



|     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15  |
| 16  | 17  | 18  | 19  | 20  | 21  | 22  | 23  | 24  | 25  | 26  | 27  | 28  | 29  | 30  | 31  |
| 32  | 33  | 34  | 35  | 36  | 37  | 38  | 39  | 40  | 41  | 42  | 43  | 44  | 45  | 46  | 47  |
| 48  | 49  | 50  | 51  | 52  | 53  | 54  | 55  | 56  | 57  | 58  | 59  | 60  | 61  | 62  | 63  |
| 64  | 65  | 66  | 67  | 68  | 69  | 70  | 71  | 72  | 73  | 74  | 75  | 76  | 77  | 78  | 79  |
| 80  | 81  | 82  | 83  | 84  | 85  | 86  | 87  | 88  | 89  | 90  | 91  | 92  | 93  | 94  | 95  |
| 96  | 97  | 98  | 99  | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |
| 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 |
| 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 |
| 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 |
| 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 |
| 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 |

# Text Classification with Cellular Automata Networks

Exploiting The Reservoir Computing Approach

Master's thesis in Data Science and AI

Oscar Johansson



MASTER'S THESIS 2024

# Text Classification with Cellular Automata Networks

Exploiting The Reservoir Computing Approach

Oscar Johansson



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

Department of Microtechnology and Nanoscience  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2024

Text Classification with Cellular Automata Networks  
Exploiting The Reservoir Computing Approach  
Oscar Johansson

© Oscar Johansson, 2024.

Supervisor: Zoran Konkoli, Department of Microtechnology and Nanoscience  
Examiner: Zoran Konkoli, Department of Microtechnology and Nanoscience

Master's Thesis 2024  
Department of Microtechnology and Nanoscience  
Chalmers University of Technology  
SE-412 96 Gothenburg  
Telephone +46 31 772 1000

Cover: Categorisation of different transition rules based on experiment performance.

Typeset in L<sup>A</sup>T<sub>E</sub>X  
Gothenburg, Sweden 2024

Text Classification with Cellular Automata Networks  
Exploiting The Reservoir Computing Approach  
Oscar Johansson  
Department of Microtechnology and Nanoscience  
Chalmers University of Technology

## Abstract

The thesis focuses on several options of exploiting reservoir computing ideas in the context of text classification, using cellular automata networks as the reservoir. The key innovative aspect of the study is related to finding options for computing with low-density networks, which should be easier to manufacture. The overarching idea is to extend Wolfram's elementary cellular automata rules and implement them within the framework of random cellular automata networks. The primary objectives of this study are as follows: (i) To determine the optimal network topologies, (ii) to identify the most effective transition rules within these networks, and (iii) to develop a robust methodological framework for finding the best networks. By exploring various network configurations, the aim is to uncover the structural characteristics that facilitate efficient information propagation and decision-making within the cellular automata framework. Observations from the study indicate that there exist cellular automata network configurations which can function as efficient reservoirs. Moreover, all considered cellular automata networks are categorised into different groups based on their performance, providing a solid foundation for further research.

Keywords: Reservoir computing, cellular automata networks, cellular automata, text classification, machine learning, natural language processing, supervised learning, information theory, elementary cellular automata.



# Acknowledgements

I would like to thank my supervisor, Zoran Konkoli, for all support and guidance I have received throughout my thesis work. I greatly appreciate your ability to create an encouraging and productive environment, allowing me to feel comfortable and do my best work. Working with you has inspired me to continue on my academic journey and pursue a PhD degree. Thank you.

I would also like to extend my thanks to everyone I have met at the Department of Microtechnology and Nanoscience at Chalmers. I greatly value the help I have received and all the conversations we have had.

Finally, I am deeply grateful to my friends and family for their interest and support, which continue to motivate me and allowing me to aim higher.

Oscar Johansson, Gothenburg, 2024-06-17



# Contents

|   |             |
|---|-------------|
| <b>List of Figures</b>                                    | <b>xi</b>   |
| <b>List of Tables</b>                                     | <b>xiii</b> |
| <b>1 Introduction</b>                                     | <b>1</b>    |
| 1.1 Problem Statement . . . . .                           | 2           |
| 1.2 Research Objectives . . . . .                         | 2           |
| 1.3 Limitations . . . . .                                 | 3           |
| 1.4 Contributions . . . . .                               | 3           |
| 1.5 Thesis Structure . . . . .                            | 4           |
| <b>2 Background</b>                                       | <b>5</b>    |
| 2.1 Reservoir Computing . . . . .                         | 5           |
| 2.1.1 Physical reservoirs . . . . .                       | 6           |
| 2.1.2 The readout layer . . . . .                         | 7           |
| 2.2 Cellular Automata . . . . .                           | 7           |
| 2.2.1 Elementary Cellular Automata . . . . .              | 8           |
| 2.2.2 Cellular Automata Networks . . . . .                | 9           |
| 2.2.3 Cellular Automata Networks as a Reservoir . . . . . | 9           |
| 2.3 Network Theory . . . . .                              | 10          |
| 2.4 Information Theory . . . . .                          | 10          |
| <b>3 Reservoir Computing Framework</b>                    | <b>13</b>   |
| 3.1 General Architecture . . . . .                        | 13          |
| 3.2 Input structure . . . . .                             | 14          |
| 3.3 Reservoir . . . . .                                   | 15          |
| 3.3.1 Network Topology . . . . .                          | 16          |
| 3.3.2 Transition Rule . . . . .                           | 18          |
| 3.4 Readout Layer . . . . .                               | 19          |
| <b>4 Methods</b>  | <b>21</b>   |
| 4.1 Classification . . . . .                              | 21          |
| 4.2 Rule Entropy . . . . .                                | 22          |
| <b>5 Results</b>  | <b>25</b>   |
| 5.1 Network Topology Performance . . . . .                | 25          |

