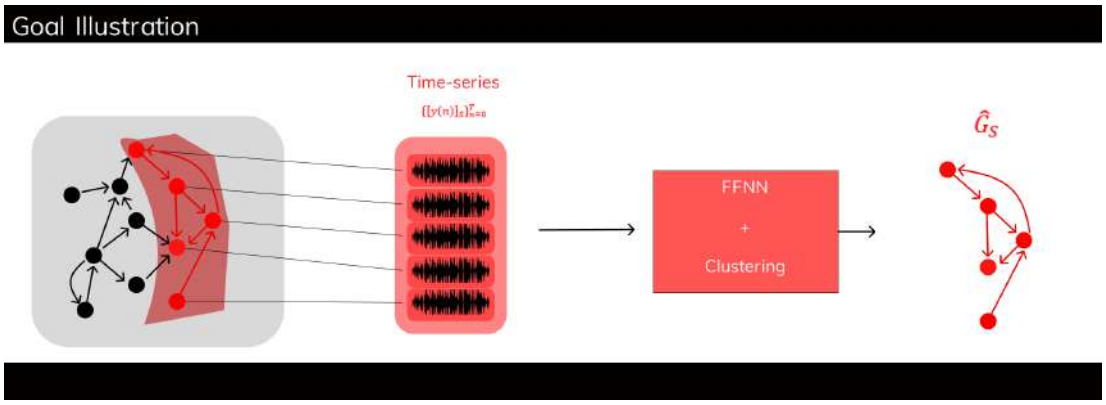


Figure 1.2: Causal inference under partial observability.

In reference [Machado et al., 2023], the approach primarily centered around the analysis of linear NDS time series perturbed by diagonal noise. However, in

our current scenario, the time series under consideration are characterized by the presence of colored noise. Consequently, the efficacy of their proposed model in our context has been considerably diminished. This is attributed to the fact that the [Machado et al., 2023] model was exclusively tailored to scenarios involving diagonal noise patterns and did not encompass the complexities associated with colored noise environments. Therefore, a novel set of feature vectors is computed from the time series characterized by colored noise to describe the connectivity between nodes. The separability properties of these features are proved, i.e., they are linearly separable, in that, there is a hyperplane (in feature space) that consistently stratifies the features stemming from connected pairs and those from disconnected pairs. Finally, for the training set up, our proposed method has as its input these new feature vector, and with a FFNN plus a clustering method applied to its output, we can consistently recover the underlying structure under partial observability, outperforming other popular methods as discussed in Chapter 5.

1.3 Outline

The structure of this thesis and a brief description of each Chapter content is provided below:

- **Chapter 2** – we set the problem formulation and introduce the main background tools in order to offer a solid ground over all the points developed along this thesis. More concretely, we present the basics of NDS, brain neuroscience and the modules of Artificial Neural Networks (ANNs) used throughout the thesis.
- **Chapter 3** – this Chapter entails a brief description of some of the related estimators in the causal inference literature.
- **Chapter 4** – provides the technical results and formulations of the proposed approach. Namely, we formally establish a novel feasibility condition on the noise statistical structure, i.e., a sufficient condition on the noise covariance that renders causal inference (under partial observability) a *well-posed* problem. That is, there exists an algorithm to consistently infer the direct network structure.
- **Chapter 5** – benchmarks our feature based approach with other popular estimators, including [Machado et al., 2023], via numerical experiments across distinct regimes of noise correlation, observability and connectivity.
- **Chapter 6** – presents the concluding remarks and proposes various promising paths for future research directions.

Chapter 2

Background

This Chapter entails the background concerning the basics of Networked Dynamical Systems, causal inference of NDS, Artificial Neural Networks and the main problem formulation of the thesis.

2.1 Networked Dynamical System

In recent years, the integration of network theory and dynamical systems has led to the emergence of a captivating framework known as networked dynamical systems. This paradigm extends the classical dynamical systems theory to account for the intricate interactions between interconnected components, making it particularly pertinent to the realm of causal inference in the scope of complex systems. Networked dynamical systems are comprised by entities or units within a system that are seldom isolated but rather interact with and influence each other. These interactions can be conceptualized as connections in a network, where nodes represent system components, and edges capture the influence or coupling between them. The dynamical behavior of each component is then contingent upon not only its internal dynamics but also the dynamics of its connected neighbors.

2.1.1 Causal Inference of Networked Dynamical Systems

The integration of network theory and dynamical systems offers a powerful toolkit for addressing the challenging problem of causal inference. Traditional causal inference methods often rely on observational data and assume static relationships. In contrast, networked dynamical systems provide a means to capture the evolving cause-and-effect relationships within a system. By studying how changes in one component's state influence the states of connected components over time, networked dynamical systems enable the identification of causal relationships in scenarios where traditional methods might fall short. The temporal evolution of the system's states can reveal causal dependencies that arise due to feedback loops, delayed responses, and hidden interactions.

The crux of networked dynamical systems indexed as directed graphs lies in their ability to causal relationships within complex systems. By encoding causal influences as directed edges, these graphs visualize not only the existence of connections but also the direction of influence. This inherent causal directionality is particularly valuable for discerning the cause-and-effect relationships that govern the system's behavior. Observing the temporal evolution of states within a directed graph-embedded dynamical system can unveil the cascade of effects triggered by initial changes. Through this lens, causal relationships emerge as paths along directed edges, revealing the sequential manner in which changes propagate through the system. The directed graph representation of networked dynamical systems finds profound applications in the field of causal inference. Directed graphs offer a holistic way to visualize and formalize causal relationships by exploiting the temporal aspects of dynamical systems. In scenarios involving complex interactions, hidden variables, and feedback loops, the directed graph representation can elucidate causal connections that might be obscured in static models. By analyzing the dependencies along directed edges, researchers gain insights into the causal structure that underlies the observed dynamics.

Networked dynamical systems for causal inference find applications across diverse fields. In biology, they can aid in understanding regulatory networks and signaling pathways. In social sciences, they provide insights into information dissemination, opinion formation, and social influence. In engineering, they contribute to the analysis of interconnected control systems and synchronization phenomena. A compelling application of networked dynamical systems for causal inference lies in the realm of neuroscience, particularly in understanding brain activity and neural networks. The brain is a complex system where neurons interact with each other through intricate synaptic connections. Networked dynamical systems provide a framework to model how the activity of individual neurons influences and is influenced by neighboring neurons over time. By observing the spatiotemporal patterns of neuronal firing, researchers can infer causal relationships among different brain regions. Insights gained from networked dynamical systems can shed light on information processing, synchronization, and information flow within the brain, ultimately advancing our understanding of cognition, behavior, and neurological disorders.

2.1.2 Markov Property

At its core, the Markov property signifies that the future state of a system depends solely on its present state and is conditionally independent of its past states given this present state. In the context of networked dynamical systems, the Markov property translates into the idea that the evolution of a component's state is influenced primarily by the states of its direct predecessors, or its parents in the directed graph. Leveraging the Markov property in the context of networked dynamical systems contributes to the accurate representation and interpretation of complex interactions. This thesis relies on this property, where in order to know the state of the system we have:

$$X(0), ..., X(n-1), X(n), X(n+1), ..., \tag{2.1}$$

being $X(n)$ the state of the dynamical system at the instant n. The *Markov* property states that given $X(n-1)$, $X(n)$ is independent of $X(z)$ for any $z < n-1$. This means that the state $X(20)$ only relies on the information of $X(19)$, and all the other past instants can be discarded. This property can be defined as:

$$P(X(n) \in A | X(n-1), X(n-2), ..., X(0)) = P(X(n) \in A | X(n-1)), \tag{2.2}$$

where P is the probability of being in state A. The concept of interest pertains to the likelihood of transitioning into state A, where, conditioned upon the entirety of the historical states, this likelihood equates to the probability of transitioning into the same state A based solely on the antecedent state. This principle holds significance due to the prevailing circumstance wherein the network, signifying the intricate interplays between distinct elements or nodes, is often obscure. However, this intrinsic attribute simplifies the task of characterizing the Functional connectivity (FC) from the available samples.

2.2 Brain Network Neuroscience

The human brain, with its intricate web of neurons and synapses, remains one of the most enigmatic frontiers of scientific exploration. Recent advancements in neuroscience have unveiled that the brain's functionality is not just a product of isolated regions but a complex interplay of connections. Brain Network Neuroscience, often referred to as Connectomics, has emerged as a transformative field that employs network theory and analysis to decipher the brain's structural and functional connectivity patterns. This Section delves into the significance of Brain Network Neuroscience and its applications, shedding light on how the network perspective enhances our understanding of the brain's intricate workings. The human brain is inherently a network: a vast assembly of neurons that communicate and exchange information through intricate pathways. Brain Network Neuroscience leverages the principles of graph theory to represent and analyze these neural connections. In this context, the brain's structural and functional elements are mapped onto nodes and edges of a graph, respectively.

The structural connectome depicts the anatomical pathways that underlie neural communication. Techniques like Diffusion MRI (dMRI) enable the mapping of white matter tracts and pathways, creating a structural network representation of the brain. Nodes in this network represent brain regions, and edges denote the presence of white matter connections. Functional connectivity captures the synchronized activity between brain regions, often measured through techniques like functional Magnetic Resonance Imaging (fMRI) and Electroencephalography (EEG). Nodes in a functional network correspond to brain regions, and edges indicate the strength of correlation or synchronization in their activity patterns.

In [Alvaro Pascual-Leone, 2000] they use a procedure reffered to as Transcranial magnetic stimulation (TMS) that basically stimulates the nerve cells in the brain. With it they could trace the timing at which activity in a particular cortical region contributes to a given task and map the functional connectivity between the brain regions. In the work developed in [Adolphs, 2003] they conclude that social behaviour is the result of a whole set of processes between the different brain's regions. The main goal of that research was to understand the neural basis of our intuitive *folk psychology*, i.e., stereotyping, intentions, beliefs, etc. Disturbances in the structural and functional connectivity of *brain hubs*, [Martijn P. van den Heuvel, 2011], are linked to neuropathology. In this work they demonstrate that there is a tendency for high-degree nodes to be more more densely connected among themselves than nodes of a lower degree. This provides important information on the higher-level topology of the brain network. They define this higher-level regions as *rich clubs* and prove that not only they are individually central but also densely interconnected. It is also suggested and reinforced that the human brain network possesses a hierarchical assortative organization, a network topology in which high-degree nodes exhibit a tendency to be interconnected. Further, they conclude that the differente regions the brain do not operate as individual entities, but instead act as a strongly interlinked collective.

2.2.1 Applications

Cognitive Function: Brain network analysis offers insights into how cognitive functions emerge from coordinated neural activity. It has illuminated networks involved in memory, attention, language, and decision-making, revealing how these functions depend on the interactions between distinct brain regions. **Neuropsychiatric Disorders**: Aberrant brain connectivity is implicated in various neuropsychiatric disorders, such as schizophrenia, depression, and autism. Network analysis has provided potential biomarkers and novel insights into the underlying mechanisms of these conditions. **Aging and Development**: The brain's network architecture evolves across the lifespan. Studying how network properties change with age sheds light on developmental trajectories and age-related cognitive changes. **Neuroplasticity and Learning**: Brain networks adapt based on experience and learning. Network analysis has illuminated the plasticity mechanisms that underlie skill acquisition and recovery after brain injury.

2.3 Problem Formulation

The intricate dynamics of networked systems have captivated researchers across various disciplines, offering a rich tapestry of interactions that underlie the behavior of interconnected components. In the realm of Networked Dynamical Systems (NDS), the interplay between components, driven by complex relationships, has given rise to a new frontier in understanding complex systems. The manifestation of these interactions often emerges as time-series data, providing a dynamic window into the behavior of the system.

Our thesis centers around the fundamental questions: i) *Feasibility*— Can we consistently recover the causal network from the observed time series data of a linear networked dynamical system? ii) *Algorithm*— How can we process the time-series data to infer its underlying causal structure?

Time-series data, capturing the evolution of system states over time, offer a granular perspective on how components interact, respond, and influence each other. By observing the temporal evolution of states, we gain insights into not only the system's present behavior but also the echoes of its past and the glimpses of its future. In this thesis, these time-series are generated under colored noise, and we only have access to a subset of the total nodes (partial observability).

A stochastic linear NDS can be defined as:

$$\mathbf{y}(n) = A\mathbf{y}(n-1) + \mathbf{x}(n), \tag{2.3}$$

being $\mathbf{y}(\mathbf{n})$ the vector containing the states of the nodes at time n, defined as $\mathbf{y}(n) = [y_1(n)y_2(n)\dots y_N(n)]$, and N being the number of nodes in the NDS. $A \in \mathbb{R}_+^N$ is the non-negative interaction matrix whose support, $\mathrm{Supp}(A)$, is the underlying causal network linking the nodes. This matrix A is assumed stable, i.e., $\rho(A) < 1$, being $\rho(A)$ the spectral radius of A. This implies, in particular, that the linear NDS (2.3) is assumed to be stable. $(x(n))_{n\in\mathbb{N}}$ is the noise process applied to the N nodes in the system where $\mathbf{x}(n) \sim \mathcal{N}(0, \Sigma)$ and $\Sigma \in \mathbb{S}_+^N$, being Σ the noise covariance matrix.

If we assume that $y(0) = x(0)$ we can define $y(n) = \sum_{i=0}^{M} A^{n-i}\mathbf{x}(i)$ with M being the total number of samples. This is proven by taking $y(1)$ and $y(2)$ where, $\mathbf{y}(1) = A\mathbf{y}(0) + \mathbf{x}(1)$, with the assumption above, $\mathbf{y}(1) = A\mathbf{x}(0) + \mathbf{x}(1)$. And $y(2) = A\mathbf{y}(1) + \mathbf{x}(2)$, and then $y(2) = A(A\mathbf{x}(0) + \mathbf{x}(1)) + \mathbf{x}(2)$ will end up in $y(2) = A^2\mathbf{x}(0) + A\mathbf{x}(1) + \mathbf{x}(2)$.

The state of a node can be defined as the sum of all the interactions between its neighbours. The next figure shows an example of this scenario:

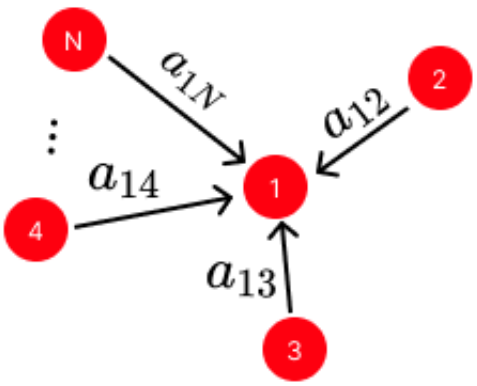


Figure 2.1: Node's state affection by its neighbours

In view of Fig. 2.1 we write equation (2.3) nodewise as

$$\mathbf{y}_i(n+1) = \sum_{j=1}^{N} a_{ij}\mathbf{y}_j(n) + \mathbf{x}_i(n+1), \tag{2.4}$$

being a_{ij} the interaction descriptor between node i and j. If $a_{ij} = 0$ then the node i does not have an edge to j, in the rest of the cases there is a link between the nodes. This stays coherent to the markov property explained in 2.1.2.

In *full* observability cases the main goal of the graph recovery is to consistently recover the support of the interaction matrix A, $Supp(A)$ from the time-series. However, as in this thesis we focus on partial observability scenarios, the goal is to consistently infer the support sub matrix of A, denoted as A_S.

Correlation matrices seamlessly transform into networks, where variables become nodes and correlations become edges. This network representation encapsulates the essence of the system's connections, transcending the limitations of individual correlations to offer a holistic view of the collective interplay. Nodes in this network signify variables, and the edges, which carry the weight of correlations, delineate the direct relationships between them. As such, correlation matrices not only quantify but also visualize the intricate web of dependencies that characterize the networked dynamical system. Let us define the k^{th}-lag correlation matrix as:

$$R_k(n) \triangleq E\left[y(n+k)y(n)^\top\right]. \tag{2.5}$$

From the time-series we can define the empirical k^{th}-lag as:

$$\widehat{R}_k(n) = \frac{1}{n}\sum_{l=0}^{n-1} \mathbf{y}(l+k)\mathbf{y}(l)^\top. \tag{2.6}$$

As the quantity of samples expands indefinitely, the empirical correlation matrices at the k^{th} lag tend to converge toward the actual correlation matrices. Our focus lies in regulating the relationship between the number of samples and the resulting accuracies, a matter commonly referred to as the sample complexity dilemma. The objective is to minimize the requisite number of samples for achieving a level of performance that rivals or surpasses that attained by other methodologies, ultimately aiming for optimal efficiency.

In the literature the main method to infer the underlying structure of the dynamical system is through an estimator that returns a scalar value for each pair of nodes. In our case, we want a matrix-value estimator that takes as input the time-series and returns a matrix as output, for any given $n \in \mathbb{N}$, defined as:

$$\begin{array}{rccc} F^{(n)}: & \mathbb{R}^{|S|\times n} & \longrightarrow & \mathbb{R}^{|S|\times|S|} \\ & \{[\mathbf{y}(\ell)]_S\}_{\ell=0}^{n-1} & \longmapsto & \mathcal{F}^{(n)} \end{array}, \tag{2.7}$$

The idea is that $\mathcal{F}_{ij}^{(n)}$ represents the link between node i and j with n observed samples. This way $\mathcal{F}^{(n)}$ can be described as the matrix of the estimated interactions between each pair of nodes. A matrix is deemed structurally consistent when its highest values align with the edges of interconnected node pairs, while the lowest values coincide with the edges connecting disconnected pairs. We formalize it with the following definition that we presented in [Machado et al., 2023].

Definition 1(structural consistency) A matrix-valued estimator $F^{(n)}$ is structurally consistent with high probability, whenever there exists a threshold τ so that,

$$\mathbb{P}\left(\mathcal{F}_{ij}^{(n)} > \tau\right) \overset{n\to\infty}{\longrightarrow} 1 \iff i \to j, \tag{2.8}$$

i.e., if the ij^{th} entry of $F^{(n)}$ is above the threshold τ, then there is an edge between the node i and j, with enough sumber of samples n.

in general, a matrix-valued estimator can be defined with an error term:

$$\mathcal{F}^{(n)} = \alpha A_S + \mathcal{E}_S^{(n)}, \tag{2.9}$$

with A_S being the ground-truth interaction matrix and $\mathcal{E}_S^{(n)}$ the error term. To keep the structural consistency of the matrix-valued estimator the error term could be different than 0. The constraint built arround the error term is defined as:

$$\mathsf{Osc}\left(\mathcal{E}_S^{(n)}\right) \triangleq \mathcal{E}_{\max}^{(n)} - \mathcal{E}_{\min}^{(n)} \leq \frac{\alpha A_{\min}^{+}}{2}, \tag{2.10}$$

where $\mathcal{E}_{\min}^{(n)}$ and $\mathcal{E}_{\max}^{(n)}$ are the minimum and maximum entries of the error matrix $\mathcal{E}_S$, A_min is the smallest positive entry of the interaction matrix, A_S, and $\mathsf{Osc}\left(\mathcal{E}_S^{(n)}\right)$ is the oscilation of a matrix $\mathcal{E}$. This means that if, and only if $\mathcal{E}_{\max}^{(n)} - \mathcal{E}_{\min}^{(n)}$ is small enough $\mathcal{F}^{(n)}$ is considered structurally consistent.

2.4 Artificial Neural Nets

Artificial Neural Networks (ANNs) are at the forefront of contemporary computational intelligence, drawing inspiration from the human brain's neural structure to enable machines to learn and solve complex tasks. With their capacity to capture intricate patterns, ANNs have transformed fields ranging from machine learning to image recognition, opening new avenues for solving challenges that were once considered insurmountable. At the core of ANNs lies the conceptualization of neurons as basic processing units. Just as biological neurons transmit signals, artificial neurons (also called perceptrons or nodes) process inputs, apply transformations, and generate outputs. These neurons are organized into layers, forming a network that emulates the structure of neural connections in the human brain.

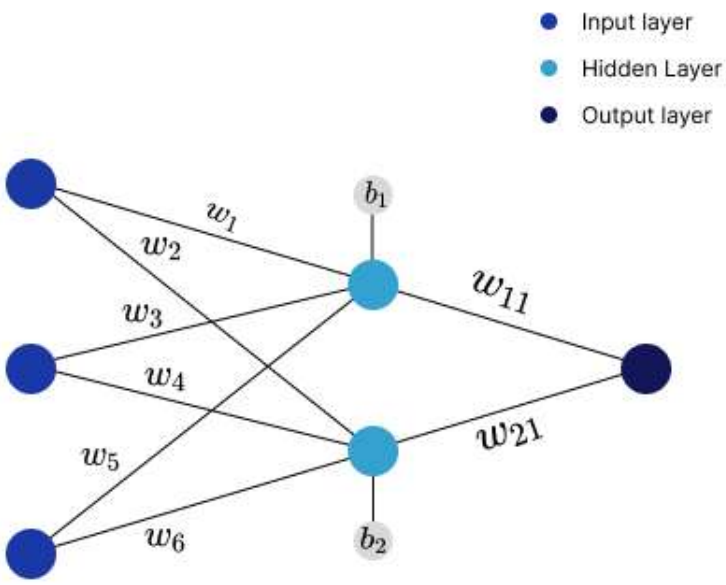


Figure 2.2: Example of an ANN structure - being w_n the weights, b_n the bias of the hidden layer defined by the method

Information flows unidirectionally from input to output through the hidden layers. During training, ANNs adjust the connections' weights to minimize the discrepancy between predicted outputs and actual outcomes. This optimization process, known as backpropagation, utilizes techniques like gradient descent to iteratively refine the network's weights. The learning rule in artificial neural networks (ANNs) is a fundamental mechanism that guides the process of adjusting the connection weights between neurons during training. The learning rule is responsible for allowing the network to learn from data, adapt to patterns, and improve its performance on specific tasks. There are two primary learning rules: supervised learning and unsupervised learning. Supervised learning is the most common learning paradigm in ANNs. It involves training the network using labeled examples, where the desired output is known for each input. The goal is for the network to learn a mapping from inputs to outputs, so it can accurately predict outputs for new, unseen inputs. Backpropagation involves two main steps: **Forward Pass** and **Backward Pass (Backpropagation):**. During the forward pass, the input is propagated through the network layer by layer to generate predictions. The difference between the predicted output and the actual target output (error) is calculated using a loss function. In the next step, the error is propagated backward through the network. The gradient of the error with respect to the weights of each neuron is computed. This gradient indicates the direction and magnitude of the weight adjustments needed to minimize the error. The computed gradients are used to update the weights of the network's connections. Gradient descent optimization methods adjust the weights by subtracting a fraction of the gradient. This process is performed iteratively over the entire training

dataset until the network's predictions converge to the desired outputs. The gradient descent equation for the update of the weights can be defined as:

$$\theta^{t+1} = \theta^t - \alpha \frac{\partial E(X, \theta^t)}{\partial \theta}, \tag{2.11}$$

being the weights and biases collectively denotes as θ and α the learning rate. The error function $\partial E(X, \theta^t)$, with X being the set of input-output pairs and θ^t the neural network parameters at the instant t.

Activation functions introduce non-linearity into ANNs, allowing them to approximate complex relationships between inputs and outputs. Common activation functions include the sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU). These functions introduce the capability to capture intricate patterns and behaviors that linear models often miss.

The resurgence of ANNs, often referred to as the deep learning revolution, has been fueled by the advancement of hardware, availability of massive datasets, and innovative algorithms. Deep neural networks, with numerous hidden layers, have demonstrated unprecedented capabilities in diverse applications, from natural language processing and speech recognition to autonomous driving and medical diagnosis. While ANNs have unlocked tremendous potential, their application raises ethical considerations such as bias in decision-making, data privacy, and the black-box nature of complex models. Efforts are underway to address these challenges, advocating for transparent and accountable AI systems.