|          |   |           |
|----------|---|-----------|
| 5.2      | Transition Rule Performance . . . . .                       | 27        |
| 5.2.1    | General Performance . . . . .                               | 27        |
| 5.2.2    | Categorisation based on Accuracy Distribution . . . . .     | 31        |
| 5.2.3    | Categorisation of Rules for $\mathcal{N}_{PD-2}$ . . . . .  | 32        |
| 5.2.4    | Categorisation of Rules for $\mathcal{N}_{PD-3}$ . . . . .  | 34        |
| 5.2.5    | Categorisation of Rules for $\mathcal{N}_{PD-4}$ . . . . .  | 36        |
| 5.3      | Information Gain . . . . .                                  | 38        |
| <b>6</b> | <b>Discussion</b>   | <b>41</b> |
| 6.1      | Finding the Correct Network Topology . . . . .              | 41        |
| 6.2      | Selecting an Appropriate Transition Rule . . . . .          | 42        |
| 6.2.1    | Categorisations of Elementary Cellular Automata . . . . .   | 42        |
| 6.2.2    | The Ideal Transition Rule . . . . .                         | 44        |
| 6.2.3    | The Inverse Rules Problem . . . . .                         | 45        |
| 6.3      | Cellular Automata Networks as a Reservoir . . . . .         | 45        |
| 6.4      | Reservoir Computing as a Methodological Framework . . . . . | 46        |
| 6.5      | Genetic Algorithm . . . . .                                 | 47        |
| 6.6      | Future Work . . . . .                                       | 47        |
| <b>7</b> | <b>Conclusion</b>   | <b>49</b> |
|          | <b>Bibliography</b>   | <b>51</b> |
| <b>A</b> | <b>Appendix 1</b>   | <b>I</b>  |

# List of Figures

|      |  |    |
|------|--|----|
| 2.1  | Visual representation of elementary cellular automata rule 91. . . . .   | 8  |
| 3.1  | An overview of the model’s architecture. . . . .   | 13 |
| 3.2  | Markov models for generating text phrases with different text structures. . . . .  | 15 |
| 5.1  | Average performance $\mu(\mathcal{N}_\gamma, \mathcal{I}, R_i)$ over all experiments for each rule $R_i, i \in \{0, 1, \dots, 255\}$ , grouped by network model $\gamma$ . The rules are sorted from worst to best performance within each group, with the x-axis representing the rank of each rule within its respective group. . . . .                                    | 25 |
| 5.2  | Average performance $\mu(\mathcal{N}_{PD-\delta}, \mathcal{I}_{a,d}, R_{91})$ with error bars representing $\sigma^2(\mathcal{N}_{PD-\delta}, \mathcal{I}_{a,d}, R_{91})$ over 10 runs for each network model $PD-\delta$ , where $\delta \in \{2, 3, \dots, 19\}$ . The shaded gray area highlights the region where a significant drop in performance is observed. . . . . | 26 |
| 5.3  | Average normalized prediction accuracy $\mu(\mathcal{N}_{PD-\alpha}, \mathcal{I}', R_i)$ over all three input sequences $\mathcal{I}'$ for each rule, based on the network models: (a) PD-2, (b) PD-3, (c) PD-4, and (d) the combined average for all three network models. Rules are presented in numerical order . . . . .   | 28 |
| 5.4  | Four groups of distribution types. . . . .   | 31 |
| 5.5  | Categorisation of rules on $\mathcal{N}_{PD-2}$ . . . . .  | 33 |
| 5.6  | Four of the best rules on $\mathcal{N}_{PD-2}$ . . . . .   | 34 |
| 5.7  | Categorisation of rules on $\mathcal{N}_{PD-3}$ . . . . .  | 35 |
| 5.8  | Four of the best rules on $\mathcal{N}_{PD-3}$ . . . . .   | 36 |
| 5.9  | Categorisation of rules on $\mathcal{N}_{PD-4}$ . . . . .  | 37 |
| 5.10 | Four of the best rules on $\mathcal{N}_{PD-4}$ . . . . .   | 37 |
| 6.1  | Visualisation of the three elementary cellular automata rules with the best performance on network model $\mathcal{N}_{PD-2}$ , $\mathcal{N}_{PD-3}$ , and $\mathcal{N}_{PD-4}$ , respectively. . . . .  | 42 |



# List of Tables

|     |   |     |
|-----|---|-----|
| 5.1 | Mean prediction accuracy $\mu(-, \mathcal{I}_{a,b}, \mathcal{R})$ , $\mu(-, \mathcal{I}_{a,c}, \mathcal{R})$ , and $\mu(-, \mathcal{I}_{a,d}, \mathcal{R})$ without reservoir, i.e., the readout layer is applied directly to the input sequence. . . . .   | 27  |
| 5.2 | Numerical results from the experiments, including the rank, mean accuracy, and standard deviation for each rule on each network model, as well as the overall results. Only a subset of the rules is shown; complete results can be found in Table A.1 in Appendix A. . . . .   | 30  |
| 5.3 | Average information gain for 2, 3, and 4 predecessors, respectively. . .  | 38  |
| A.1 | Full results for $\mathcal{N}_{PD-2}$ , $\mathcal{N}_{PD-3}$ , and $\mathcal{N}_{PD-4}$ from experiments, including the rank, mean accuracy, and standard deviation for each rule and network pair. Additionally, average results over all three networks is also included. The rules are ordered based on the overall results. . . .   | I   |
| A.2 | Full results for $\mathcal{N}_{BA}$ , $\mathcal{N}_{ER}$ , and $\mathcal{N}_{WS}$ from experiments, including the rank, mean accuracy, and standard deviation for each rule and network pair. Additionally, average results over all three networks is also included. The rules are ordered based on the overall results. . . .   | IV  |
| A.3 | Full information gain table, including the entropy $H(Y)$ , the information gain for the current (“middle”) cell $IG_M$ , and the information gain for predecessors $IG_A$ , $IG_B$ , $IG_C$ and $IG_D$ , corresponding to the order the predecessors appear in the list of predecessors. The information gain is calculated for cells with two, three and four predecessors, separately. . . . . | VII |



# 1

## Introduction

Computation fundamentally involves evaluating a function where an input is transformed into an output in a reproducible way [1]. To achieve this, mathematical abstractions representing the input and output must be embodied within physical systems [2]. Traditionally, this transformation is executed through a series of operations on the data using binary logic and electrical signals, structured according to the well-established von Neumann architecture [3]–[5]. In this standard approach, there is a clear distinction between the central processing unit, the memory, and the data transfer pathways between them. This architecture underpins most personal computers and smartphones today. However, it suffers from a significant limitation known as the von Neumann bottleneck [5], [6], which arises from the physical and practical constraints on the number and efficiency of connections between components. As systems scale up, the complexity and overhead of managing these connections increase, leading to inefficiencies and limiting overall performance.

To address these challenges, researchers have explored alternative computation methodologies [7]. Unconventional computing encompasses computational approaches that deviate from the traditional von Neumann architecture. This category includes, but is far from limited to, quantum computing, which uses quantum bits (qubits) instead of traditional bits; optical computing, which processes information using light rather than electrical signals; and neuromorphic computing, which mimics the neural structure and functioning of the human brain to perform computations. These unconventional methods offer distinct advantages and application areas where they can outperform or complement conventional techniques.

This thesis explores the application of reservoir computing for text classification. Reservoir computing is a machine learning technique that leverages principles of neuromorphic computing. A typical reservoir computer consists of a dynamical system, known as the reservoir, driven by an external input that alters its state. Instead of analysing the original input directly, the state of the system is analysed and assigned a label using a linear readout layer. In this way, the information contained in the input sequence is transformed into the information represented by the label, constituting the computational process where one type of information is converted into another. The key advantage of this approach is that the readout layer can be simple, and thus easy to train, since the reservoir performs the computation.

A critical question which will be discussed in this thesis, is how to construct an effective reservoir. This research focuses on a novel approach to building such

reservoirs using cellular automata networks. By investigating the properties and dynamics of such networks, this study aims to enhance our understanding of their potential as reservoirs in the context of text classification.

### 1.1 Problem Statement

Traditional machine learning models frequently encounter limitations, such as large model sizes, time-consuming training processes, and substantial computational resource consumption. These challenges can restrict their deployment and limit their scalability and environmental sustainability. A significant contributing factor to these limitations is the inherent inefficiencies of the von Neumann architecture. Reservoir computing has emerged as a promising solution to these issues. Unconventional computing facilitates the development of machine learning models that are smaller, require minimal training, and consume fewer resources. Such models have the potential to outperform traditional methods in various tasks [8], while offering greater efficiency and reduced costs. Furthermore, these systems could be directly integrated into hardware.

The success of reservoir computing is critically dependent on the underlying dynamical system used as the reservoir. Although certain dynamical systems have demonstrated potential, this area remains underexplored, revealing a significant gap in current research. This study addresses this gap by investigating a versatile dynamical system in Cellular Automata Networks (CAN). Cellular automata networks feature complex dynamics governed by local interactions, rendering them intriguing candidates for effective reservoirs.

Through rigorous numerical experiments, the aim is to evaluate the capabilities of CAN as a reservoir for the classification of text structures, a fundamental problem in natural language processing. The objective is to differentiate between various forms of textual organization, which could ultimately contribute to applications such as language identification, authorship attribution, and the distinction between AI-generated and human-generated text. Given the novel application of CAN in this context, initial investigations will target simpler text structures to establish baseline capabilities and insights.

### 1.2 Research Objectives

The primary objective is to identify key properties of cellular automata networks that enhance their functionality as reservoir computing systems, particularly for text classification tasks. This goal is part of a broader attempt to understand the characteristics that define an effective reservoir. Specifically, this study aims to achieve the following objectives:

1. To determine the optimal network topology of cellular automata networks in the context of reservoir computing for text classification.

2. To identify the most effective transition rule for cellular automata within these networks.
3. To develop a robust methodological framework for addressing such complex and open-ended challenges.

### 1.3 Limitations

*To determine the optimal network topology of cellular automata networks*, it is impractical to conduct an exhaustive search over all network sizes and connection organisations given the exponentially large number of possibilities. Therefore, the scope of this research is strategically narrowed. The study will utilize a fixed network size and focus on a subset of commonly observed random network topologies. By limiting the variety of topologies considered, the aim is to identify the network structures that most effectively support reservoir computing for text classification tasks. Furthermore, for simplicity matters, only binary classification tasks are considered.

*To identify the most effective transition rule*, the investigation will also address the challenge of identifying optimal transition rules for cellular automata. The scope is confined to an extension of elementary cellular automata rules applied on graphs. By restricting the ruleset, the study aims to systematically evaluate the impact of different transition rules on the performance of the cellular automata networks in a controlled manner. This approach allows for a focused analysis of how specific rules influence the dynamical properties and computational capabilities of the network.

*To develop a robust methodological framework for addressing such complex and open-ended challenges*, the primary goal of this research is not to find the highest-performing classifier but to establish a framework that identifies the optimal reservoir for the task. Consequently, the hyperparameters of the readout layer will remain constant throughout the experiments. This decision ensures that the influence of different reservoirs on system performance is assessed independently of the readout layer optimization. By maintaining consistent parameters, the study isolates the effects of network topologies and transition rules, thereby contributing to a robust methodology for future research in this domain.

Given these constraints, it is acknowledged that this study may not discover the optimal cellular automata network and reservoir computing system. However, identifying the best-performing configurations under the specified conditions will provide valuable insights into the key properties of effective cellular automata networks as a reservoir. These findings will lay a solid foundation for future research and broader applications.

### 1.4 Contributions

This study contributes several key innovations. First, an extension of elementary cellular automata rules is proposed to accommodate more than two neighbours,

generalizing their application from a grid structure to a graph structure. This broadens the scope and versatility of cellular automata in computational tasks. Additionally, a novel type of random network has been suggested to construct the cellular automata network. These networks exhibit low connectivity density and when implemented with the extended elementary cellular automata rules, they perform better than their more frequent counterparts such as Erdős-Rényi networks, small-world networks, and scale-free networks. Furthermore, implementing a reservoir computing system with cellular automata networks is by itself a novel innovation.