Transfer learning leverages pre-trained neural networks on large datasets to enhance performance on new, related tasks with limited data. Pretrained models, such as those derived from ImageNet, serve as feature extractors that capture generalizable features from vast image datasets. These features can be fine-tuned to suit specific tasks, significantly reducing training time and resource requirements.

2.4.1 CNN - Convolutional Neural Nets

Convolutional Neural Networks (CNNs), also referred to as ConvNets, represent a specialized category of artificial neural networks uniquely tailored for the analysis and processing of visual data, specifically images and videos. CNNs have exhibited exceptional efficacy in diverse computer vision tasks, encompassing image classification, object detection, and image segmentation.

The underlying principle behind CNNs is to emulate the visual processing capabilities of the human brain. Drawing inspiration from the human visual system's adeptness in recognizing patterns, features, and hierarchical representations, CNNs comprise distinct layers that collectively contribute to comprehending the input data.

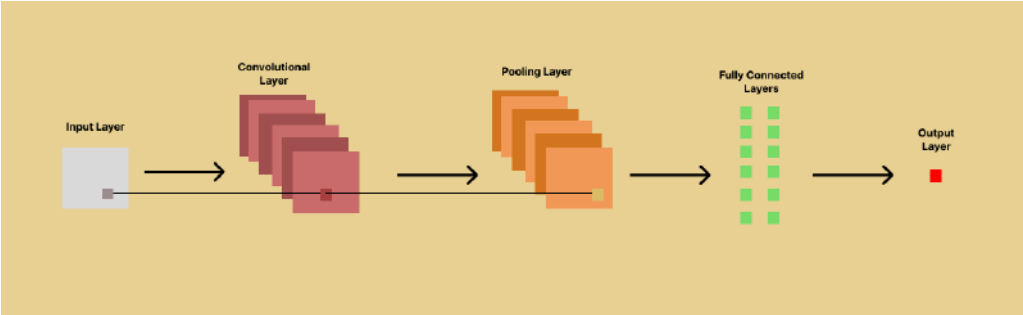


Figure 2.3: CNN architecture example

The fundamental constituents of a CNN are the convolutional layers. Within these layers, diminutive filters or kernels traverse the input image, effecting element-wise multiplication and summation to generate feature maps. These feature maps embody significant patterns and features identified within the input data. Early layers typically discern rudimentary features, such as edges, corners, or textures, while deeper layers capture more intricate and abstract features, pertinent to the given task.

To mitigate overfitting and introduce non-linearity, activation functions, such as Rectified Linear Unit (ReLU), are often employed after the convolutional layers. ReLU introduces non-linear transformations by nullifying negative values, thereby preserving positive values unaltered.

Pooling layers constitute another pivotal component of CNNs. Pooling serves to reduce the spatial dimensions of the feature maps, consequently diminishing computational complexity and engendering greater resilience to spatial translations and distortions in the input data. Max pooling represents a commonly employed pooling technique, selecting the maximum value within localized regions of the feature map.

CNNs also encompass fully connected layers, akin to conventional neural networks. These layers assimilate the high-level features extracted from the convolutional and pooling layers and ultimately yield the final output, often represented as a probability distribution across various classes in image classification tasks.

The training of CNNs typically entails the utilization of labeled datasets for supervised learning. Throughout the training process, the network's weights and biases are iteratively adjusted via backpropagation and optimization algorithms, such as gradient descent, with the objective of minimizing the discrepancy between predicted outputs and the actual ground truth labels.

The remarkable success of CNNs in diverse computer vision tasks has revolutionized the realm of artificial intelligence. These networks have evolved into indispensable tools in domains such as autonomous vehicles, medical image analysis, facial recognition, and numerous others. Moreover, CNNs have undergone adaptation to domains beyond vision, such as natural language processing and speech recognition, thus manifesting their versatility and potential across a broad spectrum of applications.

In summation, Convolutional Neural Networks represent a formidable class of neural networks, exemplifying exceptional prowess in visual data analysis and conferring machines the ability to discern and comprehend the environment in ways that were hitherto perceived as exclusively human capabilities. As ongoing research and technological advancements persist, CNNs are poised to play an increasingly critical role in shaping the future of artificial intelligence and computer vision applications.

2.4.2 FFNN - Feed Forward Neural Nets

Feed Forward Neural Networks (FFNNs), also known as multi-layer perceptrons, are a fundamental class of artificial neural networks that have revolutionized the field of machine learning and artificial intelligence. FFNNs are designed to process and analyze complex patterns in data, making them a powerful tool for a wide range of applications, including pattern recognition, function approximation, and classification.

Architecture

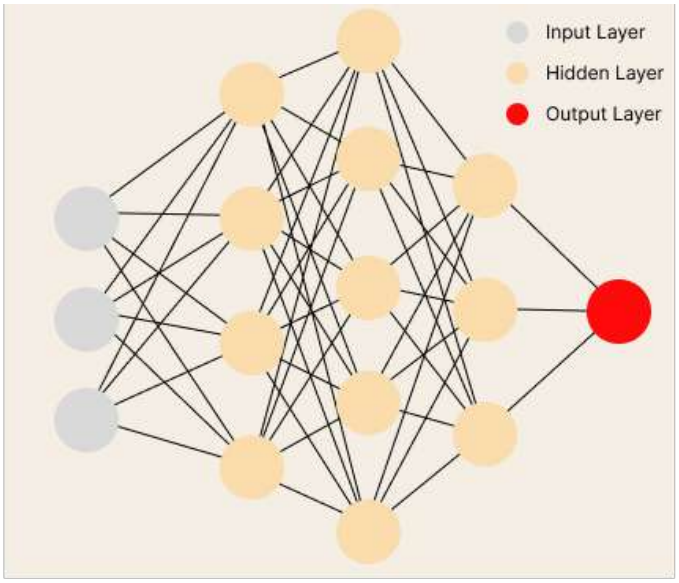


Figure 2.4: FFNN architecture example

At its core, an FFNN consists of multiple layers of interconnected artificial neurons or nodes. The network is typically organized into three types of layers: input layer, hidden layers, and output layer. The input layer receives the raw input data, while the output layer produces the final predictions or classifications. The hidden layers, as the name suggests, are intermediary layers between the input and output layers, where complex feature representations are learned.

Forward Propagation

FFNNs employ a feed-forward mechanism, where data flows through the network from the input layer to the output layer without any loops or feedback connections. The input data is fed into the input layer, and the activations of each neuron are computed based on weighted connections to the previous layer's activations. These activations are then passed through activation functions, introducing non-linearity to the network, and the process is repeated until the output layer produces the final result.

Activation Functions

Activation functions play a crucial role in FFNNs by introducing non-linearities to the model. Common activation functions include the Rectified Linear Unit (ReLU), which sets negative activations to zero, and the Sigmoid function, which squashes the activations to a range between 0 and 1. The choice of activation functions impacts the network's ability to learn complex representations and influences the overall performance of the model.

Training and Backpropagation

To make accurate predictions, FFNNs must be trained on labeled data. The training process involves adjusting the weights and biases of the neurons in the network to minimize the difference between predicted outputs and the actual ground truth labels. This optimization is achieved using an algorithm called backpropagation, which computes the gradients of the loss function with respect to the network parameters. These gradients are then used to update the weights and biases through various optimization techniques, such as stochastic gradient descent (SGD) or Adam.

Evaluation and Performance

Evaluating the performance of an FFNN is essential to assess its effectiveness on unseen data. Various metrics, such as accuracy, precision, recall, and F1-score, are commonly used to gauge the model's performance in classification tasks. For regression tasks, mean squared error (MSE) or mean absolute error (MAE) are commonly used evaluation metrics. Cross-validation is often employed to validate the model's generalization capability and identify potential overfitting issues.

Applications

FFNNs have found widespread application in numerous fields. In computer vision, they are used for image recognition, object detection, and facial recognition. In natural language processing, they are employed for sentiment analysis,

language translation, and text generation. FFNNs are also used in finance for predicting stock prices, in healthcare for disease diagnosis, and in robotics for controlling autonomous systems.

Limitations and Future Directions

Despite their success, FFNNs have some limitations, such as the need for large amounts of labeled data, computational complexity in deep networks, and difficulty in interpretability. Researchers are continuously exploring techniques to address these challenges, including transfer learning, unsupervised pre-training, and neural architecture search, to enhance FFNNs' performance and scalability.

In conclusion, Feed Forward Neural Networks are a foundational class of artificial neural networks that have significantly impacted the field of machine learning and artificial intelligence. Their ability to learn complex patterns and generalize to unseen data makes them a valuable tool in various applications, propelling advancements in diverse domains and paving the way for future breakthroughs in AI research.

2.5 Clustering Methods

2.5.1 K-means

K-means is a widely employed unsupervised clustering method in machine learning and data analysis, serving the purpose of partitioning a given dataset into K distinct clusters. The fundamental objective of K-means is to identify cluster centers (centroids) in such a way that data points within each cluster exhibit greater similarity or proximity to their respective centroids than to those belonging to other clusters.

Initialization

The K-means algorithm begins by setting the number of desired clusters, denoted as K, based on prior knowledge or domain-specific requirements. The next step entails the random selection of K initial centroids from the dataset or through alternative initialization techniques. These initial centroids serve as the starting points for the clustering process. In this project, the number of centroids is known and they are two(connected nodes and disconnected nodes).

Assignment Step

In the assignment step, every data point in the dataset is allocated to the nearest centroid based on a specified distance metric, commonly the Euclidean distance.

This operation effectively segregates the data points into K clusters according to their proximity to the respective centroids.

Update Step

Subsequently, the centroids of the K clusters are recalculated during the update step. The new centroids are computed as the mean of all data points assigned to each cluster. This recalculation process refines the positions of the cluster centers.

Iteration

The assignment and update steps are repeated iteratively until a specified stopping criterion is met. Common stopping criteria include reaching a predetermined maximum number of iterations or when the positions of the centroids stabilize and remain unchanged across successive iterations.

Convergence

K-means achieves convergence when the centroids cease to shift significantly between consecutive iterations, and data points no longer alter their cluster assignments. At this point, the algorithm has successfully identified the optimal K clusters, and the process concludes.

Final Output

The final outcome of the K-means algorithm comprises a set of K clusters, each represented by its corresponding centroid, along with the data points assigned to each cluster. Collectively, these clusters constitute a partitioning of the original dataset, with each data point associated with a particular cluster.

Considerations

The choice of initial centroids in the initialization step can influence the clustering outcome. As a result, K-means may be executed multiple times using different random initializations. The clustering result yielding the most favorable outcome, typically based on a defined criterion such as minimizing within-cluster variance, is selected as the final clustering solution.

K-means is a versatile technique with diverse applications across various domains, including image segmentation, customer segmentation, data compression, and anomaly detection. Nevertheless, selecting an appropriate value for K is essential, as an incorrect choice can lead to suboptimal clustering results. Techniques such as the elbow method or silhouette analysis are commonly employed to determine the optimal K value, taking into account the inherent characteristics of the data and the specific clustering task.

In conclusion, K-means clustering represents a prominent and widely utilized unsupervised learning method, pivotal in data analysis and pattern recognition. By iteratively refining cluster centers and partitioning data into clusters based on similarity, K-means efficiently extracts meaningful structures from unlabelled data, contributing significantly to various applications and domains in the field of machine learning.

2.5.2 Gaussian-Mixture

The Gaussian Mixture Model (GMM) clustering method is a powerful and widely-used unsupervised learning algorithm for data segmentation and clustering tasks. Unlike K-means, which assigns data points to distinct non-overlapping clusters, GMM represents clusters as a weighted combination of Gaussian distributions, allowing for flexible and probabilistic cluster assignments.

Probabilistic Representation

In GMM, each cluster is modeled as a Gaussian distribution with its mean and covariance. The dataset is assumed to be generated by a mixture of these Gaussian components, where each data point is assigned a probability of belonging to each cluster. This probabilistic representation accommodates data points that lie near the boundaries between clusters and assigns them to multiple clusters with varying probabilities.

Expectation-Maximization Algorithm

The GMM clustering process involves the Expectation-Maximization (EM) algorithm. The EM algorithm iteratively estimates the parameters of the Gaussian distributions and the probabilities of data point assignments to the clusters.

a. Expectation Step (E-step): In the E-step, the algorithm calculates the posterior probabilities of data points belonging to each cluster, given the current parameters of the Gaussian distributions. These posterior probabilities, often referred to as responsibility or membership probabilities, are computed using Bayes' theorem and determine how much each data point contributes to each cluster.

b. Maximization Step (M-step): In the M-step, the algorithm updates the parameters of the Gaussian distributions based on the weighted contributions of data points to each cluster. The mean and covariance of each Gaussian are adjusted to maximize the likelihood of the data under the current mixture model.

Initialization

To initiate the GMM clustering, initial estimates of the Gaussian parameters are required. Commonly used techniques include K-means initialization or randomly

sampling data points to initialize the means and identity matrices for the covariance matrices.

Convergence

The EM algorithm iteratively performs the E-step and M-step until the estimated parameters converge to a stable solution or a predefined convergence criterion is met. The algorithm guarantees monotonic improvement in the likelihood of the data with each iteration.

Cluster Assignment

Once the GMM model converges, the data points can be assigned to the clusters based on their highest posterior probabilities. Alternatively, data points can be assigned to multiple clusters with varying probabilities to represent uncertainty in the clustering assignment.

Choosing the Number of Clusters

Selecting the optimal number of clusters in GMM can be achieved using techniques like the Bayesian Information Criterion (BIC) or cross-validation. These methods evaluate the trade-off between model complexity and the fit to the data to avoid overfitting. In this project the number of clusters is two (connected and disconnected).

Applications

GMM clustering finds numerous applications, including image segmentation, speech recognition, anomaly detection, and data compression. It is particularly useful when the underlying data distribution is complex and multi-modal, as GMM can capture such complexity using a combination of Gaussian components.

Conclusion

Gaussian Mixture Model (GMM) clustering offers a powerful and flexible approach to data segmentation, providing a probabilistic representation that accommodates uncertain and overlapping clusters. Its ability to model complex data distributions makes it a valuable tool for a wide range of real-world applications, enabling data scientists and researchers to extract meaningful insights from their data and uncover underlying structures.

Chapter 3

Related Work

In the present Chapter, our discussion revolves around some relevant related works in the causal inference literature. In particular, some of the causal inference estimatiors discussed in this Chapter will be used to benchmark our approach in Chapter 5. As we will discuss, some of the methods discussed are tailored to linear NDS and withstand well the curse of partial observability. However, most methods loose their technical guarantees when compounding colored noise to the partial observability setting. This is still a seldom explored framework in the literature.

In Section 5, the estimators deployed for the purpose of benchmarking the performance of our method encompass the **Granger** estimator [Geiger et al., 2015; Matta et al., 2020c; Santos et al., 2020], the **Precision** matrix algorithm, the recent NIG $R_1 - R_3$ estimator [Chen et al., 2022], the R_1 estimator [Matta et al., 2020a], and lastly, the recent feature based approach [Machado et al., 2023]. The consistency of the aforementioned estimators is intricately contingent upon the inherent generative mechanism governing the time-series data. In the event that the time-series are drawn from a Gaussian multivariate distribution, the utilization of the Precision matrix represents a consistent approach for deducing the latent network structure. On the other hand, in scenarios where the samples stem from a linear dynamical system, the Granger estimator conforms to a consistent estimator.

3.1 Granger estimator

The Granger estimator is a statistical tool commonly used in econometrics and time series analysis to assess the causal relationship between two or more variables. It is named after Clive Granger, who was awarded the Nobel Prize in Economics in 2003 for his work on analyzing time series data. At its core, the Granger estimator helps determine whether past values of one variable can provide useful information in predicting another variable. In essence, it quantifies whether the historical values of one variable *Granger-cause* changes in another variable. This concept is rooted in the idea that if past values of variable X significantly improve the prediction of variable Y, then it can be inferred that X Granger-causes Y.

From the time-series coming from the system (2.3) $\{\mathbf{y}(n)\}_{n=1}^{\infty}$, if we multiply both sides by $\mathbf{y}(n)^\top$, we have

$$\mathbf{y}(n+1)\mathbf{y}(n)^\top = A\mathbf{y}(n)\mathbf{y}(n)^\top + \mathbf{x}(n+1)\mathbf{y}(n)^\top, \tag{3.1}$$

which yields,

$$E\left[\mathbf{y}(n+1)\mathbf{y}(n)^\top\right] = AE\left[\mathbf{y}(n)\mathbf{y}(n)^\top\right], \tag{3.2}$$

and then,

$$\begin{aligned} A &= \mathbb{E}\left[\mathbf{y}(n+1)\mathbf{y}(n)^\top\right]\left(\mathbb{E}\left[\mathbf{y}(n)\mathbf{y}(n)^\top\right]\right)^{-1} \\ &= R_1(n)R_0^{-1}(n) \overset{n\to\infty}{\longrightarrow} R_1(R_0)^{-1}, \end{aligned} \tag{3.3}$$

finally the estimator can be written as $R_1\left(R_0\right)^{-1}$ and can consistently estimate the underlying interaction matrix A. The equality of the final step (3.3) is done by having the one-lag correlation matrix $R_1 \triangleq E\left[\mathbf{y}(n+1)\mathbf{y}(n^\top)\right]$ and the correlation matrix $R_0 \triangleq E\left[\mathbf{y}(n)\mathbf{y}(n)^\top\right]$.

In large scale networks only some of the network nodes can be observed. The *Granger* estimator is proven to ve structually consistent at full observability cases, but can we still use it for partial observability scenarios? Under partial observability we have the following:

$$\widehat{A}_S = [R_1]_S\left([R_0]_S\right)^{-1} \neq \left[R_1\left(R_0\right)^{-1}\right]_S = A_S, \tag{3.4}$$

where S is the subset of the observable nodes and A_S is the true interation matrix of the nodes belonging to S. Being $\widehat{A}_S$ the *Granger* estimated interation matrix. Even with this result (3.4) where part of the causal information is lost, under certain regimes. Granger $\widehat{A}_S$ is provably still structurally consistent under partial observability and diagonal noise [Matta et al., 2020c, 2022; Santos et al., 2020]. However, as demonstrated in the numerical experiments, Chapter 5, this estimator exhibits poor performance under partial observability and colored noise.

3.2 Precision matrix estimator

The precision matrix, often denoted as the inverse covariance matrix or concentration matrix, is a fundamental concept in statistical analysis, particularly in the realm of multivariate data. It holds a crucial role in modeling relationships between variables and is derived from the covariance matrix, which quantifies the extent to which variables change together. The precision matrix provides valuable insights into the conditional dependencies between variables, elucidating the

direct influence that each variable has on another while controlling for the effects of other variables. Time series data entails observations of a variable recorded over discrete time intervals. These observations are often correlated due to temporal dependencies, meaning that the value of a variable at one time point can influence its value at subsequent time points. As such, the precision matrix derived from time series data encodes the strength and nature of these temporal dependencies, helping to uncover the underlying structure and interactions within the dataset.

The precision matrix is calculated by the inversion of the correlation matrix $R_0(n) = E\left[\mathbf{y}(n)\mathbf{y}(n)^\top\right]$, then $[R_0]^{-1} = E\left[\mathbf{y}(n)\mathbf{y}(n)^\top\right]^{-1}$, which leads to $\widehat{R}_0 = \frac{1}{M}\sum_{n=0}^{M}\mathbf{y}(n)\mathbf{y}(n)^\top$. Then, R_0 can be described too as $R_0 = \sum_{i=0}^{n} A^i \Sigma A^i$. Considering diagonal noise, $\Sigma = \sigma^2 I$, $R_0 = \sigma^2\left(I - A^2\right)^{-1}$.

In this case, the precision matrix can be written:

$$[R_0]^{-1} = \frac{1}{\sigma^2}\left(I - A^2\right). \tag{3.5}$$

This estimator under partial observability can have errors due to the need of all matrix entries in order to calculate correctly its inverse. In the numeric simulations the precision matrix has better performance than the other estimators, but the technical proof of its consistency is yet to be formuled. In the colored noise cases, $[R_0]^{-1} = \left[\sum_{i=0}^{n} A^i \Sigma A^i\right]^{-1}$.

3.3 $R_1 - R_3$ estimator

The NIG $R1 - R3$ estimator was proposed and studied recently [Chen et al., 2022]. It is tailored for linear NDS under diagonal noise. In particular, it conforms to an unbiased consistent estimator under partial observability and diagonal noise, i.e., it can recover the interaction submatrix A_S up to small error with high probability. We recall that by diagonal noise, we mean that the covariance matrix Σ_x of the noise process is a multiple of the identity $\Sigma_x = \sigma^2 I$.

Firstly, remark that from equation (2.5) we can derive R_1 as

$$R_1 = E\left[\mathbf{y}(n+1)\mathbf{y}(n)^\top\right], \tag{3.6}$$

and, by having the system (2.3) we have:

$$\begin{aligned} R_1 &= E\left[\left(A\mathbf{y}(n) + \mathbf{x}(n+1)\right)\mathbf{y}(n)^\top\right] \\ &= AE\left[\mathbf{y}(n)\mathbf{y}(n)^\top\right] + E\left[\mathbf{x}(n+1)\mathbf{y}(n)^\top\right]. \end{aligned} \tag{3.7}$$

Having $R_0 = E\left[\mathbf{y}(n)\mathbf{y}(n)^\top\right]$:

$$R_1 = AR_0. \tag{3.8}$$

Extending this via induction:

$$R_k = A^k R_0. \tag{3.9}$$

From (2.4) and (2.6), $R_0 = I + A^2 + A^4 + \cdots = \sum_{i=0}^{2i} A^{2i} = \left(I - A^2\right)^{-1}$. Therefore $R_1 = A + A^3 + A^5 + (\dots)$ and $R_3 = A^3 + A^5 + A^7 + (\dots)$, the $R_1 - R_3$ estimator infers consistently the interation matrix A as $R_1 - R_3 = A$. Applying the diagonal noise to the same paradigm, having in consideration the demonstration done with (2.3),

$$R_0 = \sum_{i=0}^{n} A^i \Sigma A^i, \tag{3.10}$$

and since $\Sigma = \sigma^2 I$, the difference $R_1 - R_3$ can be defined as $\sigma^2 A$.

3.4 R_1 estimator

This estimator consists in the recovery of the interaction matrix with only the one-lag correlation matrix, R_1. We can define R_1 as:

$$\begin{aligned} R_1(n) &= E\left[\mathbf{y}(n+1)\mathbf{y}(n)^\top\right] \\ &= E\left[\left(A\mathbf{y}(n) + \mathbf{x}(n+1)\right)\mathbf{y}(n)^\top\right] \\ &= AE\left[\mathbf{y}(n)\mathbf{y}(n)^\top\right] + E\left[\mathbf{x}(n+1)\mathbf{y}(n)^\top\right], \end{aligned} \tag{3.11}$$

with $E\left[\mathbf{x}(n+1)\right] = 0$ as $(\mathbf{x}(n))$ is zero mean and remarking that $\mathbf{x}(n+1)$ is independent of $\mathbf{y}(n)$,

$$R_1(n) = AR_0(n). \tag{3.12}$$

Therefore, considering (3.10) and diagonal noise $\Sigma = \sigma^2 I$, we have

$$R_1(n) = \sigma^2(A + A^3 + A^5 + \dots). \tag{3.13}$$

Considering partial observability, as the estimator is a correlation matrix, being S the set of observable nodes,

$$[R_1]_S = \sigma^2([A]_S + \left[A^3\right]_S + \left[A^5\right]_S + \dots) = \sigma^2 [A]_S + \mathcal{E}_S, \tag{3.14}$$

where $\mathcal{E}_S$ is an error term in the estimation of A_S (modulo a multiplicative constant). The error term has been studied in [Matta et al., 2020a] to establish the structural consistency of R_1 under partial observability and diagonal noise.

3.5 Feature based Approach

In the feature based causal inference approach, first proposed in [Machado et al., 2023], a feature vector is estimated to each pair of nodes from the time series. Reference [Machado et al., 2023] proposed a feature assignment so that there exists a hyperplane that separates consistently the features, with features associated with connected pairs lying on one side of the hyperplane and features of disconnected pairs on the other side. This was established formally for linear NDS under partial observability and diagonal noise. In particular, this allowed the successful deployment of supervised methods to cluster the features and perform causal inference. More concretely, the features were defined as

$$\mathcal{F}_{ij}(n) \triangleq \left(\left[\widehat{R}_D(n) \right]_{ij}, \left[\widehat{R}_{D+1}(n) \right]_{ij}, \ldots, \left[\widehat{R}_L(n) \right]_{ij} \right), \tag{3.15}$$

for each pair ij. Further, these features were normalized by the max feature value:

$$\mathbb{F}_{ij}(n) = \frac{\mathcal{F}_{ij}(n)}{max(\mathcal{F}_{ij}(n))}, \tag{3.16}$$

for $i, j = 1, 2, \ldots, N$, where N is the number of nodes. In the study conducted by [Machado et al., 2023], it has been shown that the features were linearly separable whenever $D \leq 1$ and $L \geq 3$. Subsequent to the normalization process, the aforementioned features serve as input to a Convolutional Neural Network (CNN) module, yielding an array that effectively represents the interconnection between each pair of nodes. The study employs a total of 200 features, comprising the initial set of 100 negative correlation lag matrices followed by another set of 100 representing positive correlation lag matrices.

The methodology outlined in the paper, along with the estimators discussed in Section 3.3 and in Section 3.4, exclusively relies on the time-series data pertaining to each node pair (i and j) to facilitate the computation of their respective correlation matrix entries. Conversely, Section 3.1 and Section 3.2 necessitate a more comprehensive approach involving the visualization of all entries for the inversion of the correlation matrix (R_0). Notably, the scope of the research done in [Machado et al., 2023] encompasses systems affected by diagonal noise and partial observability.

In a subsequent phase of the investigation, this thesis research focus is broadened to encompass colored noise regimes. This extension reveals a noteworthy finding: the CNN module's performance diminishes notably when subjected to colored noise conditions.

Chapter 4

Methodology and Technical Results

4.1 Technical Results

The results reported in this Chapter were submitted for publication [Santos et al., 2023]. They establish conditions on the noise structure, namely, on the covariance matrix Σ_x of the noise process where the underlying causal inference problem (under partial observability and colored noise) is feasible, namely, there is an algorithm to consistently recover the underlying causal network linking the observed set of nodes.

Assumption 1. As mencioned in Section 2.3 the interaction matrix A is said to be stable, i.e, $\rho(A) < 1$, where $\rho(A)$ is the spectral radius of A. Additionally, A is assumed to be symmetric and nonnegative.

Assumption 2 We assume that $\sigma^2 = E\left[\mathbf{x}_i^2\right]$ for all i.

Remark that $\sigma^2 \geq E\left[\mathbf{x}_i\mathbf{x}_j\right]$ for all i, j. Indeed,

$$\begin{aligned} 0 \leq E[(\mathbf{x}_i - \mathbf{x}_k)^2] &= E[\mathbf{x}_i^2] + E[\mathbf{x}_k^2] - 2E[\mathbf{x}_i\mathbf{x}_k] \\ &= 2\sigma^2 - 2E[\mathbf{x}_i\mathbf{x}_k]. \end{aligned} \tag{4.1}$$

Assumption 3**.** It is assumed that:

$$\sigma^2 > \mathbb{E}\left[\mathbf{x}_i\mathbf{x}_j\right] = [\Sigma_x]_{ij}\,, \tag{4.2}$$

for all $i \neq j$. That is, the off-diagonal entries of the noise covariance matrix are strictly smaller than the diagonal.

4.1.1 Impact of the Colored Noise

This thesis focus on Networked Dynamical Systems where the noise structure has non-zero entries on the off-diagonal of the covariance noise matrix. Extending the equation in (2.3) we get:

$$\mathbf{y}(n+1) = A\mathbf{y}(n) + \underbrace{\alpha \cdot x_1(n+1)}_{\mathcal{N}_1} + \underbrace{\frac{\beta}{\sqrt{N}} \cdot 1 \cdot 1^\top \cdot x_2(n+1)}_{\mathcal{N}_2}. \tag{4.3}$$

The noise components within our model, denoted as $\mathcal{N}_1$ and $\mathcal{N}_2$, serve distinct roles in the characterization of the Networked Dynamical System (NDS). $\mathcal{N}_1$ represents the diagonal noise, wherein solely the diagonal elements of the noise structure are influenced by the parameter α. Conversely, $\mathcal{N}_2$ signifies the colored noise, with β being the pivotal factor governing the influence of off-diagonal entries in the noise structure. Consequently, a higher value of β amplifies the impact of colored noise within the time-series data of each node. As β increases, the task of extracting information about the underlying structure of the Networked Dynamical System from the time-series data becomes progressively more challenging due to the heightened complexity at the node level. The results presented in Section 5 demonstrate that, with elevated values of β, an increased number of samples is required to achieve accurate classification of the functional connectivity.

The features employed in our analysis are constructed from the time-series data originating from the Networked Dynamical System (NDS). Consequently, the introduction of colored noise imparts an influence on the feature vector. In essence, it has been observed that higher values of β lead to increased correlation values within the time-series data, subsequently affecting the feature set. In brief, the behavior of the feature vector with the inclusion of colored noise can be described as a *shift* in the features, signifying that the centroid of the feature vector moves further away from the origin. This phenomenon is schematically depicted in Fig. 4.1.

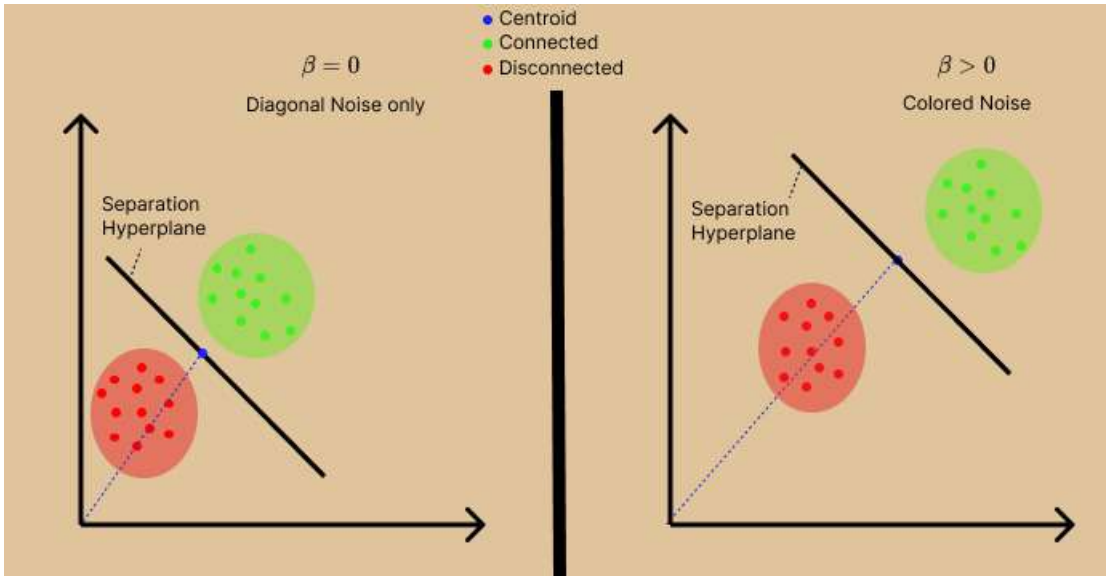


Figure 4.1: Shift simulation of the Features

4.1.2 Error characterization

In this Section, we present a demonstration of error characterization and subsequently define the theorem pertaining to the error associated with the difference between R_1 and R_3.

Remark that R_0 can be defined as:

$$R_0(n) = \sum_{i=0}^{n} A^i \Sigma A^i, \tag{4.4}$$

since A is a stable matrix, $A^n E\left[\mathbf{y}(0)\mathbf{y}(0)^\top\right] A^n$ converges to 0. Resulting, $R_0 = \sum_{i=0}^{n} A^i \Sigma_x A^i$,when $n \to \infty$. Σ_x being the covariance matrix of the noise.

And:

$$R_k = A^k R_0. \tag{4.5}$$

With (4.4) and (4.5):

$$\begin{aligned} R_k &= A^k R_0 \\ &= A^k \textstyle\sum_{i=0}^{\infty} A^i \Sigma_x A^i, \end{aligned} \tag{4.6}$$

we can decompose $\Sigma_x = \sigma_{gap}^2 I + \beta \mathbf{1}\mathbf{1}^\top + \overline{\Sigma}$, being $\sigma_{gap}^2 I$ a diagonal matrix, $\beta \mathbf{1}\mathbf{1}^\top$ the average *offset* matrix and $\overline{\Sigma}$ is the variability of the off-diagonal entries of Σ_x:

$$\begin{aligned} R_k &= A^k \textstyle\sum_{i=0}^{\infty} A^i \left(\sigma_{gap}^2 I + \beta \mathbf{1}\mathbf{1}^\top + \overline{\Sigma}\right) A^i \\ &= A^k \textstyle\sum_{i=0}^{\infty} \left(A^i \sigma_{gap}^2 I + A^i \beta \mathbf{1}\mathbf{1}^\top + A^i \overline{\Sigma}\right) A^i \\ &= A^k \textstyle\sum_{i=0}^{\infty} A^i \sigma_{gap}^2 I A^i + A^k \sum_{i=0}^{\infty} A^i \beta \mathbf{1}\mathbf{1}^\top A^i + A^k \sum_{i=0}^{\infty} A^i \overline{\Sigma} A^i \\ &= \sigma_{gap}^2 A^k \textstyle\sum_{i=0}^{\infty} A^{2i} + \beta A^k \sum_{i=0}^{\infty} A^i \mathbf{1}\mathbf{1}^\top A^i + A^k \sum_{i=0}^{\infty} A^i \overline{\Sigma} A^i. \end{aligned} \tag{4.7}$$

This way, $R_1 = \sigma_{gap}^2 A \sum_{i=0}^{\infty} A^{2i} + \beta A \sum_{i=0}^{\infty} A^i \mathbf{1}\mathbf{1}^\top A^i + A \sum_{i=0}^{\infty} A^i \overline{\Sigma} A^i$ and $R_3 = \sigma_{gap}^2 A^3 \sum_{i=0}^{\infty} A^{2i} + \beta A^3 \sum_{i=0}^{\infty} A^i \mathbf{1}\mathbf{1}^\top A^i + A^3 \sum_{i=0}^{\infty} A^i \overline{\Sigma} A^i$. Now $R_1 - R_3$ results:

$$\begin{aligned} &\sigma_{gap}^2 A \textstyle\sum_{i=0}^{\infty} A^{2i} - \sigma_{gap}^2 A^3 \sum_{i=0}^{\infty} A^{2i} \\ &= \sigma_{gap}^2 (\textstyle\sum_{i=0}^{\infty} A^{2i+1} - \sum_{i=0}^{\infty} A^{2i+3}) \\ &= \sigma_{gap}^2 A. \end{aligned} \tag{4.8}$$

Consider $A\mathbf{1} = \rho\mathbf{1}$ then:

$$
\begin{aligned}
&\beta A \textstyle\sum_{i=0}^{\infty} A^i \mathbf{1}\mathbf{1}^\top A^i - \beta A^3 \sum_{i=0}^{\infty} A^i \mathbf{1}\mathbf{1}^\top A^i \\
&= \beta \left(\textstyle\sum_{i=0}^{\infty} A^{i+1} \mathbf{1}\mathbf{1}^\top A^i - A^2 \sum_{i=0}^{\infty} A^{i+1} \mathbf{1}\mathbf{1}^\top A^i \right) \\
&= \beta \left(I - A^2 \right) \textstyle\sum_{i=0}^{\infty} A^{i+1} \mathbf{1}\mathbf{1}^\top A^i \\
&= \beta \left(I - A^2 \right) A \textstyle\sum_{i=0}^{\infty} \rho^i \mathbf{1}\mathbf{1}^\top \rho^i \\
&= \beta \left(I - A^2 \right) A \mathbf{1}\mathbf{1}^\top \textstyle\sum_{i=0}^{\infty} \rho^{2i} \\
&= \beta \left(I - A^2 \right) \rho \mathbf{1}\mathbf{1}^\top \tfrac{1}{1-\rho^2} \\
&= \beta\rho \left(\mathbf{1} - A^2 \mathbf{1} \right) \mathbf{1}^\top \tfrac{1}{1-\rho^2} \\
&= \tfrac{\beta\rho\left(\mathbf{1}-\rho^2\right)}{1-\rho^2} \mathbf{1}\mathbf{1}^\top \\
&= \beta\rho \mathbf{1}\mathbf{1}^\top .
\end{aligned}
\tag{4.9}
$$

Finally:

$$
\begin{aligned}
&A \textstyle\sum_{i=0}^{\infty} A^i \overline{\Sigma} A^i - A^3 \sum_{i=0}^{\infty} A^i \overline{\Sigma} A^i \\
&= \textstyle\sum_{i=0}^{\infty} A^{i+1} \overline{\Sigma} A^i - A^2 \sum_{i=0}^{\infty} A^{i+1} \overline{\Sigma} A^i \\
&= \left(I - A^2 \right) \textstyle\sum_{i=0}^{\infty} A^{i+1} \overline{\Sigma} A^i .
\end{aligned}
\tag{4.10}
$$

The previous argument yields the following theorem.

Theorem 1. *Under the Assumption* 1*, the NDS yields*

$$
\widehat{R}_1(n) - \widehat{R}_3(n) \overset{n\to\infty}{\longrightarrow} \sigma_{gap}^2 A + \underbrace{\beta\rho \mathbf{1}\mathbf{1}^\top + \left(I - A^2 \right) \left(\sum_{i=0}^{\infty} A^{i+1} \overline{\Sigma} A^i \right)}_{=:\mathcal{E}}, \tag{4.11}
$$

and under partial observability:

$$
\left[\widehat{R}_1(n) \right]_S - \left[\widehat{R}_3(n) \right]_S \overset{n\to\infty}{\longrightarrow} A_S + \mathcal{E}_S . \tag{4.12}
$$

4.1.3 Separability & Stability under noise

In this Section, we present and formalize a theorem that establishes the necessary and sufficient conditions for the consistent recovery of the underlying structure of the Nonlinear Dynamical System (NDS). Within this formal condition, it is assured that the information contained within the observable data is retrievable.

The *oscilation* of a matrix is the difference between the maximum and minimum entries of the matrix and be be defined as:

$$Osc(\mathcal{E}) = \mathcal{E}_{max} - \mathcal{E}_{min}. \tag{4.13}$$

For the next demonstrations let us define some important properties:

Property 1 Given $\overline{A}$, a symmetric stochastic matrix:

$$Osc\left(\overline{A}v\right) \leq Osc\left(v\right), \tag{4.14}$$

with each entry of $\overline{v} \triangleq \overline{A}v$ lying in the convex hull [Hiriart-Urruty and Lemaréchal, 2001] of the set $\{v_1, v_2, \ldots, v_N\}$ of the entries of the vector $v \in \mathbb{R}^N$, particularly, $\overline{v}_i \in [v_{min}, v_{max}]$ for all i.

Property 2

$$Osc(\alpha v) = |\alpha| Osc(v), \tag{4.15}$$

for any $v \in \mathbb{R}^N$ and $\alpha \in \mathbb{R}$. If v_{max} is the max entry of a matrix than, $\alpha v_{max} > \alpha v_i$ for any $i = 1, 2, \ldots, N$.

Property 3

$$Osc\left(B + C\right) \leq Osc(B) + Osc(C), \tag{4.16}$$

for any $B, C \in \mathbb{R}^N$

Property 4

If $Osc\left(Bv\right) \leq k_b Osc(v)$ and $Osc\left(Cv\right) \leq k_c Osc(v)$,

then, $Osc\left(CBv\right) \leq k_c Osc\left(Bv\right) \leq k_b k_c Osc\left(v\right)$,

for all $v \in \mathbb{R}^N$ with $k_b, k_c > 0$.

Having $A = \rho\overline{A}$ and the properties $(p_1, p_2, p_3, p_4$:

$$\begin{aligned} Osc\left(\left(I - A^2\right)V\right) &\leq Osc(V) + Osc\left(A^2 V\right) \\ &\leq Osc(V) + Osc\left(\left(\rho\overline{A}\right)^2 V\right) \\ &\leq Osc(V) + \rho^2 Osc\left(\overline{A}^2 V\right) \\ &\leq \left(1 + \rho^2\right) Osc(V). \end{aligned} \tag{4.17}$$

Then:

$$\begin{aligned} Osc\left(\left(I-A^2\right)V\right) &= Osc\left(\left(I-A^2\right)\rho\overline{A}V\right) \\ &= \rho Osc\left(\left(I-A^2\right)\overline{A}V\right) \\ &\leq \rho\left(1+\rho^2\right)Osc(V). \end{aligned} \tag{4.18}$$

If we consider $\sum_{i=0}^{\infty} A^i\overline{\Sigma}A^i$ as a monotonous function (with $\rho(A) < 1$) we can define the following:

$$\frac{\overline{\Sigma}_{min}}{1-\rho^2}\mathbf{1}\mathbf{1}^\top = \overline{\Sigma}_{min}\sum_{i=0}^{\infty} A^i\mathbf{1}\mathbf{1}^\top A^i, \tag{4.19}$$

and:

$$\frac{\overline{\Sigma}_{max}}{1-\rho^2}\mathbf{1}\mathbf{1}^\top = \overline{\Sigma}_{max}\sum_{i=0}^{\infty} A^i\mathbf{1}\mathbf{1}^\top A^i, \tag{4.20}$$

we can set the following boundaries:

$$\overline{\Sigma}_{min}\sum_{i=0}^{\infty} A^i\mathbf{1}\mathbf{1}^\top A^i \leq \sum_{i=0}^{\infty} A^i\overline{\Sigma}A^i \leq \overline{\Sigma}_{max}\sum_{i=0}^{\infty} A^i\mathbf{1}\mathbf{1}^\top A^i. \tag{4.21}$$

This way,

$$Osc\left(\sum_{i=0}^{\infty} A^i\overline{\Sigma}A^i\right) \leq \frac{\left(\overline{\Sigma}_{max}-\overline{\Sigma}_{min}\right)}{1-\rho^2}. \tag{4.22}$$

Therefore:

$$\begin{aligned} Osc(\mathcal{E}) &= Osc\left(\left(I-A^2\right)A\sum_{i=0}^{\infty} A^i\overline{\Sigma}A^i\right) \\ &\leq \rho\left(1+\rho^2\right)Osc\left(\sum_{i=0}^{\infty} A^i\overline{\Sigma}A^i\right) \\ &\leq \frac{\rho\left(1+\rho^2\right)}{1-\rho^2}\left(\overline{\Sigma}_{max}-\overline{\Sigma}_{min}\right) \\ &= \frac{\rho\left(1+\rho^2\right)}{1-\rho^2}Osc\left(Off\left(\overline{\Sigma}_x\right)\right). \end{aligned} \tag{4.23}$$

Now, with high probability,

$$Osc\left(\mathcal{E}\right) \leq \frac{A^+_{min}}{2}, \tag{4.24}$$

where A^{+}_{min} is the smallest non-zero entry of A. This means that a pair ij considered connected and kl disconnected, then $F_{ij} > F_{kl}$, which by definition 1 guarantees structural consistency. Therefore the structure can be recovered via thresholding the off-diagonal entries of F. Remark that the behaviour of the error matrix $\mathcal{E}$ is the *ruler* to whether the structure information is lost or not in the time-series. When this error matrix is flat enough the estimator F matrix entries are the shifted entries of A. Therefore we need to establish a caracterization of the error and study the *flatness* of the colored noise.

Consequently:

$$\begin{aligned} Osc(\mathcal{E}) &\leq \frac{\sigma^2_{gap} A^{+}_{min}}{2} \\ \frac{\rho\left(1+\rho^2\right)}{1-\rho^2} Osc\left(Off\left(\overline{\Sigma}_x\right)\right) &\leq \frac{\sigma^2_{gap} A^{+}_{min}}{2}. \end{aligned} \tag{4.25}$$

Finally, the next theorem can be defined.

Theorem 2. *Let $A = \rho\overline{A}$ be a stochastic matrix with $\rho \in]0,1[$, if*

$$\frac{Osc\left(Off\left(\Sigma_x\right)\right)}{\sigma^2_{gap}} \leq \frac{A^{+}_{min}\left(1-\rho^2\right)}{2\rho\left(\rho^2+1\right)}, \tag{4.26}$$

with $\sigma^2_{gap} \triangleq \sigma^2 - max_{i\neq j}\mathbb{E}\left[\mathbf{x}_i\mathbf{x}_j\right] > 0$ and A^{+}_{min} is the smallest nonzero entry of the interaction matrix A, then the centered features $\mathcal{F}_{ij}(n)_{i\neq j}$, $C\left(\mathcal{F}_{ij}(n)_{i\neq j}\right)$, are leanearly separable and stable.

4.1.4 Exogenous Intervention

If we add an exogenous intervention, ξ in the system:

$$\mathbf{y}(n+1) = A\mathbf{y}(n) + \mathbf{x}(n+1) + \xi(n+1), \tag{4.27}$$

where $\xi(n) \sim \mathcal{N}\left(0, \sigma^2_{\xi}\right)$ is i.i.d, and independent of $\mathbf{x}(n)$ the equation (4.26) becomes:

$$\frac{\mathsf{Osc}(\mathsf{Off}(\Sigma))}{\widetilde{\sigma}^2_x + \sigma^2_{\xi}} \leq \frac{A_{\min}(1-\rho^2)}{2\rho(1+\rho^2)}, \tag{4.28}$$

where $\widetilde{\sigma}^2_x := \sigma^2 - max_{i\neq j}\mathbb{E}\left[\mathbf{x}_i\mathbf{x}_j\right]$. This way, if the exogenous intervention is high enough, regardless of the covariance matrix of the input $\mathbf{x}(n)$ the features are linearly separable. The idea is to increase the diagonal characteristic of the noise structure making it less colored.

4.2 Methodology

In this segment, we elucidate the procedural progression employed in transitioning from the synthesis of artificial data and its corresponding time-series to the derivation of relevant features, culminating in the classification of the network's nodes into the categories of connected and disconnected entities.

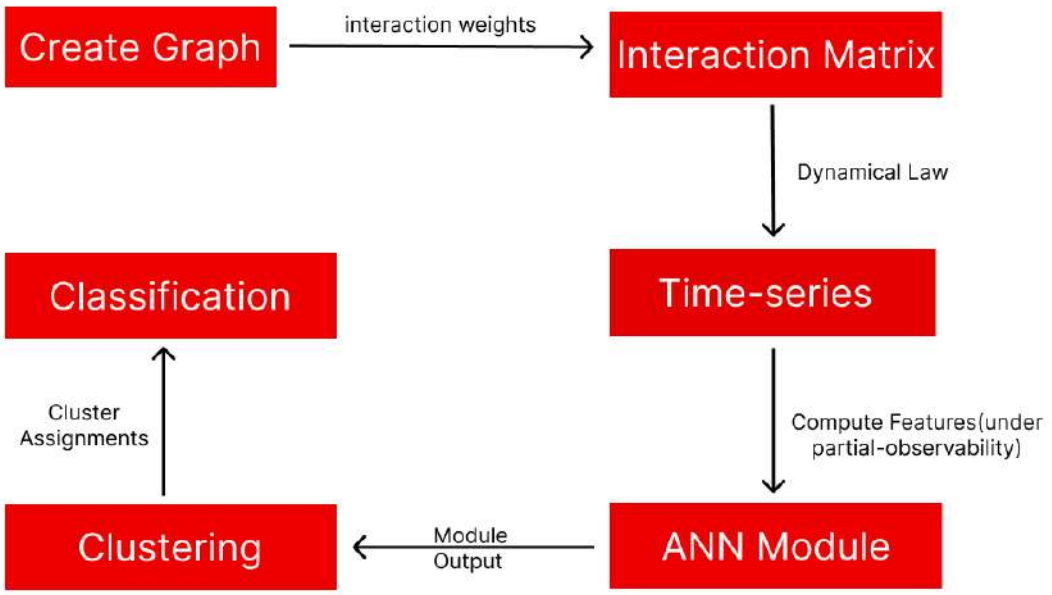


Figure 4.2: Summary of the scheme to obtain the synthetic time series data and perform classification.