## 1.5 Thesis Structure

This thesis is organized into seven main chapters. Chapter 1 introduces the research problem, objectives, limitations, and contributions of the thesis. Chapter 2 presents the background on reservoir computing, cellular automata, network theory, and information theory, including discussions on physical reservoirs, the readout layer, and the use of cellular automata networks as reservoirs. Chapter 3 outlines the reservoir computing framework, detailing the input, reservoir, and readout layer components, while introducing the various network topologies and transition rules to be examined. Chapter 4 describes the methods employed for classification and rule entropy analysis within the reservoir computing framework. Chapter 5 presents the experimental results, discussing the performance of different network topologies and transition rules. Chapter 6 interprets the results, discussing the selection of network topologies and transition rules, the evaluation framework, and other related topics, while also proposing future research directions. Finally, Chapter 7 summarizes the key findings and highlights the contributions of the research. The thesis is supported by an appendix with complete experiment results.

# 2

## Background

This chapter covers key concepts and theories related to reservoir computing, cellular automata, network theory, and information theory. By establishing this groundwork, the aim is to contextualize the study and highlight the significance of the chosen methodologies and approaches.

### 2.1 Reservoir Computing

Reservoir computing is a machine learning method, a type of neuromorphic computing centred around the concept of universal spontaneous computation implemented via a *reservoir* [9]. The reservoir is realised as a high dimensional dynamical system with a rich internal state space that evolves over time and interacts with its environment [10]. The core idea is that such dynamical systems are inherently capable of computation and information processing [11], [12]. Specifically, the computational power of the dynamical system is assessed by observing the evolution of its state space in response to external inputs. Ultimately, the state of the system is transferred to a simple, linear readout layer that generates the output.

The primary applications of reservoir computing lie in classification and prediction tasks. Since the reservoir is realised by dynamical systems that evolve over time, this approach excels at processing time-dependent and sequential data. Key areas of application include time series prediction, speech recognition, signal processing, bioinformatics, and pattern recognition. These diverse applications highlight the versatility and robustness of reservoir computing in handling complex real-world data.

The concept of reservoir computing emerged as a response to the challenges associated with training neural networks, and it was developed independently by two different research groups. The training process of recurrent neural networks faces several problems, including vanishing and exploding gradients [13], overfitting, high computational costs, and extensive data requirements [14]. To address these issues, the idea to entirely forego training of the network, except for the readout layer, was proposed. This led to the introduction of two novel systems: Echo State Networks [15], [16], which are randomly initiated recurrent neural networks that possess the echo state property, and Liquid State Machines [17], [18], which utilize randomly connected spiking neural networks.

Since the introduction of reservoir computing, the concept of the reservoir component has been generalized to incorporate any arbitrary dynamical system with specific key properties. Firstly, the reservoir should be high-dimensional to project the input into a complex state space, allowing for detailed representations. Secondly, its dynamics should be rich, enabling the separation of similar inputs within the state space. Additionally, the elements within the reservoir must exhibit nonlinear behaviour to effectively process complex patterns in the input data. Lastly, the reservoir should also possess the echo state property, which means the system has a fading memory where the influence of past inputs gradually diminishes.

Within the context of reservoir computing, it is widely believed that optimal performance is achieved when the reservoir operates at the *edge of chaos* — a critical transition point between order and chaos. Research has shown that dynamical systems can perform complex computations most effectively at this boundary, where the system balances stability and adaptability, enabling efficient information processing [19]. Consequently, reservoirs functioning at the edge of chaos are considered highly desirable for maximizing computational capabilities.

### 2.1.1 Physical reservoirs

A primary appeal of reservoir computing lies in the ability to leverage physical reservoirs. A classic study demonstrated how waves on the surface of a bucket of water could function as a Liquid State Machine for speech recognition tasks, showing that a simple readout layer suffices for classification [20]. Although the practicality of that particular reservoir was not ideal, significant advances have been made in various other fields of physical reservoir computing [21]. Concrete examples of physical reservoirs that have been explored include memristors [22], photonic systems [23], spintronic devices [24], biological systems [25], and mechanical systems [26], [27].

These physical reservoirs offer unique advantages due to their inherent properties. For instance, memristors can emulate synaptic plasticity, photonic systems benefit from the speed of light for high-speed data processing, spintronic devices exploit electron spin for energy-efficient computations, biological systems harness the complex dynamics of neural tissue, and mechanical systems leverage the non-linear dynamics of physical materials.

Simulating physical reservoirs is also a common practice, especially in the early stages of developing reservoir computing systems. Examples include using neural networks, cellular automata, or Boolean networks. This approach is often more practical during the initial development phases of reservoir computing, as it is more cost-efficient and simpler to implement. Simulations allow researchers to test and refine their models before deploying physical systems, which can be more expensive and complicated to build.

### 2.1.2 The readout layer

The role of the readout layer in reservoir computing is not to perform the main computation; rather, this computation is carried out by the reservoir itself. The readout layer assesses the state of the reservoir and maps these states to output labels or values [9]. Ideally, the readout layer should be kept as simple as possible, often involving a linear transformation of the reservoir's state space.

The weights of the readout layer are the only parts of the system that require training. Since the readout layer is simple, it can be trained with a relatively small amount of data, making the training process fast and resource efficient. This simplicity and efficiency in training are among the primary benefits of using reservoir computing. It allows for reduced computational costs and time compared to traditional methods, which is particularly advantageous when working with large datasets or when rapid training is required.

Furthermore, the simplicity of the readout layer contributes to the overall robustness and flexibility of the reservoir computing framework. It can easily adapt to different types of tasks, such as classification, regression, and signal processing, without the need for complex training algorithms. This versatility, combined with the inherent efficiency of the training process, underscores the practical utility of reservoir computing in various real-world applications.