In brief, our approach can be succinctly outlined through a sequence of six distinct steps. Initially, we establish a directed graph in accordance with specific rules, subsequently endowing its edges with weights, a process elaborated upon in Section 4.2.1. By employing a prescribed dynamical law delineated in Section 4.2.2, the ensuing time-series are generated. Subsequent to this, under conditions of partial observability, computation of the feature vector is effectuated, as expounded upon in Section 4.2.3. It is pivotal to note that this procedural trajectory remains consistent during both the training and testing phases, diverging thereafter.

During the training phase, the dataset is partitioned into discrete subsets for training and testing purposes. The Artificial Neural Network (ANN) models, pivotal to the numeric results elucidated in a subsequent Section, are then subjected to a training regimen. Conversely, during the testing phase, partial observability conditions are replicated. Employing the feature vector derived from observable nodes, clustering and subsequent classification procedures are executed on the module's output, attained by inputting the aforementioned feature vector.

4.2.1 Generate Graph and Adjacency matrix

The majority of conducted experiments involve the utilization of synthetic data to establish solid and controlled technical foundations for feature-based causal inference methods. Subsequently, these techniques are applied to actual data.

The Erdős-Rényi method, named after mathematicians Paul Erdős and Alfréd Rényi, constitutes a fundamental and widely employed approach within the realm of network theory and synthetic data generation. This method entails the construction of random graphs with a specified number of nodes and edges, thereby facilitating the emulation of various network structures for analytical and experimental purposes.

In the context of data synthesis, the Erdős-Rényi method serves as a mechanism to generate synthetic networks endowed with distinct topological characteristics. This is achieved by initially defining a set of nodes, followed by a stochastic process of edge allocation between these nodes. Specifically, each pair of nodes possesses a probability, p associated with the presence of an edge connecting them. This stochasticity in edge formation gives rise to a diverse array of network configurations, ranging from sparsely connected structures to densely interconnected networks. The parameters of node count and edge probability are amenable to variation, affording us the capacity to explore a wide spectrum of network configurations. This latitude in parameter manipulation facilitates the examination of an extensive array of distinct networks, thereby enhancing the breadth of our method's generalization and expanding its applicability.

The application of the Erdős-Rényi method within the generation of synthetic data offers a means to create controlled environments for testing and validating analytical algorithms, predictive models, and various network-oriented investigations. The resulting synthetic networks enable researchers to assess the performance of methods under varying network densities, connectivity patterns, and other structural attributes.

After building the structure of our graph we need to attribute weights on the edges resulting our interaction matrix A. We adopt a popular method often referred to as Laplacian rule [Sayed, 2014]. More concretely, the matrix A is defined as

$$\begin{cases} A_{ij} &= \alpha \frac{G_{ij}}{d_{\max}(G)}, & \text{for } i \neq j \\ A_{ii} &= \beta - \sum_{k \neq i} A_{ik}, & \text{for all } i \end{cases}, \tag{4.29}$$

where $d_{\max}(G)$ is the maximum degree of the underlying graph G, i.e., the maximum number of neighbors a node admits in the graph, and $0 < \alpha \leq \beta < 1$ are some parameters of the Laplacian model. In particular, the rows of A sum to $\beta < 1$ and its support is given by the generated graph G. As mention before, the interaction matrix A is stable, with a $\rho(A) < 1$ thanks to this rule.

4.2.2 Dynamical Law & Time-series

The principal objective of this thesis resides in the coherent inference of the latent architecture characterizing networked dynamical systems through analysis of their corresponding time-series. To accomplish this, the dynamical law governing the generation of said time-series is formulated with direct consideration of the graph structure established in the preceding stages defined as:

$$\mathbf{y}(n+1) = A\mathbf{y}(n) + \underbrace{\alpha \cdot x_1(n+1)}_{\mathcal{N}_1} + \underbrace{\frac{\beta}{\sqrt{N}} \cdot 1 \cdot 1^\top \cdot x_2(n+1)}_{\mathcal{N}_2}, \tag{4.30}$$

where $\mathcal{N}_1$ is the diagonal component of the noise with covariance matrix $\alpha^2 I > 0$, and $\mathcal{N}_2$ is the colored component of the noise with standard deviation β. In the numeric results Chapter 5 we explore the influence of the noise variation by changing β in the proposed method's performance.

4.2.3 Features

The set of features computed from the time-series characterized with colored noise can be defined as the Cartesian product:

$$\mathcal{K}^M(n) \triangleq \mathcal{T}^M(n) \times \mathcal{F}^M(n). \tag{4.31}$$

Specifically, for each pair ij:

$$\mathcal{K}_{ij}^M(n) = \left(\mathcal{F}_{ij}^M(n), \mathcal{T}_{ij}^M(n)\right). \tag{4.32}$$

Further, $\mathcal{T}^M(n) = \left\{\mathcal{T}_{ij}^M(n)\right\}_{ij}$

$$\mathcal{T}_{ij}^M(n) = \left(\left[\left(\left[\widehat{R}_0(n)\right]_S \right)^{-1} \right]_{ij}, \ldots, \left[\left(\left[\widehat{R}_M(n)\right]_S \right)^{-1} \right]_{ij} \right),$$

The ensemble of features under consideration constitutes the inversion of the correlation lag moments inherent to the complementary portion of the feature set denoted as $\mathcal{F}^M(n)$. The outcomes derived from the incorporation of this particular set of inverted correlation matrices substantiate the assertion that the aforementioned features enhance the distinctiveness of the dataset. This enhancement is reflected in the discernible gap, referred to as the "identifiable gap," between the feature representations of interconnected and disconnected pairs. Notably, this gap is more pronounced within the feature set $K^M(n)$ as compared to the separate subsets, namely $\mathcal{T}^M(n)$ and $\mathcal{F}^M(n)$, as formally expounded in Lemma 2 within the reference [Machado et al., 2023]. This particular attribute endows the novel feature ensemble with paramount suitability for the application of supervised methodologies in the context of clustering objectives.

As delineated within the technical findings segment, the featured attributes put forth in this study maintain qualities of linear separability and structural integrity even under variations in the noise profile, as defined by the dynamical law denoted as equation (4.30). More precisely, these attributes exhibit resilience when subjected to minor perturbations in the off-diagonal elements of the correlation matrix Σ, as elucidated in Theorem 2, or in scenarios where exogenous interventions attain a certain threshold, as expressed in inequality (4.28). Characterized by these attributes, the Feed Forward Neural Networks (FFNN) trained utilizing these features, post normalization with the "Standard Scaler" technique, yield competitive performance metrics vis-à-vis alternative estimation methodologies.

4.2.4 Features's Normalization

Adding the colored noise factor to the equation the features used to describe the dynamical system suffer a shift in the features space. In order to counter this shift, the features are standardized with Standard Scaler. The StandardScaler is a data preprocessing technique commonly used in machine learning and data analysis. It belongs to the category of feature scaling methods, which are employed to normalize the features of a dataset, ensuring they all have comparable scales.

The primary objective of the StandardScaler is to standardize the features such that each feature's distribution has a mean of zero and a standard deviation of one. The standardization process involves two main steps: mean centering, for each feature (column) in the dataset, the mean value (μ) is calculated. Subsequently, the mean value is subtracted from each data point in that feature, resulting in the feature having a mean of zero; scaling: following mean centering, the feature is scaled by dividing each data point by the standard deviation (σ) of that feature. This operation ensures that the feature has a standard deviation of one.

That way the standard score of a sample x is calculated as:

$$Z = \frac{x - \mu}{\sigma}, \tag{4.33}$$

where Z is the standardized value of the sample x, μ is the mean and σ is the standard deviation of the feature vector.

Overall, feature scaling is a crucial preprocessing step that significantly impacts the performance of machine learning models. The StandardScaler is a widely adopted tool in achieving this goal, ensuring that the features' scales are harmonized and conducive to effective model training and prediction.

4.2.5 Training Structure

In machine learning we have different types of models that differ in structure and in strengths and weaknesses considering the final goal that we want to achieve. However, in the training phase all models need a training set and a testing set. A

validation set can be useful too and is used in most cases.In our case the three are present in the training phase.

The synthetic data used in this project have some parameters in play such as: number of features, probability of the connectivity between the network's nodes, the number of nodes belonging to the network, the noise that we want to apply to our scenario, and some other hyper parameters relevant to the results but the time needed to explore that is way more higher than the time budget to deliver this project.

In the proposed approach we train the Feed-Forward Neural Network (FFNN) models using a dataset with an underlying structure of 100 nodes. This dataset has several subsets that differ on the noise applied to them and in the number of samples. This way we can have a more robust model that has a good performance overall. Having the dynamical law (4.30), the dataset used in the results illustrated in Fig. 5.19 to Fig. 5.24 is computed from time-series differing in the β term, being $\beta \in [0, 10, 20, \ldots, 50]$.

It is imperative to underscore that both the training and testing phases impose a substantial computational burden. Notably, circumstances arise wherein the exploration of factors such as the number of samples, the frequency of runs, the count of nodes, and similar parameters, could have been pursued more extensively with access to more advanced hardware resources. Regrettably, due to temporal constraints associated with the available resources, the exploration of these elevated values and intriguing scenarios remained beyond the scope of our investigation.

4.2.6 Clustering

In some experiments we resort to clustering algorithms to help with the partitioning of the features associated with connected and disconnected pairs of nodes.

The objective of clustering is to group a set of data in a way that the data belonging to same group also called cluster are more similar to each than to those in distinct groups. In our scenario we want to separate the agents that are connected and the ones that are disconnected. Cluster analysis itself is not a specific algorithm. We chose to test k-means algorithm and Gaussian Mixture.

4.2.7 Classification

After the clustering applied to the output given by the FFNN model we consider one centroid as the connected centroid and the other as the disconnected centroid. The nodes belonging to them are then classified by the corresponding centroid that they belong to.

Chapter 5

Numeric Simulations

This Chapter contains a comprehensive collection of experiments associated with our feature based method. They are conducted on linear NDS with underlying direct graphs. Some numerical experiments have been submitted for publication [Santos et al., 2023].

5.1 Mutual-information based formulation

This experiment tried to elucidate the information contained in the features of the work [Machado et al., 2023] about the structure underlying the network system. With it we wanted to build solid ground on the number of features needed in order for the model of the referred paper obtain a competitive performance.

Briefly, Mutual Information is a fundamental concept in information theory that quantifies the degree of dependence or shared information between two random variables. It measures the reduction in uncertainty about one variable when the value of the other variable is known. In essence, mutual information gauges how much knowledge of one variable helps predict the other.

Mathematically, the mutual information between two discrete random variables X and Y is calculated by considering the probabilities of their joint occurrences and individual occurrences. It is defined as the difference between the entropy of X and the conditional entropy of X given Y:

$$I(X;Y) = H(X) - H(X|Y). \tag{5.1}$$

Here, $H(X)$ represents the entropy of variable X, which quantifies its intrinsic uncertainty. $H(X|Y)$ represents the conditional entropy of X given Y, indicating the remaining uncertainty about X when Y is known.

In Fig. 5.1 the mutual information between the ground-truth interaction matrix A and the Granger estimated interaction matrix, $\widehat{A}$, and the correlation lag moments developed in [Machado et al., 2023], $\widehat{R}_k(n)$.

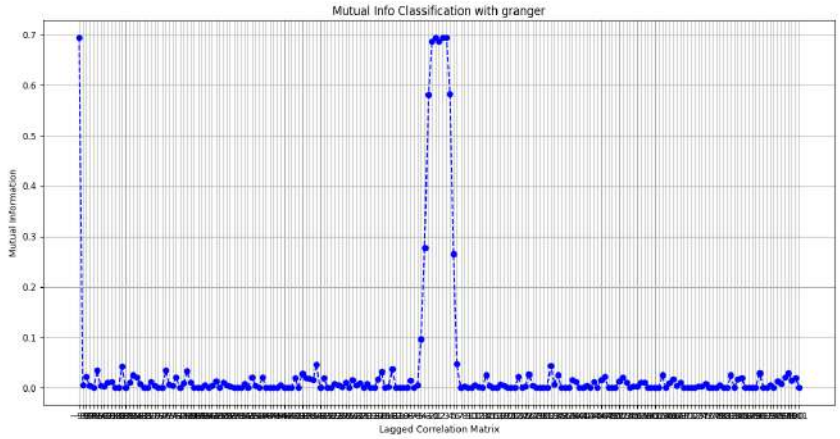


Figure 5.1: Mutual information is maximal at the Granger estimator and low-lag estimators.

The initial datum on the abscissa corresponds to the Granger estimator, while the subsequent entries pertain to the correlation lag moments. Notably, a prior observation in [Machado et al., 2023] elucidated that the Support Vector Machine (SVM) applied in experimentation exhibited elevated weights predominantly for lower lag moments, and conversely, lower weights for higher lag moments. The outcome depicted in Fig. 5.1 aligns harmoniously with this antecedent finding. However, in the context of employing Feed Forward Neural Networks (FFNNs), the weight distribution across the spectrum of correlation lag moments is found to be more evenly divided.

Next we do the same test but with a random generated interaction matrix A_r. This way we want to verify that the mutual information is a metric that is consistent in this type of framework.

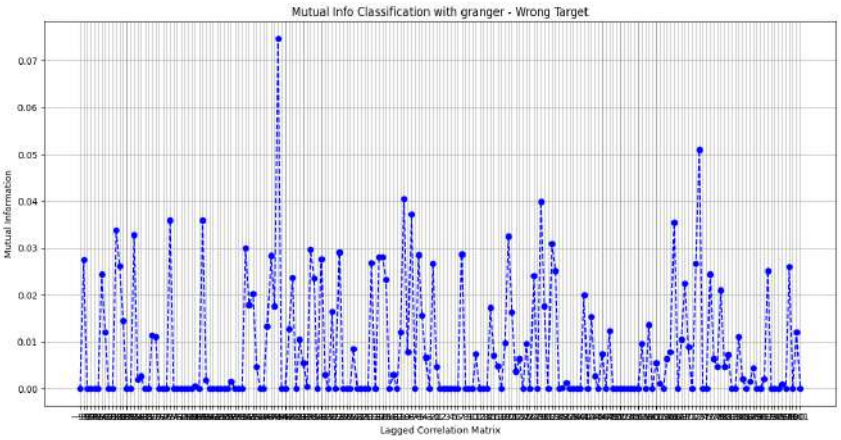


Figure 5.2: Mutual information is low when the underlying structure does not match the correct one.

From the observation of the Fig. 5.2, the values of the mutual information are way lower than the values obtained in Fig. 5.1.

What happens if we change the network structure? Will the low lag moments still get the highest mutual information values? This questions are the reason of the following picture 5.3.

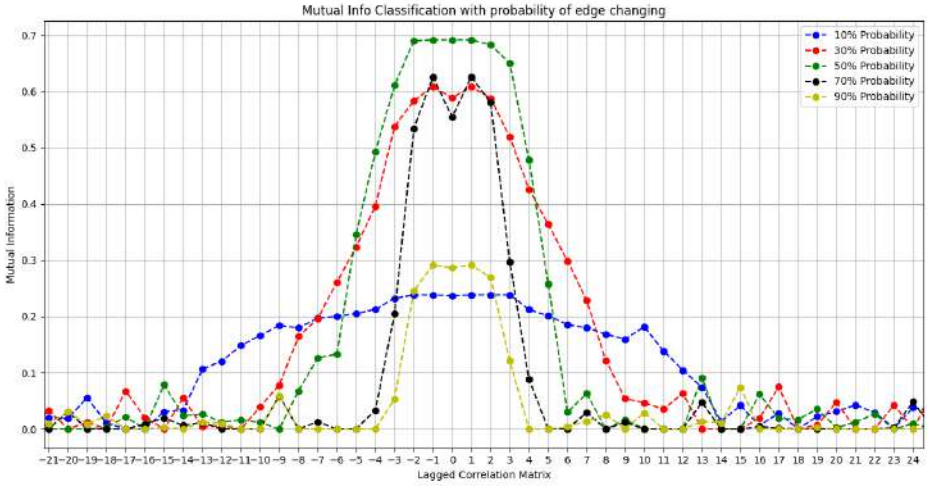


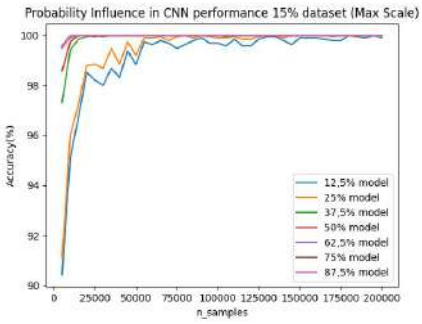
Figure 5.3: Mutual information exhibits wider base for sparser networks. Higher order lag-moments play a role in the estimation.

As expounded upon in Section 4.2.1, the manipulation of network connectivity is achievable through the manipulation of the Erdős-Rényi algorithm's probability parameter denoted as p. The tested values of this parameter are presented in the legend located at the upper corner of the figure. A discernible trend emerges from this investigation: as the probability p increases, the foundational extent of the mutual information tends to contract, a phenomenon attributable to the fact that sparser networks (characterized by lower p values) yield broader foundational distributions. This deduction gains further reinforcement through the analysis conducted in the experiment detailed in Section 5.2. This Section has results that were obtained in cooperation with another student with similar thesis (Diogo Rente).

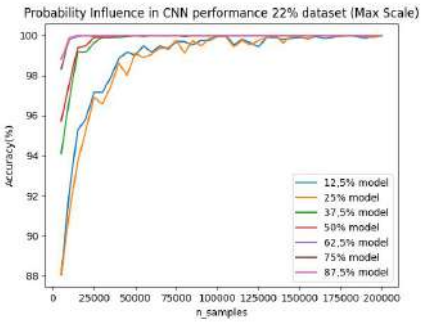
5.2 Regime Analysis

In order to better understand the old model's (Convolutional Neural Network (CNN)) behaviour along the different types of networks and the interference of our features array size in it, we tested, firstly, what would result if we change the probability of connectivity between the nodes of the network in the training phase. Then, run across different probability values in the testing phase with the aim of verifying the difference in the results. After that, we lock the probability and explore the number of features influence in the CNN model response.

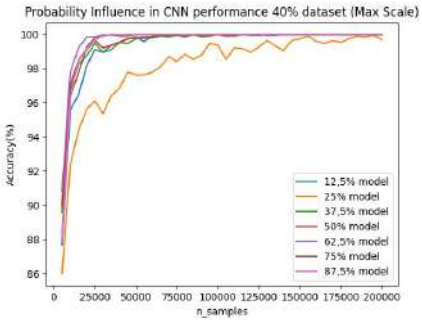
5.2.1 Probability Variation



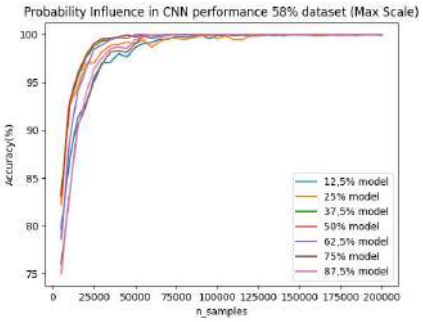
(a) 15% Probability test



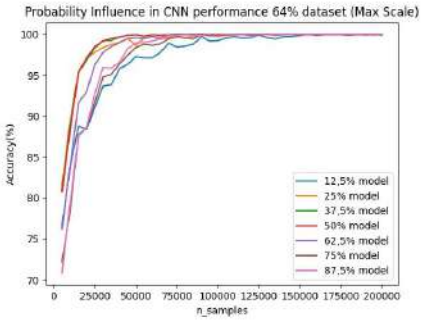
(b) 22% Probability test



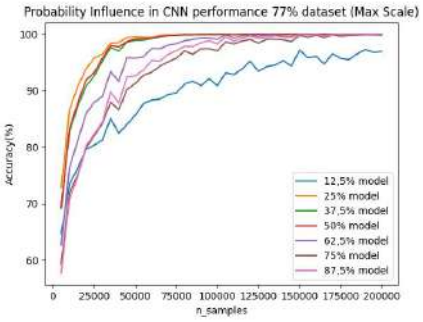
(c) 40% Probability test



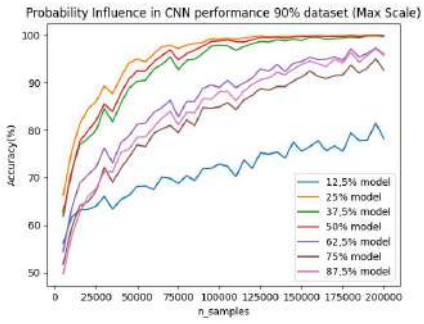
(d) 58% Probability test



(e) 64% Probability test



(f) 77% Probability test



(g) 90% Probability test

Figure 5.4: Probability Test

All the lines represent the accuracy of a CNN model which is trained with the probability of the network's connectivity labeled with the corresponding color. This tests were made with only diagonal noise presented in the time-series. When thinking about this experiment, we expected that the models trained in sparser networks would get better results when testing with sparse networks versus the models trained with dense networks. For example, the models trained with 12.5%, 25%,37.5% got a lower performance comparing to the 87.5% and 62.5%, in figures (a), (b) and (c). And in the dense case (fig (e),(f) and (g)) the top three models with the best performance are models trained with sparse networks which goes against our initial thought. The closest result that defends our speculation is (d) where the best model is the one trained with 50% connectivity of the network, similar to the network tested (58%).

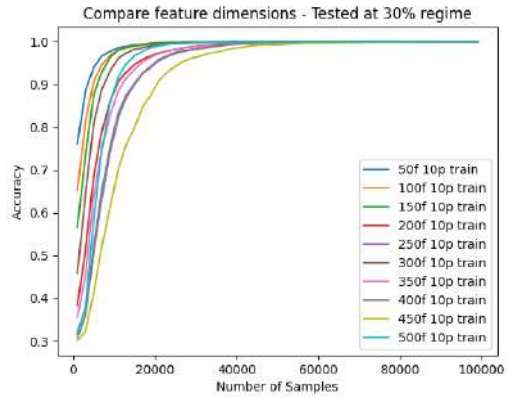
Therefore, there is not a trivial conclusion that we can take from this tests, only that the probability of the network's nodes being connected does affect the behaviour of CNN models and there is not one that can generalize well enough.

5.2.2 Influence of the Feature Vector

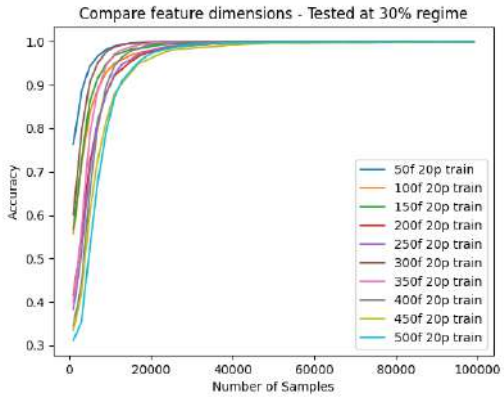
A significant facet in the architecture of CNNs lies in the number of features they consider during their operation. This parameter, often referred to as the "feature count," directly influences the network's capacity to discern intricate details and generalize from the input data.

In this Section, we embark on a comprehensive exploration of the influence that the number of features considered holds within the realm of CNNs. By systematically varying the feature count and meticulously observing its effects on network performance, we gain profound insights into the interplay between model complexity, computational efficiency, and predictive prowess. By demystifying the intricate relationship between feature count and CNN performance, this exploration contributes to the broader understanding of CNN architecture design, enabling practitioners to make informed decisions when tailoring models for specific tasks.

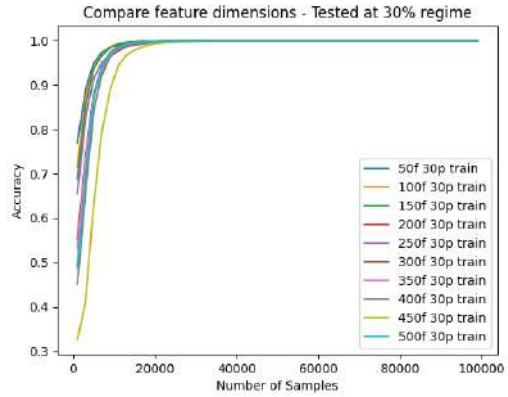
In this experiment we consider two regimes in the testing phase, a sparse network and a dense one being the connectivity of the sparser one 30% and 70% for the dense case. The models used will variate in the network's connectivity and the number of features used in the training phase.



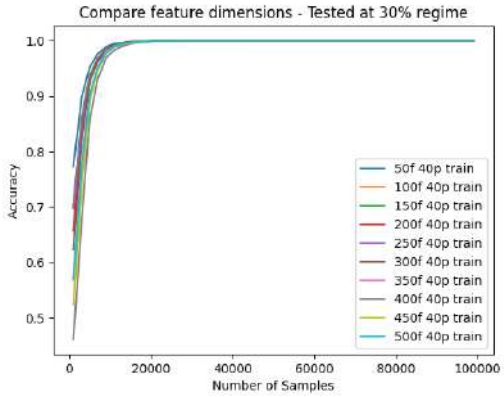
(a) Trained at 10%



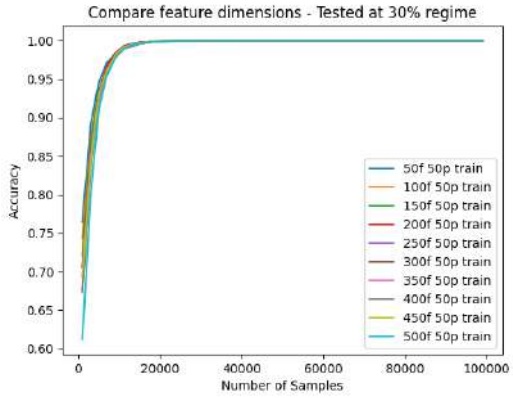
(b) Trained at 20%



(c) Trained at 30%

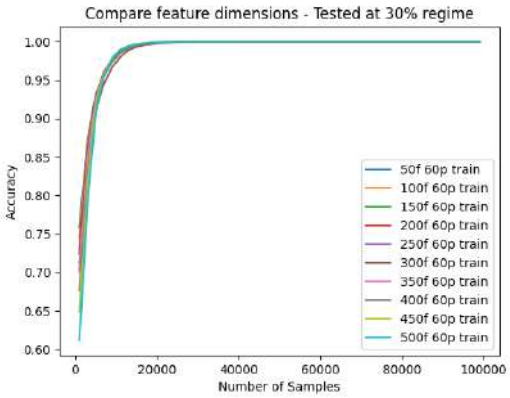


(d) Trained at 40%

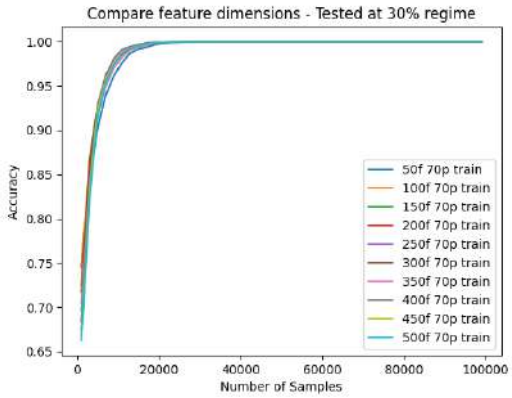


(e) Trained at 50%

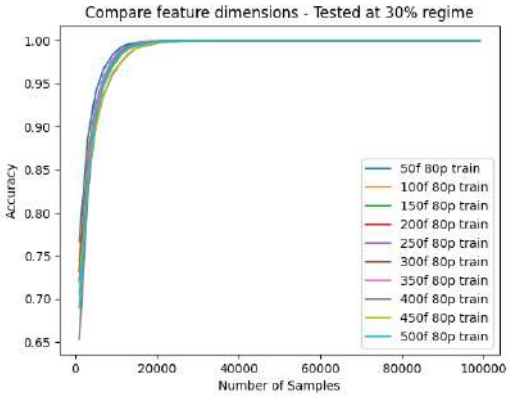
Figure 5.5: Feature Influence - 30% test - sparse networks



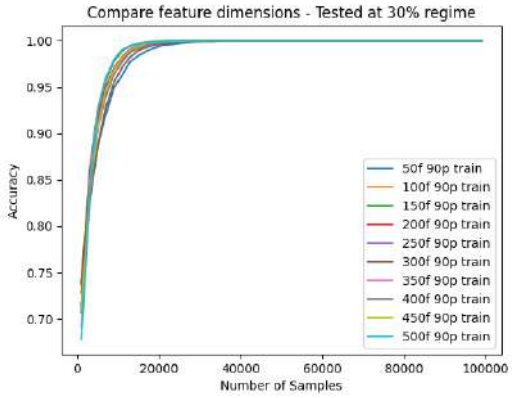
(a) Trained at 60%



(b) Trained at 70%



(c) Trained at 80%

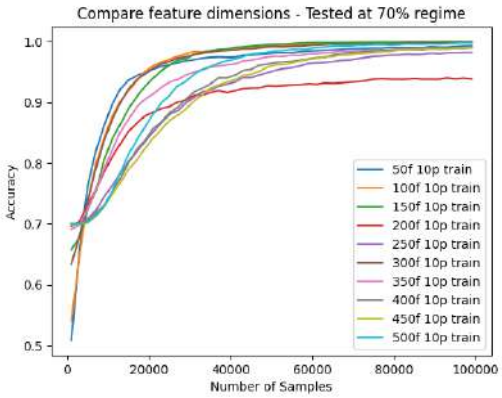


(d) Trained at 90%

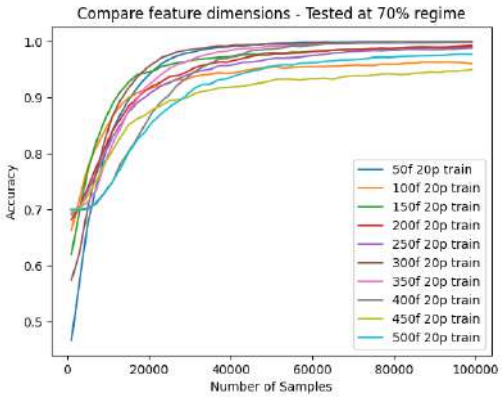
Figure 5.6: Feature Influence - 30% test - dense networks

The expected result having in consideration the results in 5.1 was a gradual increase in the number of features needed to get the best model's performance because of the training phase increasing connectivity.

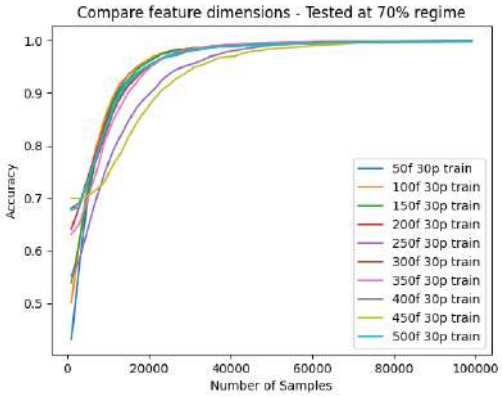
Instead, the number of features required to get the best performance through out the training connectivities is not trivial. At 10%, 20%, 40%, 80% the best performance was obtained with 50 features and in the rest of the cases the number of features that lead to a better performance varies.



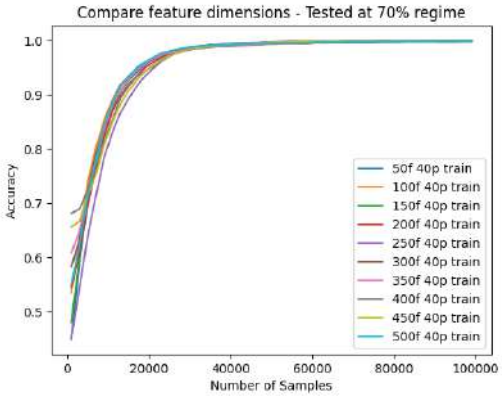
(a) Trained at 10%



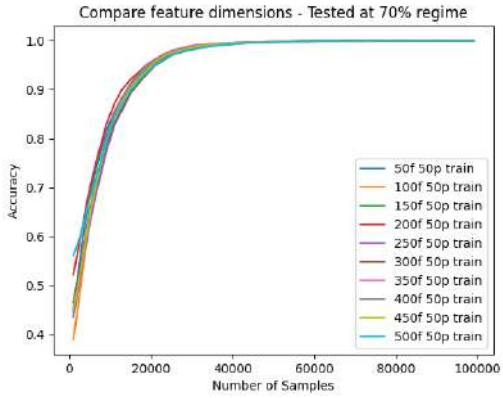
(b) Trained at 20%



(c) Trained at 30%

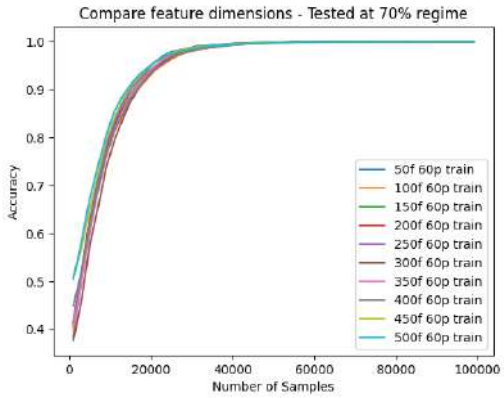


(d) Trained at 40%

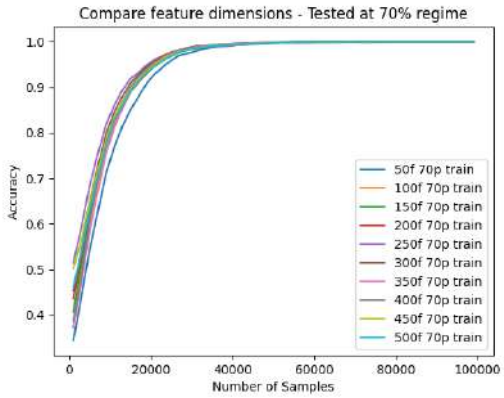


(e) Trained at 50%

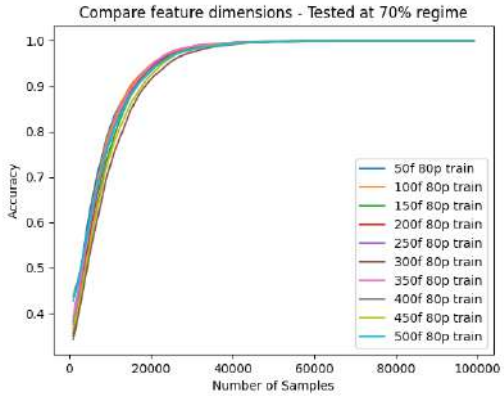
Figure 5.7: Feature Influence - 70% test - sparse networks



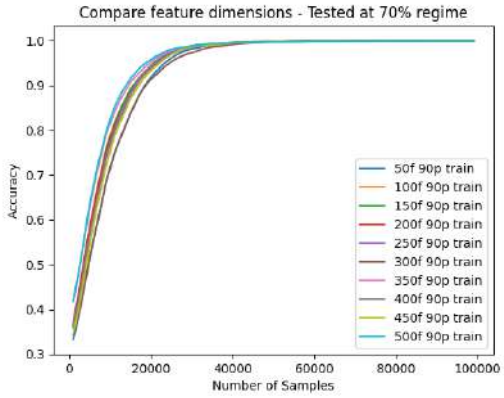
(a) Trained at 60%



(b) Trained at 70%



(c) Trained at 80%



(d) Trained at 90%

Figure 5.8: Feature Influence - 70% test - dense networks

In this case, a denser one than in Fig.5.5 and Fig.5.6, there is not a constant number of features that provides the best performance along the different training probabilities of the network's connectivity and are different from the results obtained in the 30% test(5.5 and 5.6).

Concluding, there is a influence in the performance of the model but from a certain value of the networks's connectivity the influence is not that critical.

5.2.3 Tree Experiment

In this Section we take the two experiments made in 5.2.1 and 5.2.2 to build a type of tree framework where we take the model trained at 50% connectivity to be the root of the tree, and its two leafs are the models trained at 25% and 62,5% connectivities. The left leaf will be the model trained at 62,5% and the right one is the model trained at 25%. The models are chosen due to their performance in the probability variation test, where, in general, the network being a sparser one the 62,5% model as a good performance and if the network is a denser one, the 25% model as a good performance too.

So based on the output of the tree's root the leaf is chosen. If the output reveals a network with a connectivity lower than 50% the left leaf is chosen and used to classify the network's nodes connected or disconnected. The same classification procedure is done at the right leaf, which is chosen if the root's output reveals a denser network (connectivity above 50%). The goal was to build a pipeline that could have good performances independently of the network's connectivity.

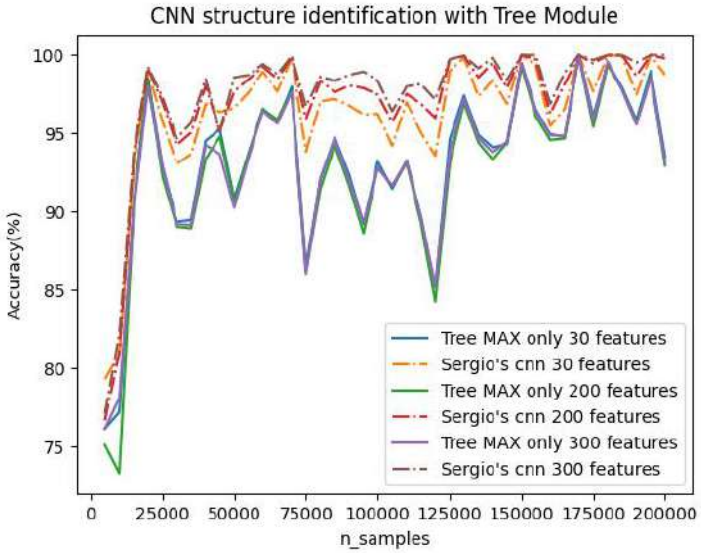


Figure 5.9: Tree classification results

In 5.9 are the results of three trees and three solo models that are the three's roots. They differ from each other in the number of features used in the training phase. The signature "Tree max" refers to the normalization done to the features, and that is the division by the feature's max value used in the CNN model proposed in [Machado, 2022].

Observing the results, the solo models have a better performance comparing to the Three pipeline built. So our approach to build a stronger pipeline through the regimes of the network failed. The model resorts to much in the root's decision and it can only get good performance after the decision is 100% accurate.

5.3 Colored noise on the CNN

Prior to exploring alternative artificial neural network (ANN) modules and structures, it is imperative to conduct a preliminary assessment to ascertain the consistent performance of the convolutional neural network (CNN) module presented in the reference [Machado et al., 2023]. This assessment is aimed at evaluating the module's ability to effectively discern the underlying structure of a noisy dataset (NDS), particularly when the noise structure includes non-zero values in its off-diagonal elements, which is indicative of colored noise.

To initiate this assessment, we commence by analyzing the classification results

obtained from a dataset in which the parameter denoted as β is set to zero ($\beta = 0$). In this configuration, the absence of colored noise is ensured, and only the diagonal elements of the noise structure possess non-zero values.

Subsequently, we proceed to increment the β noise parameter to modest levels, such as 0.1 and 0.5. Throughout this process, it is worth noting that all datasets undergo a maximum normalization procedure, consistent with the methodology outlined in the aforementioned reference [Machado et al., 2023].

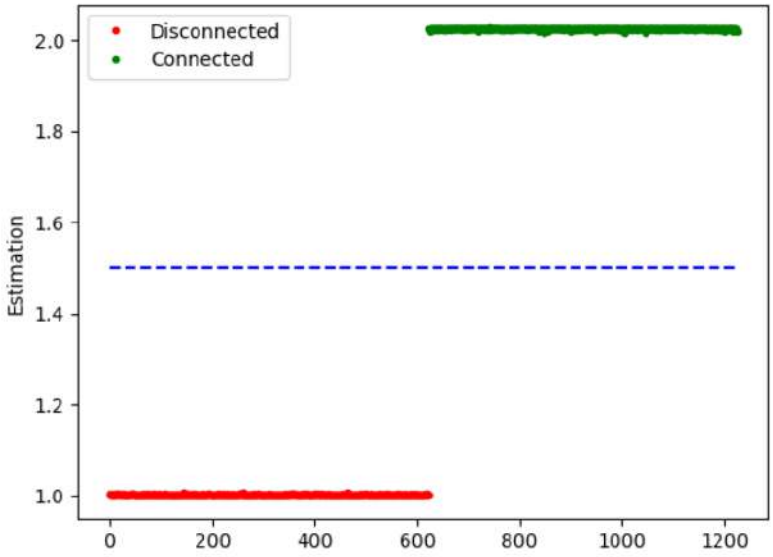
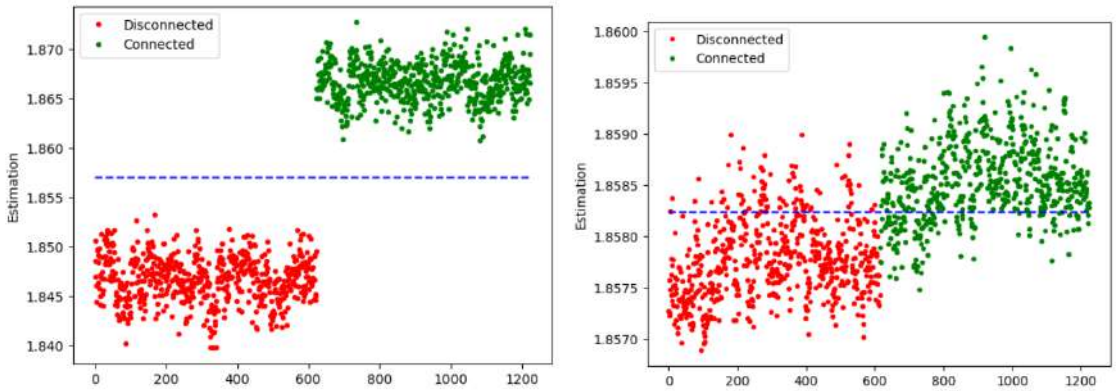


Figure 5.10: Data classification with diagonal noise



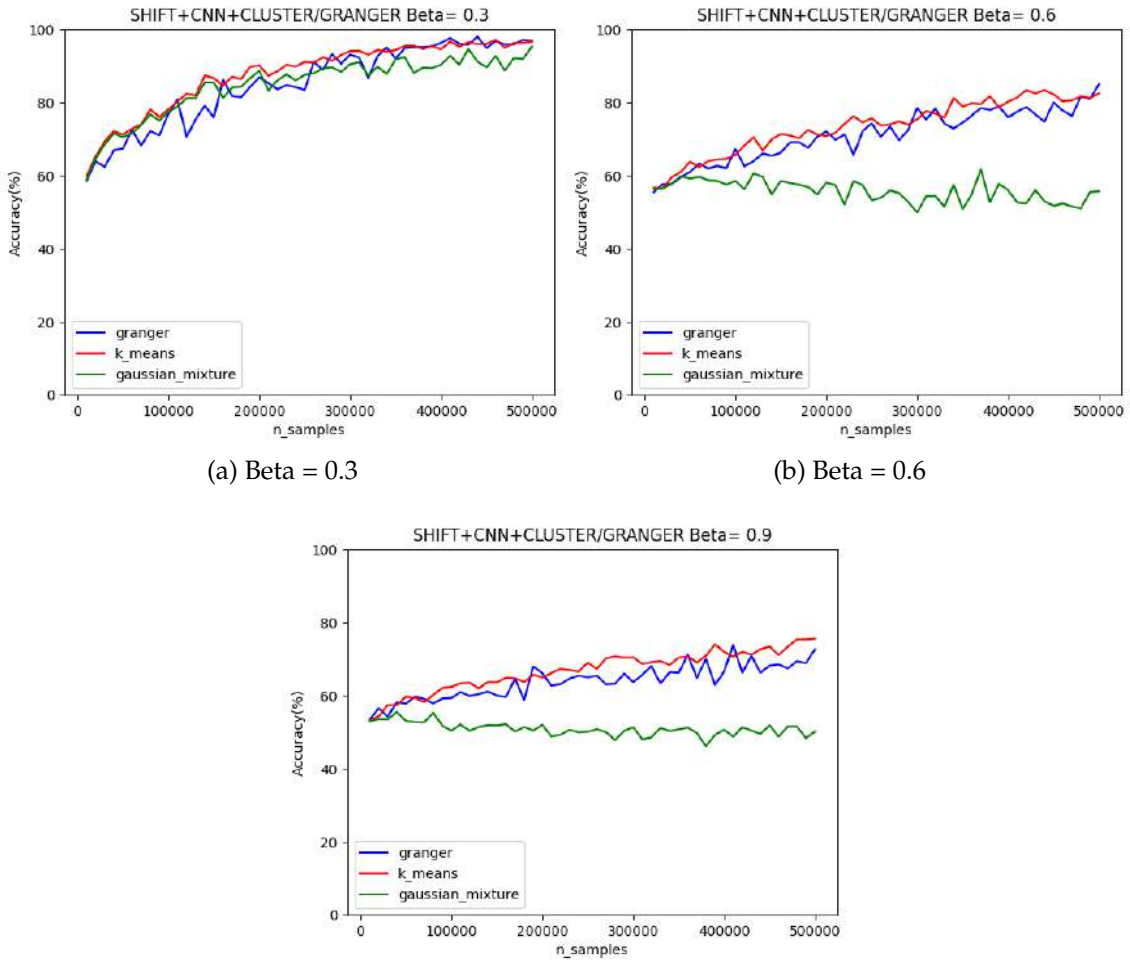
(a) $\beta = 0.1$ (b) $\beta = 0.5$

Figure 5.11: Data classification with colored noise

Upon careful examination of the obtained results, it becomes evident that the estimation of link strength pertaining to both connected and disconnected pairs of nodes begins to converge, exhibiting a decreasing trend in the value of the Identifiability Gap (IdGap) metric [Matta et al., 2022]. This convergence is particularly notable as the magnitude of the colored noise factor is progressively augmented.

The post-data handling strategy, as detailed in [Machado et al., 2023], can be succinctly summarized as a thresholding mechanism applied to the output produced by the Convolutional Neural Network (CNN) module. Specifically, pairs of nodes falling above this threshold are classified as linked, while those falling below it are categorized as disconnected. However, the observed behavior presents a challenge for this thresholding technique. For instance, in the illustrative example shown in Fig 5.11 where $\beta = 0.5$, it becomes apparent that the connected and disconnected pairs can no longer be effectively distinguished. Consequently, in response to this challenge, our subsequent approach involved the application of clustering methods, as referenced in Section 4.2.6, to the output generated by the CNN module. We employ the Granger causality analysis as a baseline reference.

The model used was trained with a balanced network, i.e., the number of connected pairs is equal to the number of disconnected pairs, being the probability parameter p in the Erdős-Rényi graph construction algorithm equal to $0.5(50\%)$.



(a) Beta = 0.3

(b) Beta = 0.6

(c) Beta = 0.9

Figure 5.12: Results of clustering the CNN output

One notable observation derived from the results is that, as the colored noise factor is elevated, there is a corresponding increase in the requisite number of samples needed to attain a satisfactory level of classification accuracy for discerning

the underlying functional connectivity of the system.