## 2.2 Cellular Automata

Cellular Automata are simple, yet powerful mathematical models used to represent physical, biological, and computational systems. Initially proposed by John von Neumann [28] and extensively studied by Stephen Wolfram [29], [30], cellular automata consist of a grid of cells that interact with their neighbours according to predefined local transition rules. The state of the cells updates synchronously in discrete time steps. Since the cells only interact with their local neighbourhood, the system is decentralised, with no central controller. Despite the simplicity, cellular automata can exhibit complex dynamics and emergent behaviours from the interaction of simple rules. This makes them valuable for modelling a wide range of real-world systems.

A notable example of cellular automata that produce complex behaviour from simple rules is Conway's Game of Life (for an introduction see [31] and references therein), which has been demonstrated to simulate a Turing machine [32], a fundamental model of computation capable of performing any calculation that a digital computer can. This capability to simulate a Turing machine highlights the immense computational power inherent in cellular automata. It underscores their potential as computational devices capable of complex data processing and algorithm execution. The Game of Life, thus, serves as a powerful example of how simple, local interactions can lead to complex, global behaviours, making cellular automata a valuable tool in both theoretical and applied computational research.

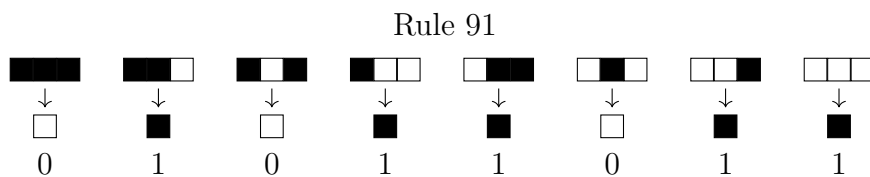


Figure 2.1: Visual representation of elementary cellular automata rule 91.

### 2.2.1 Elementary Cellular Automata

Elementary cellular automata are the simplest class of cellular automata, defined on a one-dimensional grid [33], [34]. Each cell interacts with its immediate left and right neighbours. The rule can be seen as a Boolean function that takes three bits  $x_1$ ,  $x_2$ , and  $x_3$  as input and produces a single bit as output, denoted as  $y = r(x_1, x_2, x_3)$ . A simple combinatorial argument leads to the conclusion that one can construct  $2^{2^3} = 256$  such rules in total. Such a rule explains how to update a cell that is in the state  $x_2$  that is influenced by two neighbouring cells in states  $x_1$  and  $x_3$ . The variable  $y$  is the new state of the cell after the update. Assume that the automaton of interest is made of  $N$  cells  $c_1, c_2, \dots, c_{i-1}, c_i, c_{i+1}, \dots, c_N$ . Then a cell  $c_i$  is updated according to the rule

$$c_i(t+1) = f(c_{i-1}(t), c_i(t), c_{i+1}(t)) \quad (2.1)$$

These rules are named based on the binary representation of their outputs. For example, if the resulting sequence of  $f(1, 1, 1), f(1, 1, 0), \dots, f(0, 0, 0)$  is the binary sequence  $01011011_2$ , as illustrated in Figure 2.1, where the black and white cells represents 1 and 0, respectively. Then, this function  $f$  corresponds to rule 91, since  $01011011_2 = 91_{10}$ . A specific mapping from a particular input to an output for  $f$  can also be referred to as  $c_{i-1}(t), c_i(t), c_{i+1}(t) \rightarrow c_i(t+1)$ . Then, the first mapping in rule 91 is written as  $111 \rightarrow 0$ .

Elementary cellular automata can be categorized based on the patterns they produce over time when starting from a random initial state. Stephen Wolfram introduced four classes of behaviour [33]:

1. Rules that converge to a uniform state.
2. Rules that converge to repetitive or stable states.
3. Rules that produce seemingly random states.
4. Rules that produce complex structures, including both repetitive and stable areas, as well as regions of complicated behaviour.

Class 4 is particularly interesting because it demonstrates a balance between order and chaos, where some structures are stable while others exhibit complex, dynamic behaviours. Notably, rule 110, a member of Class 4, has been proven to be Turing complete [35], meaning it can be used to perform any computation that a universal Turing machine can. This highlights the impressive computational capabilities of even the simplest cellular automata.

## 2.2.2 Cellular Automata Networks

Cellular Automata Networks (CAN) extend traditional cellular automata by replacing the standard grid structure with a graph structure [36]. In the literature, this concept is also known as *graph cellular automata* [36], *cellular automata on graphs* [37], or simply CAN [38]. In this setup, each cell corresponds to a node in a graph, where the neighbourhood of a cell is defined by edges connected to the node in the graph, rather than proximity on a grid.

CAN can be defined for both undirected and directed graphs. In the former case, the transition rule of the cellular automata functions similarly to traditional cellular automata, with connected nodes as neighbours and the degree as the neighbourhood size. In the latter case, it is important to distinguish between the in-degree and out-degree of a node to define the neighbourhood for a cell accurately. The neighbourhood corresponds to cells connected through incoming edges, and the neighbourhood size equals the in-degree of the node. To avoid confusion, this type of neighbour will be referred to as a *predecessor*. Similarly, cells connected by outgoing edges are referred to as *successors*. Importantly, the transition rule only considers predecessors to determine the next state of a cell.

This adaptation allows for the use of cellular automata to model systems that deviate from the grid structure. By incorporating the flexibility of network topologies, CANs are particularly suited for simulating dynamics on networks found in the real world, such as social networks, transportation networks, and interconnected systems in biology and finance. This network-based approach offers a richer framework for exploring the dynamics of complex systems, where the structure of connections can significantly influence emergent behaviour.

## 2.2.3 Cellular Automata Networks as a Reservoir

Before discussing cellular automata networks, it is important to note that traditional cellular automata have been successfully employed or discussed as a reservoir in various studies [10], [39]–[42]. In these studies, elementary cellular automata were primarily utilized. Additionally, a key paradigm of reservoir computing is that the readout layer should only access the current state of the reservoir. However, when cellular automata are used, it is common practice for the readout layer to access the entire evolution of the cellular automata, or at least the recent history of the evolution.