One notable observation derived from the results is that, as the colored noise factor is elevated, there is a corresponding increase in the requisite number of samples needed to attain a satisfactory level of classification accuracy for discerning the underlying functional connectivity of the system.

5.4 New features

In order to simplify and get better performances, we came together with some new features. In Section 5.4.1 the link between each pair of nodes is described with the value estimated by the granger estimator, with the one-lag-correlation matrix(R_1) and its inverse(R_1^{-1}), with the three-lag-correlation matrix(R_3) and, finally, with the zero-lag-correlation matrix inverted(R_0^{-1}). This selection of features was based in some results along the innumerous tests made on behalf of the colored noise regime. In which we observed that these seven variables could be useful or would help in classifying the link between the two nodes. Section 5.4.2 is the result of the observation of the output given through the experiment made in Section 5.4.1. Remark that this thesis focus on directed graphs.

As explained in 4.1.1, the features suffer a deviation or a *drift* when colored noise it's considered. In order to counter this behaviour, the features are normalized by a *Standard Scaler*.

The structural approach involving the extraction of features followed by the application of standard scaling to these features was found to be incompatible with the architecture of the Convolutional Neural Network (CNN) module as detailed in [Machado et al., 2023], i.e., the CNN has poor performance. The underlying reason for this incompatibility remains somewhat elusive. In lieu of adjusting the hyperparameters of the CNN model's structure to address this issue, we opted to explore Feedforward Neural Network (FFNN) models that incorporate the newly devised structural framework and feature set.

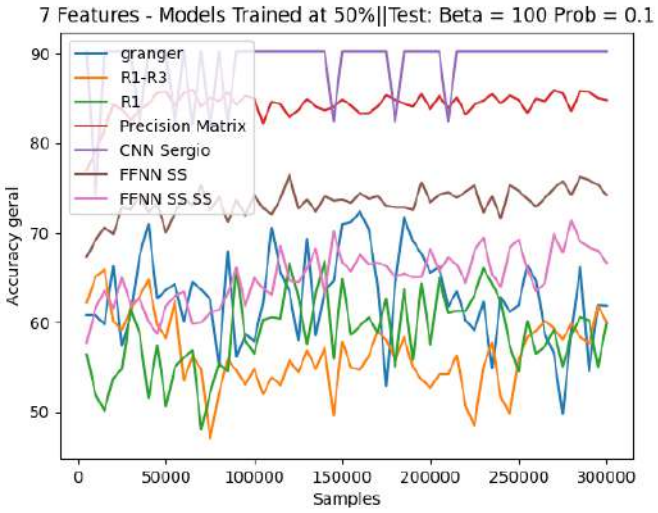
5.4.1 7 features

Instead of discarding right at the start the estimators that we could find in the literature, we built a feature array that include them rather then just our correlation lag moments. In total, the relation between two nodes is described by 7 features, its estimated value calculated with granger, the zero, one and three lag correlation matrices and the inverted matrices of the same lag moments.

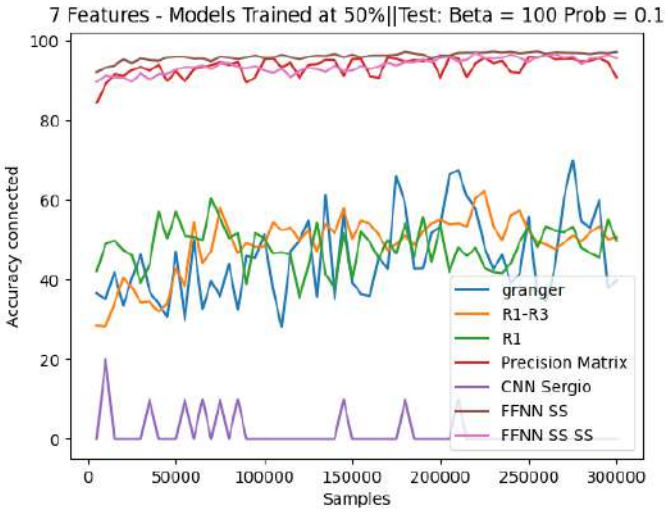
$$\mathcal{F}_{ij}^{(n)} = \left(G_{ij}, [R_0]_{ij}, \left[R_0^{-1}\right]_{ij}, [R_1]_{ij}, \left[R_1^{-1}\right]_{ij}, [R_3]_{ij}, \left[R_3^{-1}\right]_{ij} \right)^{(n)}, \tag{5.2}$$

Being G the value estimated by the granger estimator, and R_k the k lag moments. ij referring to the link between node j and i and (n) to the n instant in time.

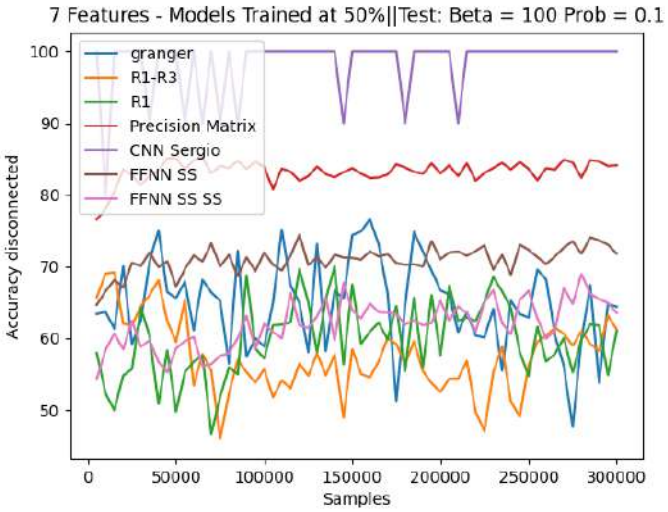
Note that this tests were obtained with few runs. That is, the number of cycles we use to get the performance is low in order to run the test a little bit faster. This way the curves of the performances will be less smooth. It only justifies the increase of number of runs if the outcome is interesting to the goal we are trying to reach.



(a) 10% Geral

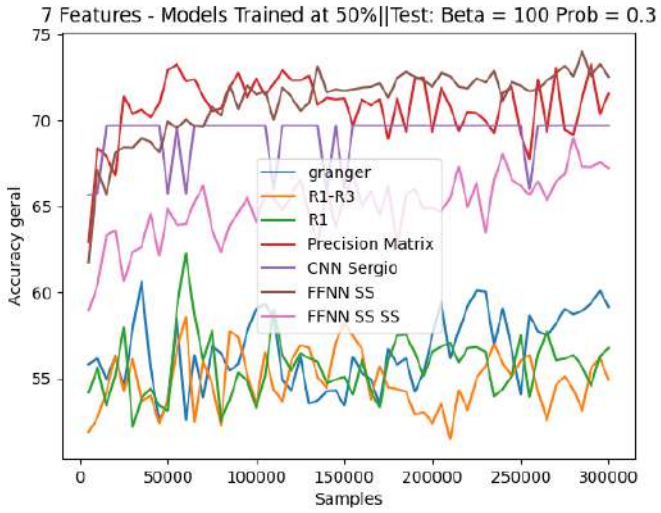


(b) 10% Connected

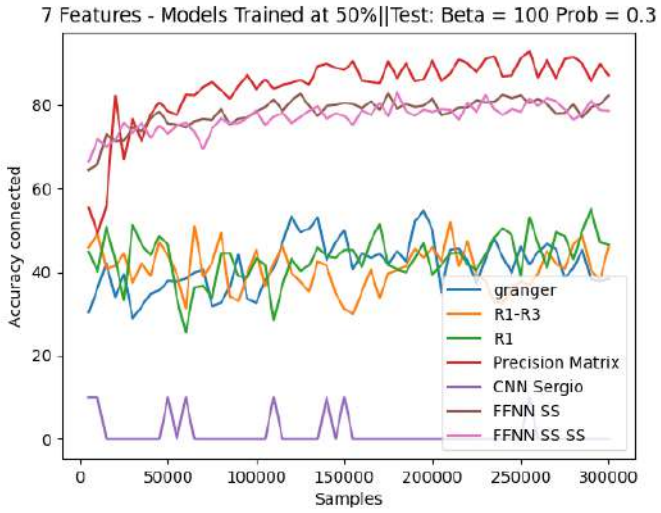


(c) 10% Disconnected

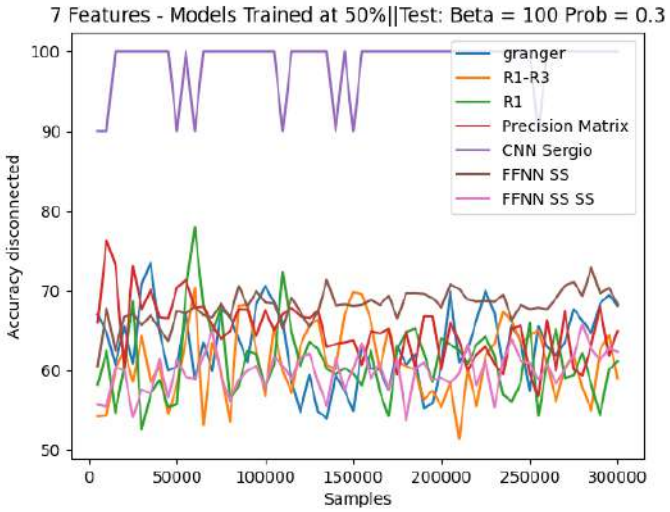
Figure 5.13: 7 features Results - 10% test



(a) 30% Geral

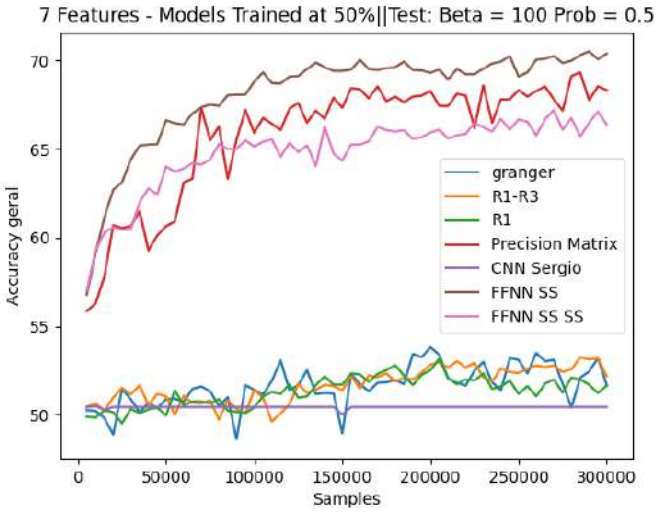


(b) 30% Connected

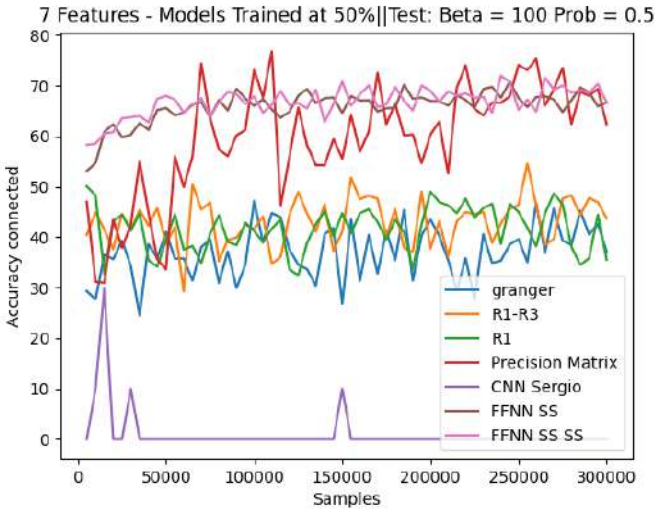


(c) 30% Disconnected

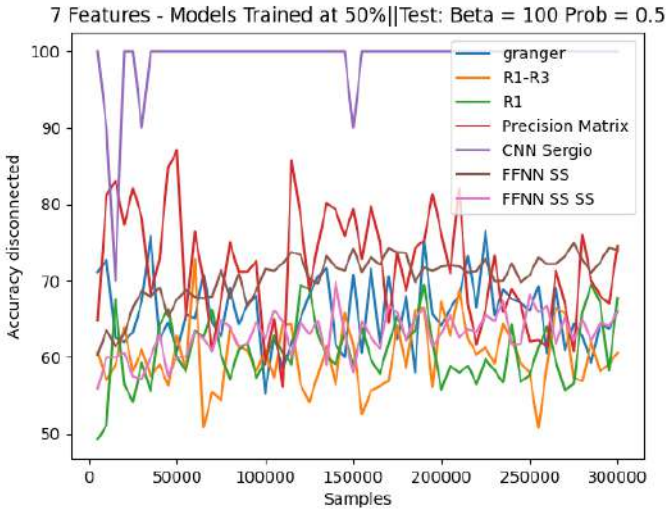
Figure 5.14: 7 features Results - 30% test



(a) 50% Geral

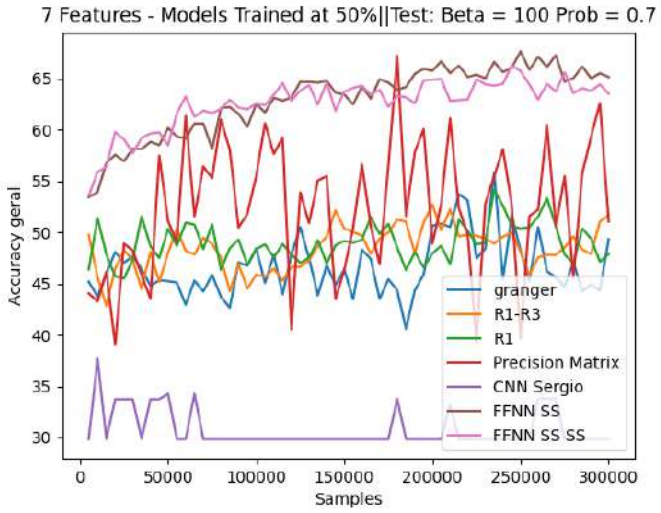


(b) 50% Connected

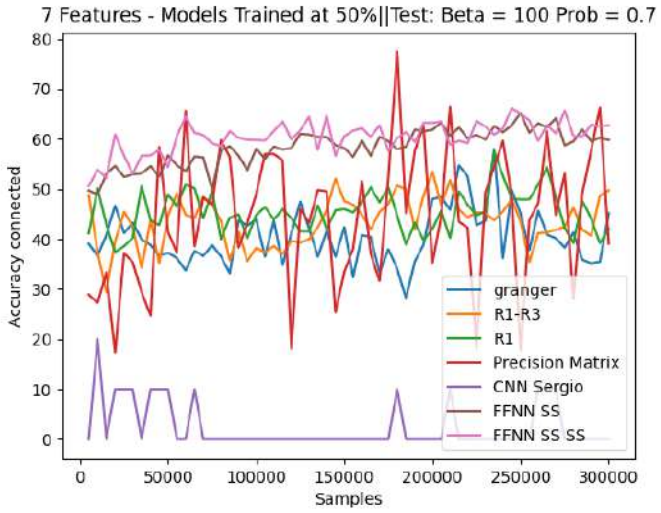


(c) 50% Disconnected

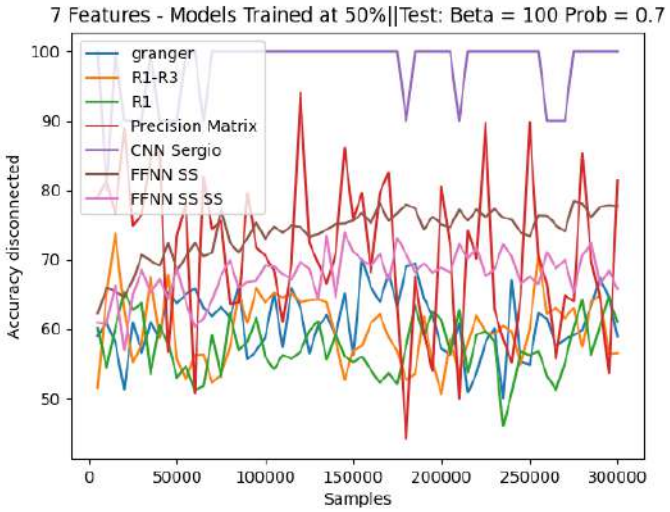
Figure 5.15: 7 features Results - 50% test



(a) 70% Geral

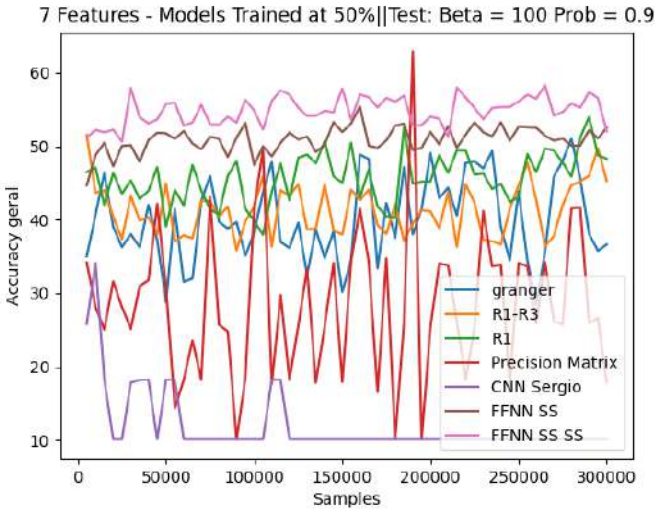


(b) 70% Connected

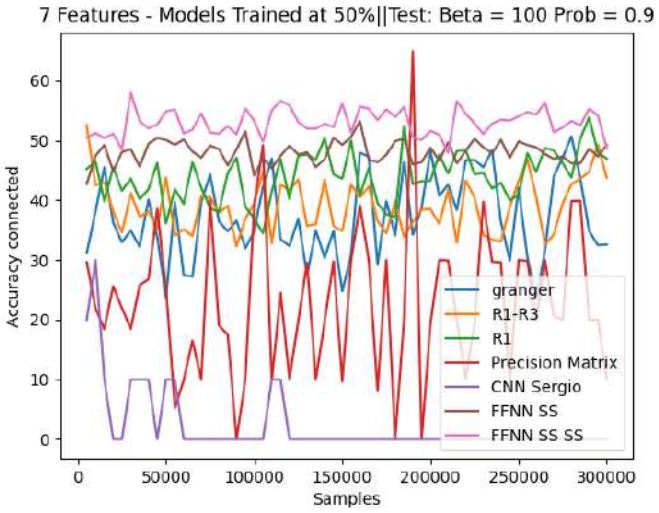


(c) 70% Disconnected

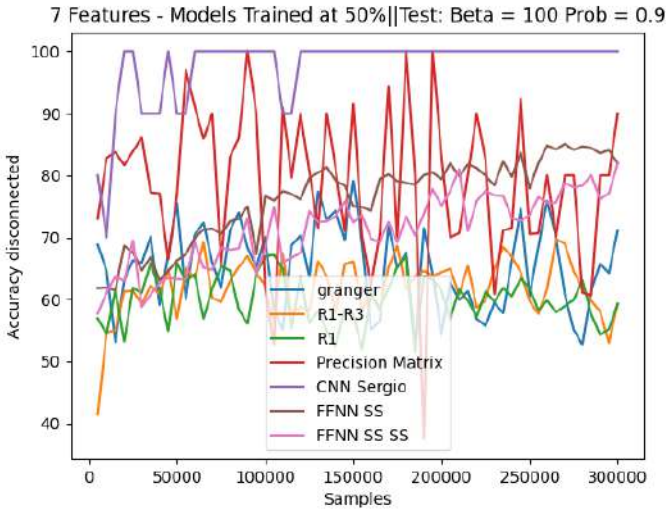
Figure 5.16: 7 features Results - 70% test



(a) 90% Geral



(b) 90% Connected



(c) 90% Disconnected

Figure 5.17: 7 features Results - 90% test

All models have been trained under a connection probability regime of 50% and the network directed. The examination encompasses varying connectivity probabilities, as designated in the accompanying figures. The evaluation entails a comparative analysis of performance involving four estimators: *Granger*,$R_1 - R_3$, R_1 and *Precision Matrix* against the backdrop of the legacy CNN model (referred to as *CNN Sergio*), as well as our two recently developed Feedforward Neural Network (FFNN) models.

For the legacy CNN, the feature set comprises two hundred correlation lag matrices, encompassing the initial one hundred negative lags and the subsequent one hundred positive lags. Conversely, the FFNN models utilize the previously elucidated array of seven new features.

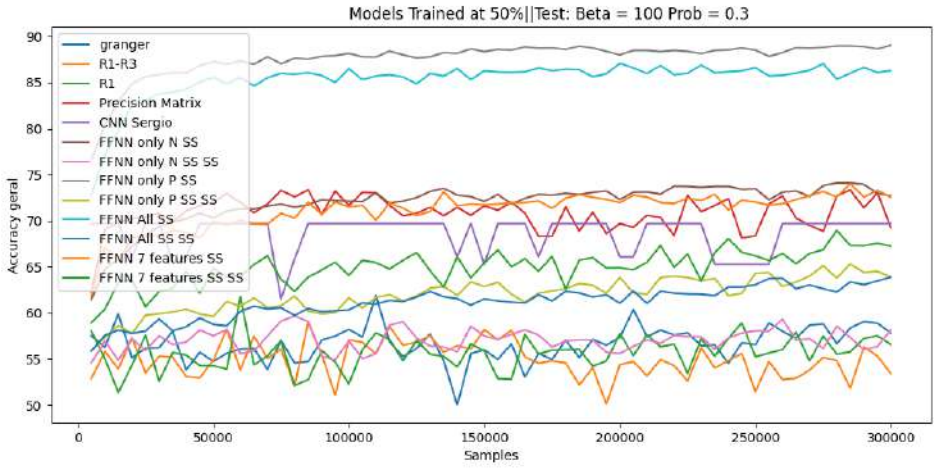
Each probability value is represented through a trio of graphics, elucidating accuracy across distinct nodes. Two of these graphics specifically delineate accuracy within the context of connected and disconnected nodes. This threefold categorization of accuracy has been preferred over the pursuit of a singular balanced metric, aligning with a pragmatic approach.

Upon perusal of the visual representations, it is evident that the top-performing entities are consistently the FFNNs and the Precision Matrix estimator. Notably, the legacy CNN's performance is not considered, as discerned from the accuracies of both connected and disconnected nodes, where the model tends to classify nearly all links between nodes as disconnected.

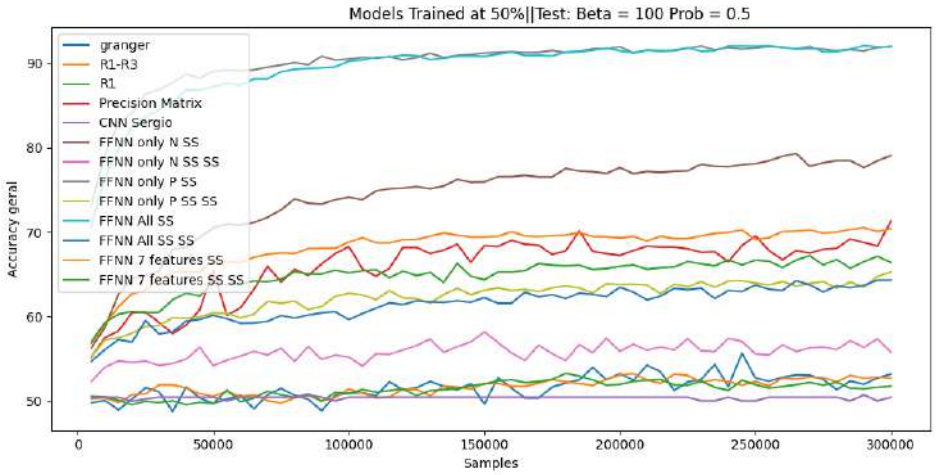
While the obtained results do not fall within the realm of the least favorable outcomes, the Precision Matrix estimator remarkably remains in close proximity and sometimes even surpasses the performance of this novel architecture.

5.4.2 Mix Features

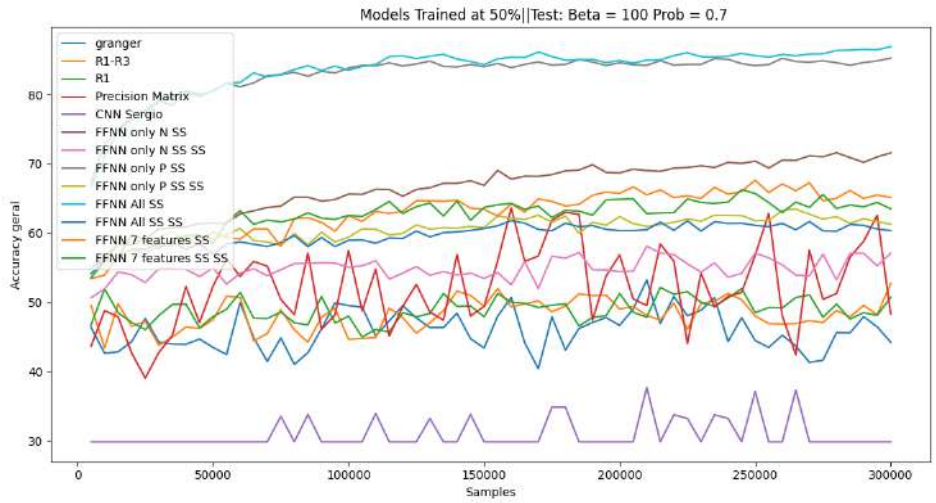
With the experiment made in 5.4.1 in mind, we came to the conclusion that the presence of the estimator (granger) in the feature array was not helping in the model's classification. Thus, instead of adding the estimator to the feature vector, we choose to include more correlation lag moments and their invertion, , $\mathcal{K}^M(n)$, as explained in 4.2.3. This way, we now consider two hundred correlation lags, being them the first fifty negative and positive correlation lags and their invertion. A feature set with only the positive lags and other only with negative lags (and their inverted lags) were tested too.



(a) 30% Test



(b) 50% Test



(c) 70% Test

Figure 5.18: Different Features Sets Results

Allow me to elucidate the legend provided: The term "*only N*" signifies a model that underwent training and testing exclusively using negative correlation lags and their corresponding inversions. Conversely, the designation "*only P*" pertains to a model exclusively incorporating positive lag moments and their corresponding inversions. The nomenclature "*All*" denotes a model that was trained using a comprehensive feature set consisting of both positive and negative lags, along with their respective inversions. The term "*7 features*" designates a model expounded upon in 5.4.1. All the networks used are directed.

In addition, the abbreviation "*SS*" denotes the utilization of the StandardScaler technique for normalizing the feature set. Meanwhile, "*SS SS*" indicates that both the time series data and the feature set underwent normalization using the StandardScaler methodology.

Upon meticulous analysis of Fig. 5.18, it is evident that the models trained solely with positive lags and those trained with a comprehensive set of all lags exhibit optimal performance. Subsequently, in the forthcoming figure, we opt to adopt the model employing all lags as our preferred methodology. We intend to compare this selected model across various regimes with the alternative methodologies delineated in Section 3.

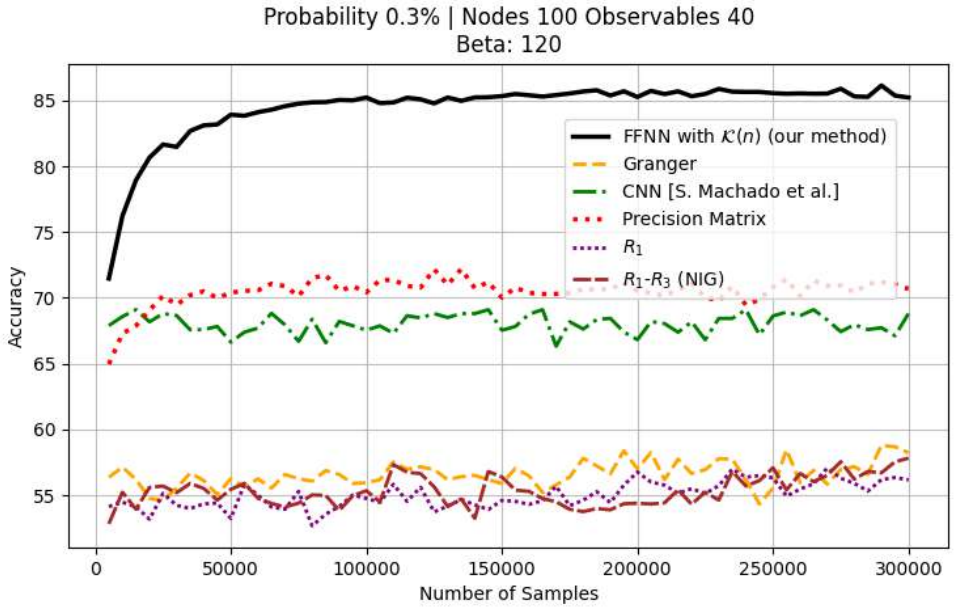


Figure 5.19: Mix Features Results - Scenario 1 - 30%test

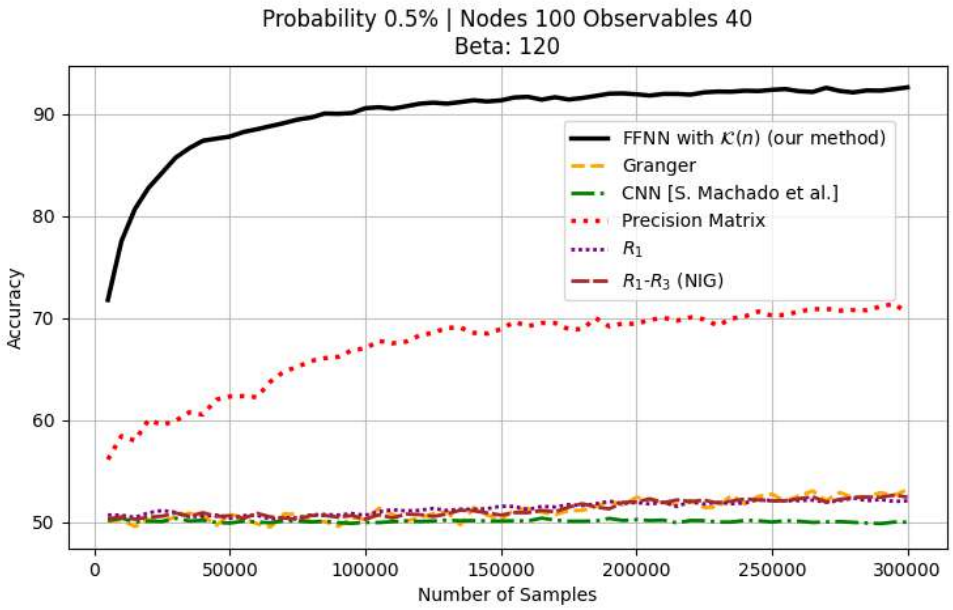


Figure 5.20: Mix Features Results - Scenario 1 - 50%test

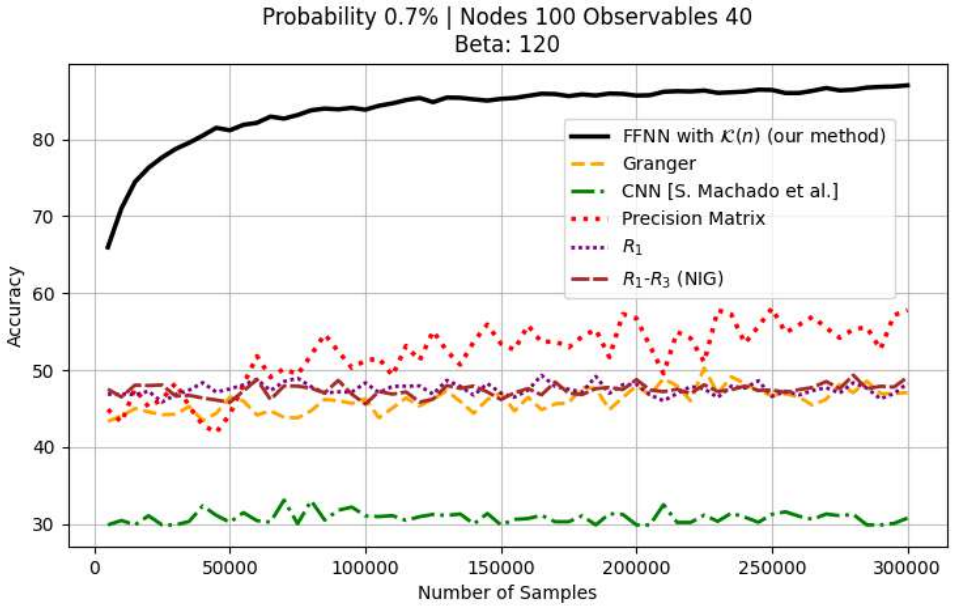


Figure 5.21: Mix Features Results - Scenario 1 - 70%test

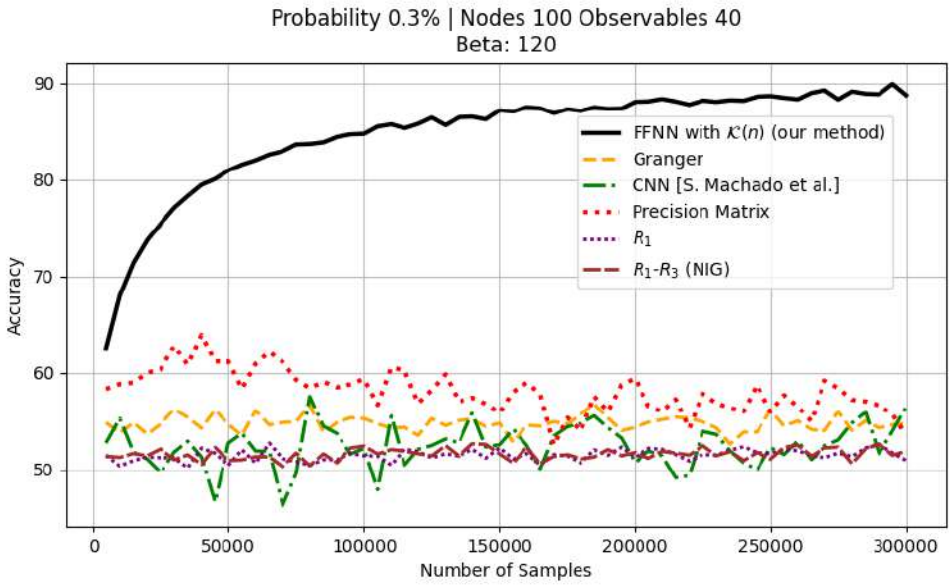


Figure 5.22: Mix Features Results - Scenario 2 - 30% test

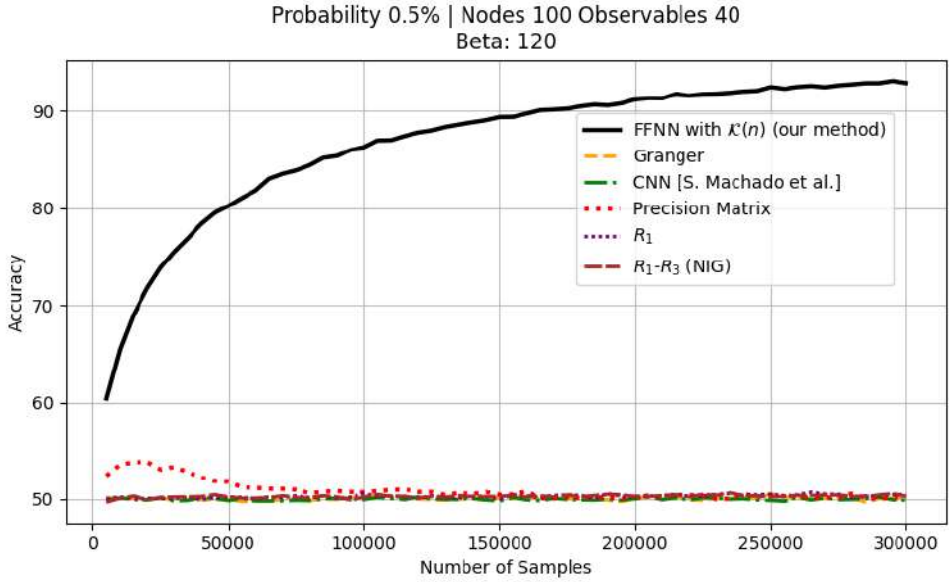


Figure 5.23: Mix Features Results - Scenario 2 - 50% test

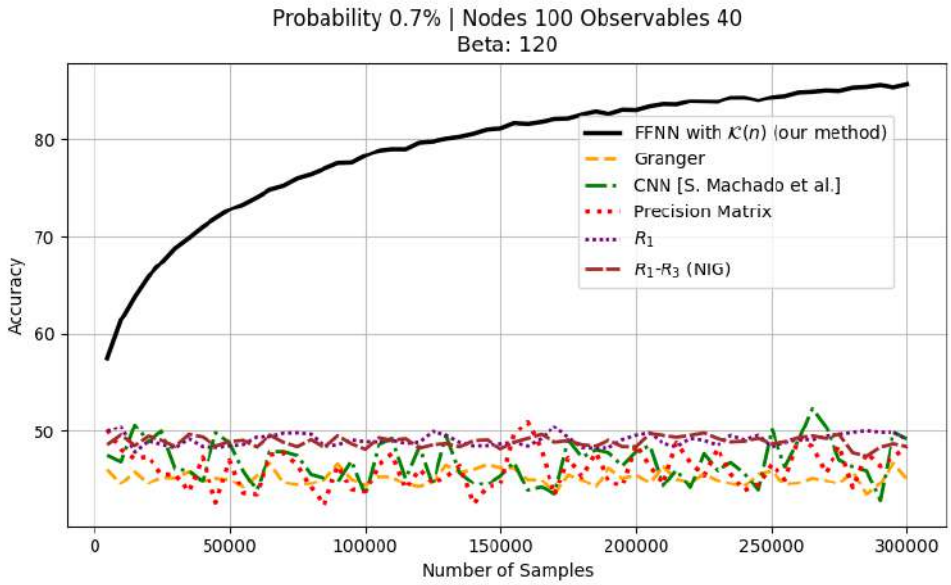


Figure 5.24: Mix Features Results - Scenario 2 - 70% test

The scenario 1 the underlying directed network used in the test has 40 nodes in total with only 20 observable and $\beta = 100$. In scenario 2 the β value is set to 120 and the total nodes of the network are 100 with only 40 observable.

Evident from Fig. 5.19 to Fig. 5.24, our proposed methodology consistently outperforms all other methods evaluated under the three distinct regimes tested at stage 3. This noteworthy achievement can be attributed to the synergistic amalgamation of correlation lags—both positive and negative—alongside their respective inverted matrices.

Our approach entails the training of a Feedforward Neural Network (FFNN) with a directed network architecture characterized by a 50% connectivity rate. The incorporation of varied beta values, inducing diverse colored noise patterns within the time series data, further enriches our feature set. Subsequently, these novel features are harnessed for model training.

In comparative terms, the performance exhibited by all competing estimators, as well as the legacy CNN model operating with its traditional features, falls short of attaining the level achieved by our novel model. The potency of our approach, underscored by the integration of diversified features and strategic network design, establishes a performance benchmark that remains unattainable by existing methodologies.

Another observation that can be made is the increase of the classification's difficulty in denser cases. In denser cases the network is complex and requires more samples to get the same performance as in sparse networks.

The next experiments show the influence of the values of some parameters in the models performance. In common they have as root a directed network with a total of 100 nodes and a total of 500000 samples. The colored noise parameter β is 100 excluding the one experiment where this value is changing. The observable nodes is fixed to 40 in the experiment where this value isn't changing. The prob-

ability of the network's structure connectivity is 50% except for the experiment where this same parameter varies.

Influence of the number of observable nodes

The next experiment shows the influence of the total nodes N and obervable nodes S ratio, $\frac{N}{S}$ in the performance of the methods.

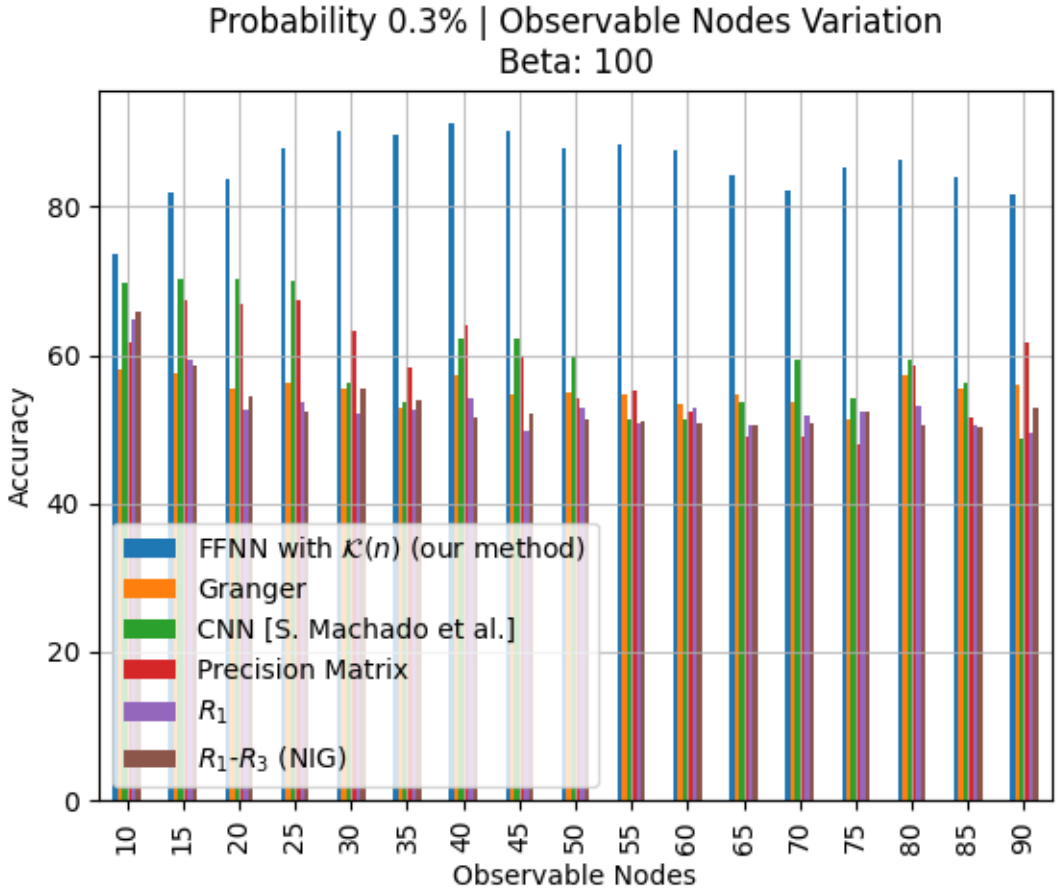


Figure 5.25: Influence of the number of Observable nodes - 30% Test

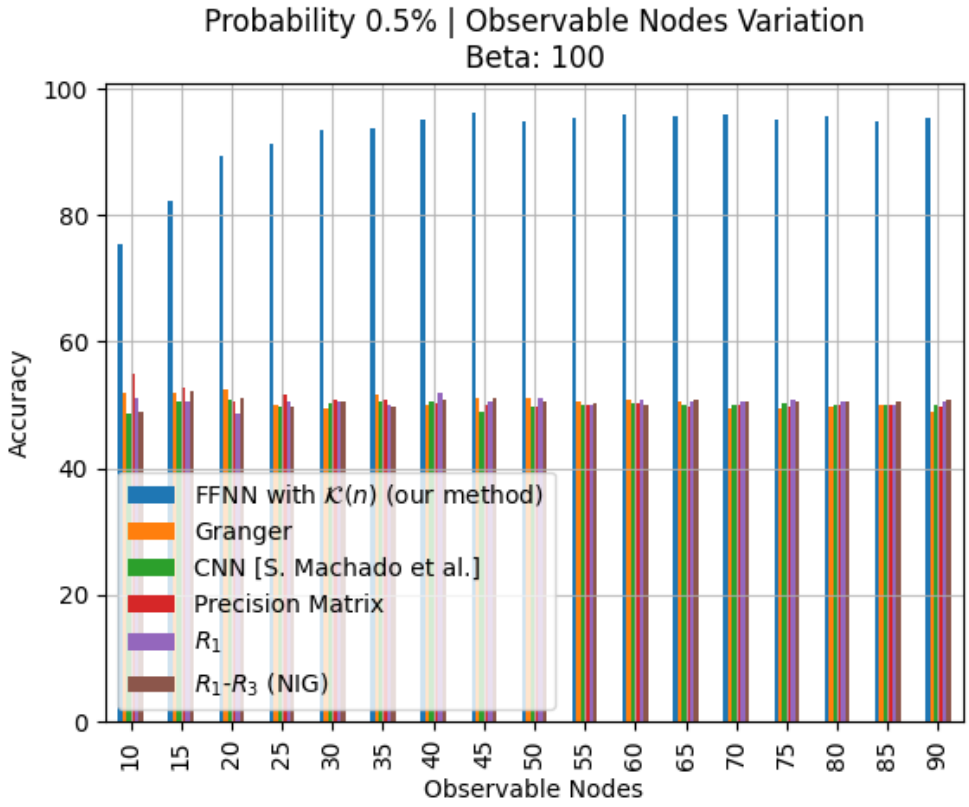


Figure 5.26: Influence of the number of Observable nodes - 50% Test

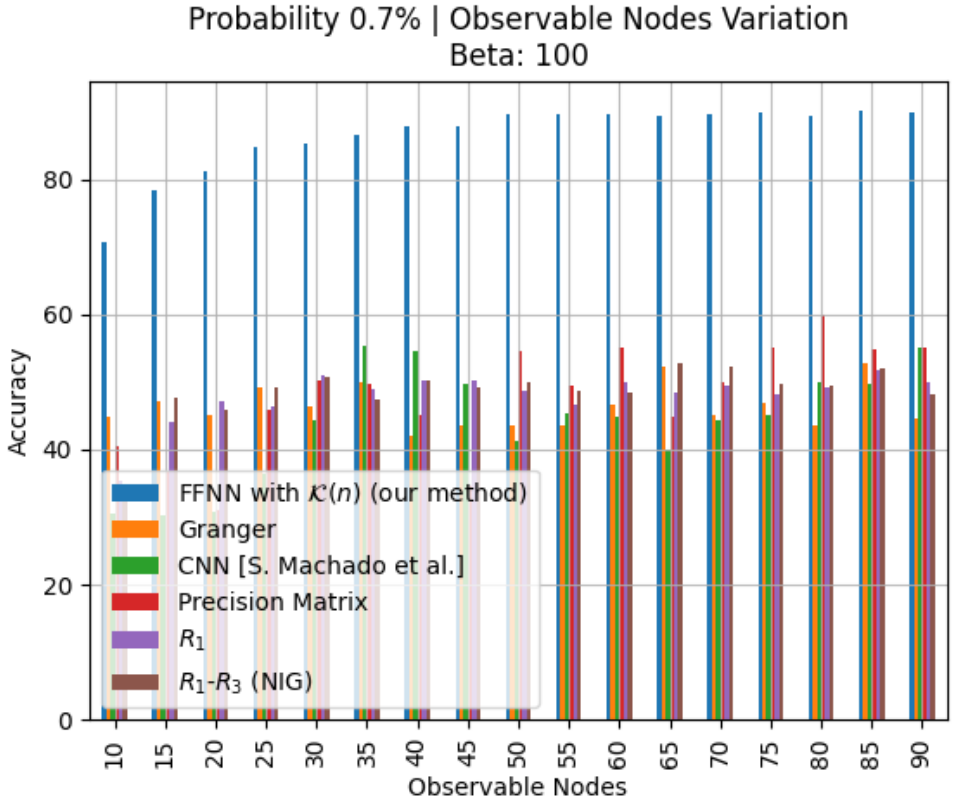


Figure 5.27: Influence of the number of Observable nodes - 70% Test

In this case, from the observation of Fig.5.25 to Fig.5.27, the proposed method surpasses every other approach in all partial observability cases, being the network balanced, sparse or dense.