Despite the success of traditional cellular automata, the utilization of CAN as a reservoir has not been extensively studied. In the context of reservoir computing, CAN was first introduced to model a memristor network, which was then used as a reservoir to classify protein toxicity [43]. Consequently, the use of CAN remains a novel approach in the field of reservoir computing.

## 2.3 Network Theory

Network theory has been extensively studied, and for the purpose of this thesis, only a few key concepts are pertinent. Networks can model real-world systems, and the structure of the connections between entities in these systems varies depending on the specific system. The connection structure is commonly known as the network's topology and there are three network types that are particularly relevant for this study.

The random graph, first introduced by Erdős and Rényi [44] is generated by creating an edge between each pair of nodes with a certain probability  $p$ , resulting in what is known as an Erdős-Rényi graph. When discussing random networks, this type of graph is what one usually has in mind.

Building on the work of Erdős and Rényi, Watts and Strogatz introduced the small-world network [45]. Recognizing that few real-world systems are completely random, this topology aims to balance regularity and randomness. It is constructed by starting with a regular network where each node is connected to its nearest neighbours. Subsequently, each edge is rewired to a randomly chosen node with a probability  $q$ , thereby creating long-distance connections.

Similarly, the scale-free network was introduced by Barabási and Albert [46] to address the limitations of random networks in modelling certain real-world systems. In a scale-free network, the likelihood of a node acquiring new connections is proportional to its existing degree. Thus, nodes with higher degrees are more likely to attract additional edges. This preferential attachment process results in networks characterized by a few highly connected hubs and many nodes with lower degrees.

These network types — random, small-world, and scale-free — provide a foundational framework for understanding the structural properties of complex networks. The topology of these networks offers unique insights into the dynamics and connectivity patterns that can emerge in real-world systems.

## 2.4 Information Theory

Information theory, introduced by Claude Shannon [47] provides a mathematical framework for quantifying information. This theory is fundamental in understanding and optimising communication systems, computational devices, and many other applications in modern technology. In this thesis, information theory is used to analyse the efficiency and capabilities of different transition rules on specific networks.

One of the core concepts in information theory is *Shannon entropy*. Entropy measures the uncertainty or unpredictability associated with a random variable. For a discrete random variable  $Y$  with possible outcomes  $\{y_1, y_2, \dots, y_n\}$ , the Shannon entropy  $H(Y)$  is defined as:

$$H(Y) = - \sum_{i=1}^n p(Y = y_i) \log_2 p(Y = y_i)$$

In this equation,  $p(Y = y_i)$  represents the probability of the outcome  $y_i$ .

Shannon entropy indicates how much information is needed, on average, to describe the state of a random variable. Higher entropy indicates greater unpredictability, while lower entropy signifies more predictability. For example, a fair coin toss, with binary outcomes {Heads, Tails} each having a probability of 0.5, has a Shannon entropy of:

$$H(X) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1 \text{ bit.} \quad (2.2)$$

In contrast, a biased coin that always lands on heads has zero entropy, as there is no uncertainty in the outcome.



# 3

## Reservoir Computing Framework

This chapter introduces the reservoir computing framework and details the implementation and parameter choices of classification models in this study. The chapter is organized into four sections. The first section describes the general architecture of the system. The second section provides a detailed description of the input text structures. The third section introduces the reservoir, along with the variations in network topologies and transition rules evaluated in this study. Finally, the fourth section presents the concrete implementation of the readout layer.

### 3.1 General Architecture

The architecture of the models adheres to standard reservoir computing practices when cellular automata is utilised as the reservoir. The key difference is the substitution of cellular automata with CAN. However, this exchange does not significantly alter the general architecture due to the modular nature of the reservoir computing framework. Figure 3.1 provides an illustration of the model, highlighting its components and their interactions. Note that a CAN with directed edges are used.

In this architecture, the input sequence is encoded as a vector and interacts with the reservoir through input nodes. Specifically, each input node is randomly connected to a single cell in the reservoir as a predecessor, and its order as a predecessor

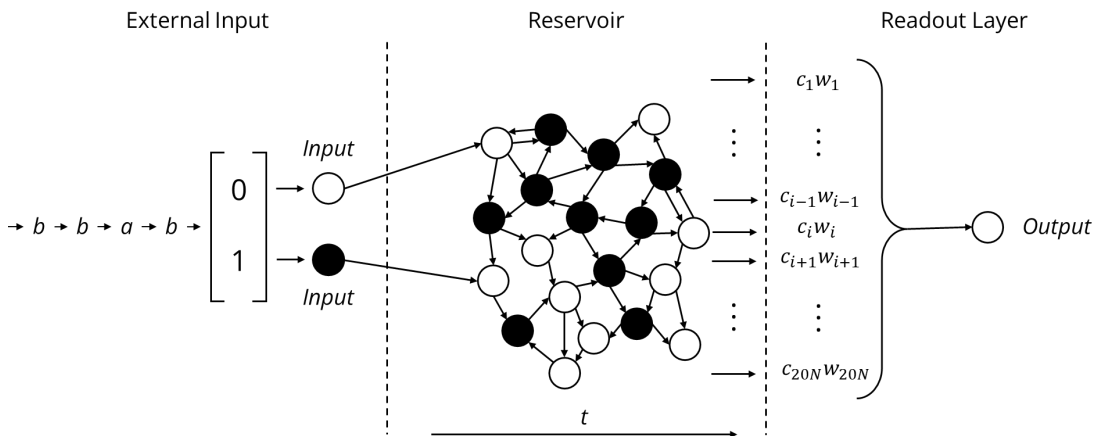


Figure 3.1: An overview of the model's architecture.

is also randomized. Alternative approaches include driving certain cells in the reservoir directly with the input sequence or allowing an input node to have multiple connections to the reservoir. The effects of such alternative implementations have not been studied in this research.