Influence of the Beta Parameter

The next experiment shows the influence of the β colored noise parameter in the performance of the methods.

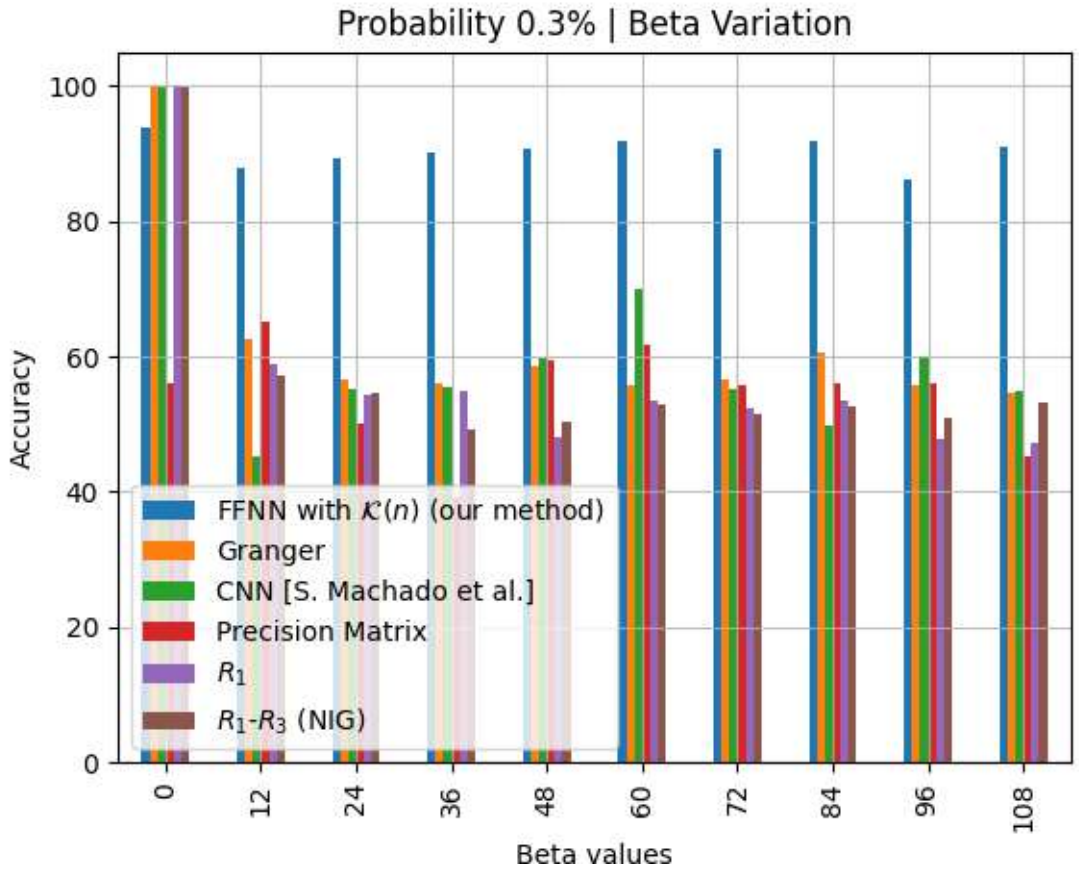


Figure 5.28: Influence of Beta - 30% Test

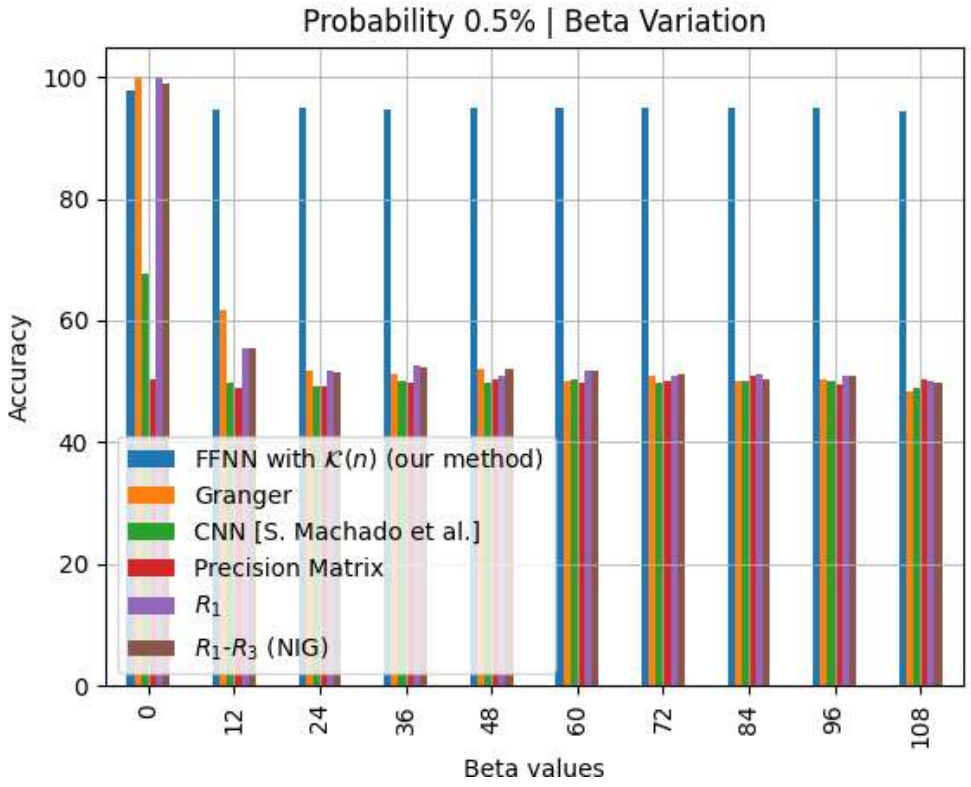


Figure 5.29: Influence of Beta - 30% Test

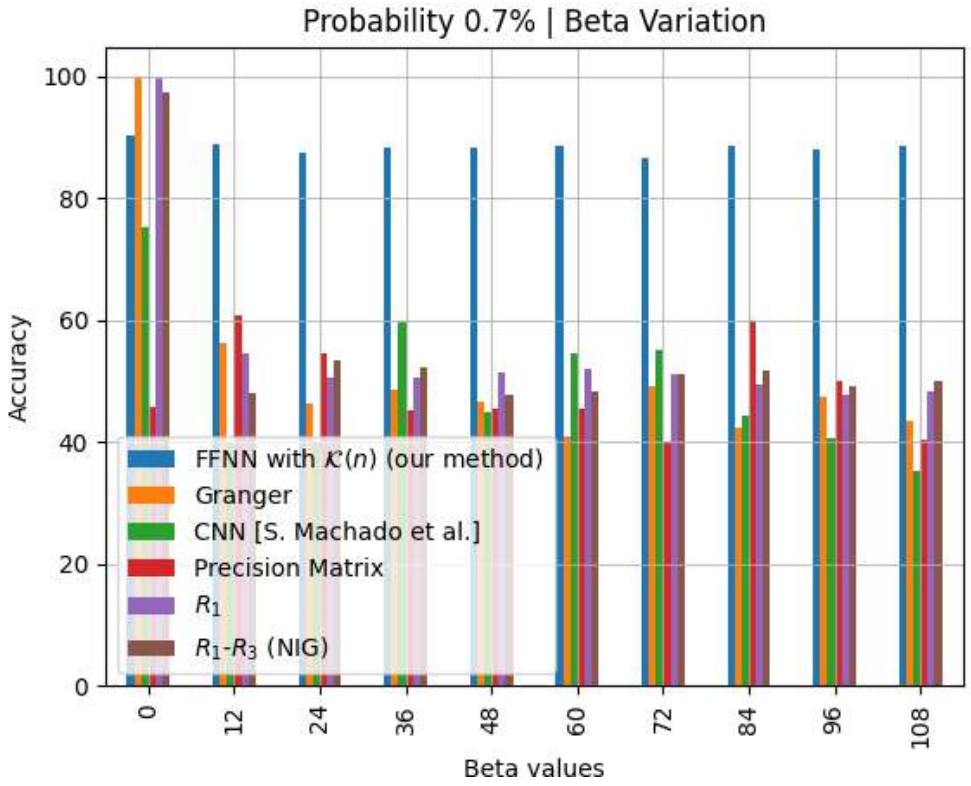


Figure 5.30: Influence of Beta - 30% Test

The noise will be diagonal when $\beta = 0$. As we can see from Fig.5.28 to Fig.5.30, when there is only diagonal noise applied in the network, the other methods can keep up with the one proposed in this thesis. However, as soon as we *turn on* the colored noise i.e., we increase the value of β all the performances of the other methods decreases and ours stays with a good performance.

Influence of the Connectivity

In this experiment the probability of the connectivity p in the Erdős-Rényi algorithm is changed. The purpose is to analyse the influence of the network structure in the performance of the models.

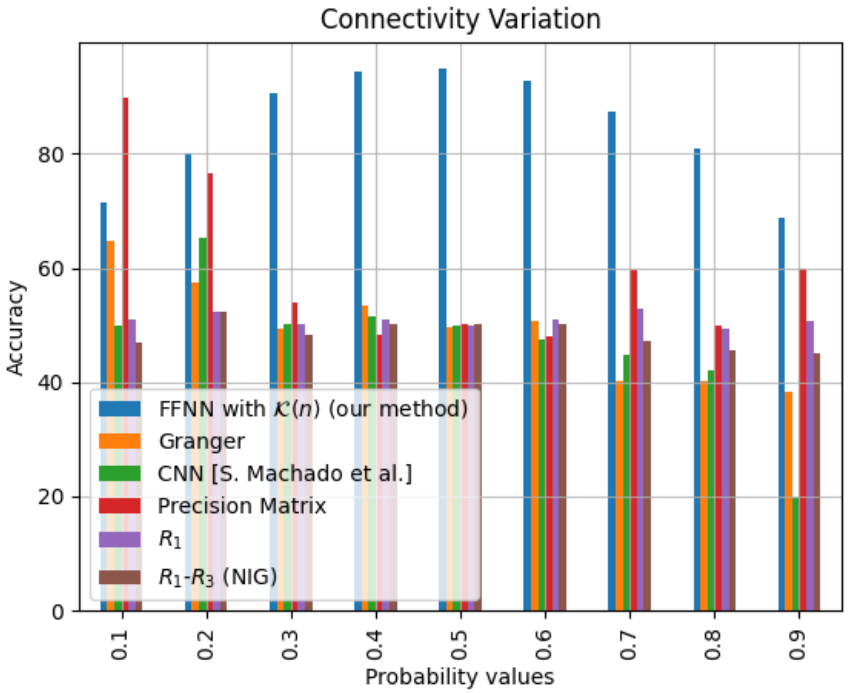


Figure 5.31: Influence of Connectivity

Our proposed method only fails in the sparser case when the probability linked with the connectivity of the network is $0.1(10\%)$. As we increase the directed network connectivity our method has the best performance. Remark that denser cases are harder then sparser cases.

5.5 Real Data

This Section contains the results in real data that we could find and test it with the same logic as the experiments shown before. In Section 5.5.1 we use a dataset that has adjacency matrices obtained through tractography images of the brain.

5.5.1 Real Network

Reference [Škoch, 2022] by segmenting the tractography results into larger anatomical units, the researchers gain insights into the structural relationships between different parts of the brain. This process culminates in the creation of a structural connectivity matrix, which offers estimates of connection strength among all regions of interest. However, the processing of raw data is intricate, computationally demanding, and necessitates expert quality control, potentially discouraging researchers with limited experience in the field.

To address this challenge, the researchers present a valuable contribution: a dataset of brain structural connectivity matrices that are preprocessed and ready for modeling and analysis. This dataset is designed to be accessible to a wide community of scientists, enabling researchers to delve into brain connectivity research without grappling with the complexities of data processing.

The dataset not only includes brain structural connectivity matrices but also provides the underlying raw diffusion and structural data. Moreover, it offers basic demographic information pertaining to 88 healthy subjects. By making this dataset available, the researchers aim to facilitate and accelerate advancements in brain connectivity research, opening doors for interdisciplinary collaboration and fostering a deeper understanding of the human brain's intricate workings.

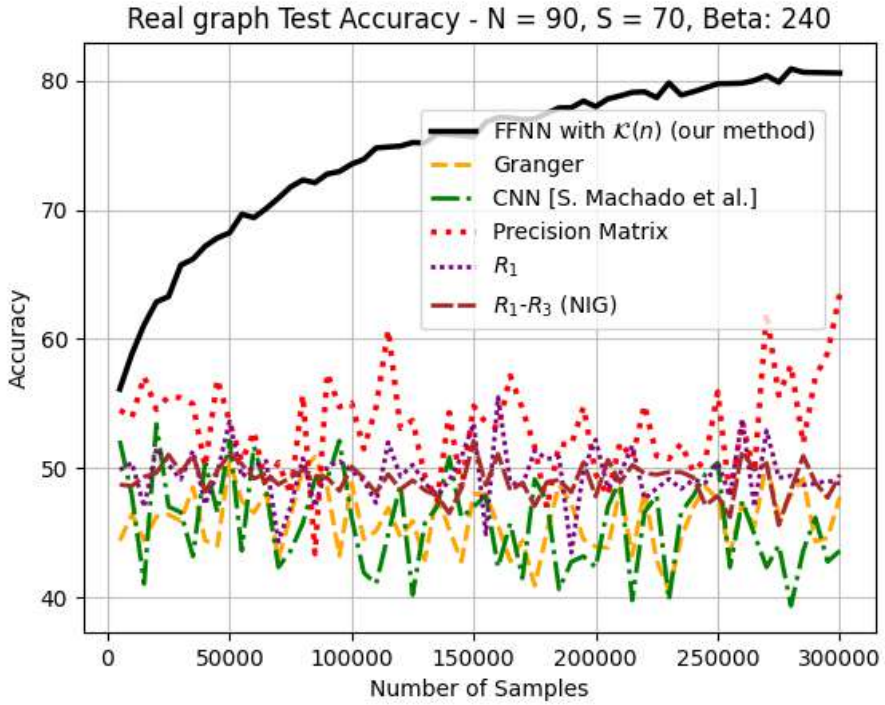


Figure 5.32: Real network Results

A noteworthy outcome becomes evident in Fig. 5.32, wherein our model demonstrates superior performance compared to the alternative approaches delineated in Chapter 3. Specifically, the model in question is characterized as a network

comprising 90 nodes, encompassing 70 nodes that are observable, and subjected to colored noise with a beta value of 240.

Chapter 6

Concluding Remarks

This Section provides a succint overview of the contributions in the thesis and points to feature directions.

6.1 Contributions

This thesis was focused on tackling the intricate task of deducing the interaction structure within a networked dynamic system characterized by linearity and stochastic behavior. The interactions structure was abstracted as a directed network that needed to be infered from the time series activity at distinct nodes. Specifically, we delved into the problem of time series data originating from these systems, which were further perturbed by the presence of colored noise. We assumed partial observability, where the data only covers a portion of the nodes, leaving some unobserved (partial observability). In this context, we introduced an inventive collection of feature vectors that are derived from the available time series. These feature vectors represent statistical descriptors, offering insights into the connections and direction of connections between pairs of nodes. Our contributions are twofold. First, we established that this set of features is endowed with linear separability, which means that, with a high degree of probability, it's possible to identify a hyperplane within the feature space. This hyperplane effectively segregates the features linked with connected node pairs from those linked with disconnected pairs. Second, we substantiated the structural reliability of the feature vectors via a rigorous proof under specific parameter settings (2). Consequently, this implies that if we possess the correct hyperplane for separation, we can consistently categorize node pairs as either connected or disconnected. To leverage this, a variety of machine learning techniques can be trained on these distinctive features. In our case, we opted to utilize Feedforward Neural Networks (FFNNs) to harness the power of this feature set for causal inference, leading to an advanced method. The trained FFNN exhibited exceptional generalizability. Despite being trained on a specific synthetic network – one generated using an Erdős–Rényi random graph model with 100 nodes and a connection probability of 0.5 – it displayed robust performance across a wide spectrum of connectivity scenarios, even encompassing real-world networks. Our findings

have been submitted for publication, and we're also in the process of preparing additional results for dissemination.

6.2 Future Work

In the preliminary report, our ultimate objective encompassed the application of our devised methodology to authentic brain data networks. As outlined in Section 5.5.1, the employed network structure is derived from neuroimaging data and is formulated through statistical procedures that capture patterns of brain activity. Concurrently, we aspired to progress toward the utilization of real-time series data sourced directly from genuine brain functional magnetic resonance imaging (FMRI) or electroencephalography (EEG) signals. This endeavor entails the utilization of statistical methodologies to generate these time series, introducing an additional facet to our experimental framework and potentially bolstering the credibility of our approach.

Moreover, the preliminary report alludes to the exploration of Nonlinear Dynamical Systems. It is important to note that this facet of the research has yet to be subjected to empirical investigations. It is therefore suggested that this represents a forthcoming avenue of inquiry, wherein the proposed methodology could be applied, examined, and potentially refined to accommodate the intricacies of such systems. The prospect of developing a novel feature vector tailored to the characteristics of Nonlinear Dynamical Systems is also contemplated, provided the need arises for achieving optimal performance within this domain.

References

T. Javidi A. Lalitha and A. D. Sarwate. Social learning and distributed hypothesis testing. *IEEE Transactions on Information Theory*, 64:6161–6179, September 2018. doi: 10.1109/TIT.2018.2837050.

Adolphs. R. cognitive neuroscience of human social behaviour. *Nat Rev Neurosci 4*, 2003.

John Rothwell Alvaro Pascual-Leone, Vincent Walsh. Transcranial magnetic stimulation in cognitive neuroscience – virtual lesion, chronometry, and functional connectivity. *Current Opinion in Neurobiology*, 2000.

Yupeng Chen, Zhiguo Wang, and Xiaojing Shen. An unbiased symmetric matrix estimator for topology inference under partial observability. *IEEE Signal Processing Letters*, 29(02):1257–1261, 2022. doi: 10.1109/LSP.2022.3177076.

C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968. doi: 10.1109/TIT.1968.1054142.

A. Ganesh, L. Massoulié, and D. Towsley. The effect of network topology on the spread of epidemics. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 2, pages 1455–1466, March 2005. doi: 10.1109/INFCOM.2005.1498374.

Philipp Geiger, Kun Zhang, Bernhard Schoelkopf, Mingming Gong, and Dominik Janzing. Causal inference by identification of vector autoregressive processes with hidden components. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1917–1925. PMLR, 07–09 Jul 2015.

Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of Convex Analysis*. Grundlehren Text Editions. Springer-Verlag Berlin Heidelberg, 2001. ISBN 978-3-540-42205-1. doi: 10.1007/978-3-642-56468-0.

Haiqing Huang and Mingzhou Ding. Linking functional connectivity and structural connectivity quantitatively: A comparison of methods. *Brain Connectivity*, 6(2):99–108, september 2016.

Ali Jadbabaie, Pooya Molavi, Alvaro Sandroni, and Alireza Tahbaz-Salehi. Non-bayesian social learning. *Games and Economic Behavior*, 76(1):210 – 225, 2012. ISSN 0899-8256. doi: https://doi.org/10.1016/j.geb.2012.06.001. URL `http://www.sciencedirect.com/science/article/pii/S0899825612000851`.

Raphael Liégeois, Augusto Santos, Vincenzo Matta, Dimitri Van De Ville, and Ali H. Sayed. Revisiting correlation-based functional connectivity and its relationship with structural connectivity. *Network Neuroscience*, 4(4):1235–1251, 2020. doi: 10.1162/netn_a_00166.

Po-Ling Loh and Martin J. Wainwright. Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses. *The Annals of Statistics*, 41(6):3022 – 3049, 2013. doi: 10.1214/13-AOS1162. URL `https://doi.org/10.1214/13-AOS1162`.

Sergio Machado. Learning the graph of networked dynamical systems under partial-observability via artificial neural networks. September 2022.

Sergio Machado, Anirudh Sridhar, Paulo Gil, Jorge Henriques, Jose M. F. Moura, and Augusto Santos. Recoverying the graph underlying networked dynamical systems under partial-observability: a deep learning approach. In *Proceedings of the 37th AAAI Coference on Artificial Intelligence (to appear)*, AAAI'37. AAAI, 2023.

Olaf Sporns Martijn P. van den Heuvel. Rich-club organization of the human connectome. *Journal of Neuroscience*, 2011.

V. Matta, A. Santos, and A. H. Sayed. Graph learning over partially observed diffusion networks: Role of degree concentration. June 2020a. URL `https://arxiv.org/abs/1904.02963v2`.

Vincenzo Matta, Virginia Bordignon, Augusto Santos, and Ali H. Sayed. Interplay between topology and social learning over weak graphs. *IEEE Open Journal of Signal Processing*, 1:99–119, 2020b. doi: 10.1109/OJSP.2020.3006436.

Vincenzo Matta, Augusto Santos, and Ali H. Sayed. Graph learning under partial observability. *Proceedings of the IEEE*, 108:2049 – 2066, 11 2020c. doi: 10.1109/JPROC.2020.3013432.

Vincenzo Matta, Augusto Santos, and Ali H. Sayed. Graph learning over partially observed diffusion networks: Role of degree concentration. *IEEE Open Journal of Signal Processing*, pages 335–371, 2022. doi: 10.1109/OJSP.2022.3189315.

Flaviano Morone, Kevin Roth, Byungjoon Min, H. Eugene Stanley, and Hernán A. Makse. Model of brain activation predicts the neural collective influence map of the brain. *Proceedings of the National Academy of Sciences*, 114(15):3849–3854, 2017. doi: 10.1073/pnas.1620808114.

Xiao-Long Ren, Niels Gleinig, Dirk Helbing, and Nino Antulov-Fantulin. Generalized network dismantling. *Proceedings of the National Academy of Sciences*, 116(14):6554–6559, 2019. ISSN 0027-8424. doi: 10.1073/pnas.1806108116. URL `https://www.pnas.org/content/116/14/6554`.

Augusto Santos, Vincenzo Matta, and A. H. Sayed. Local tomography of large networks under the low-observability regime. *IEEE Transactions on Information Theory*, 66:587 – 613, 01 2020. doi: 10.1109/TIT.2019.2945033.

Augusto Santos, Diogo Rente, Rui Seabra, and José M. F. Moura. Learning the causal structure of networked dynamical systems under latent nodes and structured noise. In *Submitted*, 2023.

Ali H. Sayed. Adaptation, Learning, and Optimization over Networks. *Found. Trends Mach. Learn.*, 7(4-5):311–801, 2014. ISSN 1935-8237. doi: 10.1561/2200000051.

CJ Stam, BF Jones, G Nolte, M Breakspear, and Ph Scheltens. Small-World Networks and Functional Connectivity in Alzheimer's Disease. *Cerebral Cortex*, 17 (1):92–99, 2007. doi: 10.1093/cercor/bhj127.

Zhijiang Wang, Zheng-Jia Dai, Gaolang Gong, Changsong Zhou, and Yong He. Understanding structural-functional relationships in the human brain: A large-scale network perspective. *The Neuroscientist : a review journal bringing neurobiology, neurology and psychiatry*, 21, 06 2014. doi: 10.1177/1073858414537560.

Rehák Bučková B. Mareš J. Škoch, A. Human brain structural connectivity matrices–ready for modelling. *Sci Data 9*, 2022. doi: 10.1038/s41597-022-01596-9.