The readout layer has access to more than just the current state of the reservoir. Specifically, the readout layer receives the weighted sum of the states from the last 20 discrete time steps of the reservoir’s evolution,  $w_1c_1 + \dots + w_{i-1}c_{i-1} + w_ic_i + w_{i+1}c_{i+1} + \dots + w_{20N}c_{20N}$ , where  $c_i$  is the state of the  $i$ -th cell in the evolution,  $w_i$  is the weight and  $N$  is the network size. In this study  $N = 20$  is used. The entire evolution spans 100 time steps; however, to allow the system to transition away from its initial random state, only the latter part of the evolution is used. The 20 last time steps are used, given that  $N = 20$ , ensuring the ability to track the impact of a single input on each cell within the networks regardless of topology. Additionally, given that the CANs considered in this study are relatively small, the evolution of each cell within the CAN is transferred to the readout layer. An alternative approach would be to transfer the evolution of only a subset of cells, but this option was not explored in this research.

## 3.2 Input structure

The input consists of phrases with varying text structures. To construct these phrases, the simplest possible non-trivial alphabet,  $\mathcal{A} = \{a, b\}$ , comprising only two letters, is used. Thus, each input phrase is represented as a sequence of  $a$  and  $b$  characters.

Prior to being processed by the reservoir, each letter in the input sequence undergoes one-hot encoding. This encoding scheme assigns a unique binary vector to each letter. Since, there are two letters, the vector’s length is also two, with

$$a = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

One input node is then assigned to each position in the vector, corresponding to each letter in  $\mathcal{A}$ . Specifically, an input node is activated, i.e., state set to 1, if the current letter in the input sequence matches the corresponding letter in  $\mathcal{A}$ ; otherwise, the node remains inactive, i.e., state set to 0. Consequently, the system is equipped with two input nodes. Although a single input node could suffice due to the binary nature of the input, one-hot encoding is employed for its simplicity and scalability to larger alphabets. This approach ensures that each unique letter is distinctly represented by a separate input node, facilitating a more intuitive input representation for the system.

To introduce structure to the input sequence, each phrase is generated by a Markov model. The four Markov models considered in this study are displayed in Figure 3.2. Each Markov process initiates in State 0. As the system transitions to new states, letters are generated and appended to the phrase. Figure 3.2a represents purely random text, where the letters  $a$  and  $b$  have equal probabilities of occurrence. Figure 3.2b also represents random text; however, the letter  $a$  is more likely to

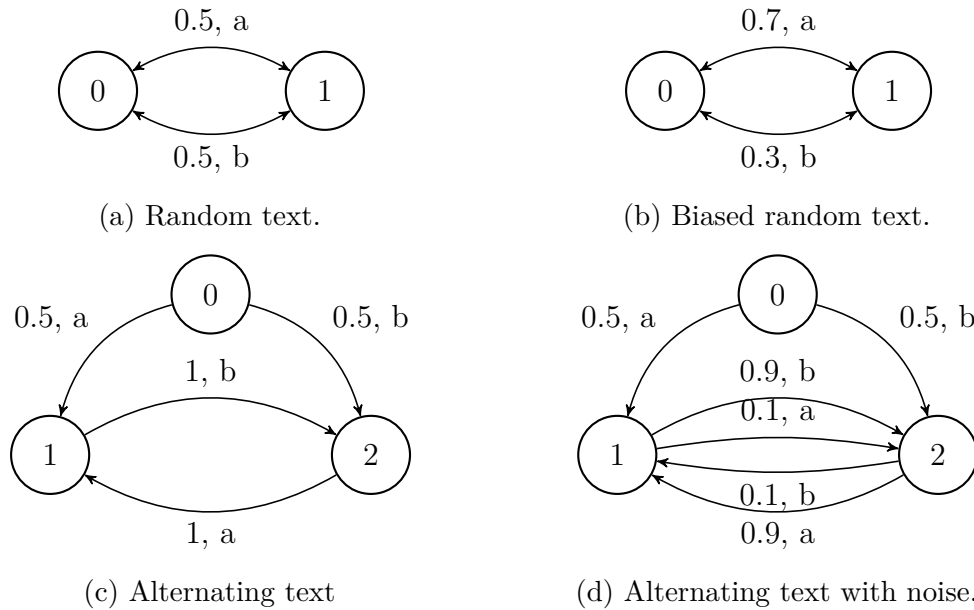


Figure 3.2: Markov models for generating text phrases with different text structures.

appear than  $b$ , with probabilities  $p(a) = 0.7$  and  $p(b) = 0.3$ . Figure 3.2c generates text that alternates between  $a$  and  $b$ , although the initial letter in the phrase is random. Similarly, Figure 3.2d also generates alternating text, but this time with noise. Noise is introduced with a small probability  $p(\text{noise}) = 0.1$  and consists of the opposite letter in  $\mathcal{A}$ . These Markov chains produce phrases exhibiting some of the most fundamental properties of text, providing a basis for evaluating the computational capabilities of cellular automata networks in text classification tasks. Specifically, in these tasks, the random text is compared to input with varying letter frequency, high correlation, or correlation with noise present.

### 3.3 Reservoir

A range of CAN with different combinations of network topologies and transition rules will be considered for the reservoir. In this study, a unique pair of network topology and transition rule will be referred to as a configuration. Consistent with the first two objectives of this thesis — to identify the optimal network topology and transition rule — all other hyperparameters are kept constant. Specifically, the network size is limited to 20 cells, excluding the input nodes. This relatively small network size is deliberately chosen to enhance the training efficiency of the classifier and to showcase the performance capabilities of even the simplest models.

The reservoir operates through a straightforward mechanism. At each discrete time step, the states of the cells are synchronously updated according to the defined transition rule. Input nodes, however, are updated based on the incoming input sequence rather than the transition rule. This iterative update process allows the system to evolve over time, and it is this dynamic evolution of cell states that performs the core computational tasks within the reservoir.

### 3.3.1 Network Topology

This study will examine four distinct types of network models and evaluate the impact of different transition rules on each model. Three standard network models are employed, specified as Algorithm 1, Algorithm 2, and Algorithm 3. These algorithms are adaptations from previous work on reservoir computing with memristor networks [43] and ensure the formation of weakly connected directed graphs. Additionally, a specialized model developed specifically for this study is defined as Algorithm 4.

Algorithm 1 generates networks based on the Erdős-Rényi (ER) model [44]. These networks consist of a predetermined number of nodes  $n$  with each pair of nodes connected by an edge with a fixed probability  $p$ . Let the number of edges be defined as  $d$ . To ensure network connectivity, the algorithm deviates from the classical ER framework by initially constructing a tree structure, to which additional edges are randomly added. This method guarantees connectivity and allows for precise control over the total number of edges. Thus, guaranteeing a network density  $p$ . This approach facilitates more equitable comparisons between different networks, while retaining the fundamental characteristics of the ER model: edges can be created between any two nodes, and the creation of edges remains independent. Furthermore, the network includes driven input nodes, whose states are regulated by an input sequence. These driven nodes are added at random after the network has been created and are unique in that they can only serve as predecessors to other nodes and are not themselves successors to any node.

---

**Algorithm 1** Weakly connected directed Erdős-Rényi model

---

```
1: Create a node in a random state.
2: for  $i = 1, 2, \dots, n - 1$  do
3:   Create a node in a random state.
4:   Create a directed edge from the last node created to a random existing node
   with random orientation.
5: end for
6: while  $d < p \cdot n(n - 1)$  do
7:   Select two random nodes  $a$  and  $b$ .
8:   if There exists no edge from  $a$  to  $b$  then
9:     Create an edge from  $a$  to  $b$ .
10:  end if
11: end while
```

---

Algorithm 2 generates networks based on the Watts-Strogatz (WS) model [45], commonly known as small-world networks. These networks comprise a predetermined number of nodes  $n$ , with each node initially connected to its  $k$  nearest neighbors, where  $k$  is an even number. Thereafter, each edge is rewired with a fixed small probability  $q$ , reassigning the second node of the edge to a randomly chosen node while avoiding self-loops and duplicate edges, thus creating long-distance connections. In a traditional small-world network, the number of edges  $d$  is a multiple of  $n$ . To allow  $d$  to be any number, the algorithm has been modified by adding more edges to the initial regular structure until the desired edge density  $p$  is achieved. This

---

**Algorithm 2** Weakly connected directed Watts-Strogatz model

---

```

1:  $k = 2\lfloor p(n-1) \rfloor$ 
2: Compute the missing edges  $m = p \cdot n(n-1) - n \cdot k/2$ 
3: Create  $n$  nodes in random states.
4:  $k = k + 2$ 
5: for  $i = 1, 2, \dots, n$  do
6:   if  $i = m + 1$  then
7:      $k = k - 2$ 
8:   end if
9:   for  $j = 1, 2, \dots, k/2$  do
10:    Create an edge from node  $i$  to node  $i + j \bmod n$  with a random direction.
11:    With probability  $q$  rewire the edge into a long-distance connection to a
    random node.
12:   end for
13: end for

```

---

modification ensures that the algorithm produces a network that corresponds exactly to the classical WS network when  $p(n-1)$  is an integer. Otherwise, the network includes some additional edges. Similarly to the previous algorithm, driven input nodes are added to the network after it has been generated.

Algorithm 3 generates networks based on the Barabási-Albert (BA) model [46]. The algorithm for generating BA networks proceeds as follows: initially, a small number of nodes are established. At each subsequent time step, a new node is introduced along with a fixed number of edges connecting it to the existing nodes. However, this attachment process is not uniform; the probability that a new node will connect

---

**Algorithm 3** Weakly connected directed Barabási-Albert model

---

```

1: if  $p < 0.25$  then
2:    $n_0 = \left\lfloor \frac{n+1 - \sqrt{(n+1)^2 - 4(p \cdot n(n-1) + 1)}}{2} \right\rfloor$ 
3:    $h = n_0$ 
4: else
5:    $n_0 = \left\lfloor \frac{2n+1 - \sqrt{(2n+1)^2 - 8(p \cdot n(n-1) + 1)}}{4} \right\rfloor$ 
6:    $h = 2n_0$ 
7: end if
8:  $t = n - n_0$ 
9: Create  $n_0$  nodes in a random state.
10: for  $i = 1, 2, \dots, t$  do
11:   Create a node in a random state.
12:   Create  $h$  edges from the last node created to random existing nodes with
    random orientation.
13: end for

```

---

to an existing node is proportional to the number of edges that the existing node already possesses, following a preferential attachment mechanism. The model is characterized by three parameters: the initial number of nodes  $n_0$ , the number of edges added with each new node  $h$ , and the total number of time steps  $t$ . For the purposes of this study, these parameters are represented as the number of nodes  $n$  and edge density  $p$ , as detailed in [43]. Like the previous algorithms, driven input nodes are incorporated into the network after it has been generated.

Algorithm 4 is not derived from a standard network model. Instead, it is characterized by a fixed number of predecessors for each node. The algorithm generates the network by iterating over all  $n$  nodes and assigning  $\alpha$  unique predecessors to each node. Consequently, the edge density  $p$  is constant for a given  $\alpha$ . This approach produces a network with properties like those of an ER network, but with a predetermined number of predecessors for each node. Additionally, driven input nodes are incorporated into the network. Unlike the previous algorithms, which add new edges for the driven nodes, this algorithm randomly replaces existing predecessors with driven nodes, thereby maintaining a constant number of predecessors for each node. Additionally, while this algorithm does not guarantee weak connectivity, the probability that the network is not weakly connected is insignificant. Moreover, the reservoir’s ability to compute does not depend on weak connectivity. In this study, this network model is referred to as the Fixed Predecessor model (PD- $\alpha$ ), where  $\alpha$  denotes the number of predecessors.

---

**Algorithm 4** Directed Fixed Predecessor (PD- $\alpha$ ) model

---

- 1: Create  $n$  nodes in random states.
  - 2: **for**  $i = 1, 2, \dots, n$  **do**
  - 3:     Randomly assign  $\alpha$  predecessors to node  $i$ .
  - 4:     Create directed edges from all predecessors to node  $i$ .
  - 5: **end for**
- 

#### 3.3.2 Transition Rule

The transition rule is applied uniformly across each cell in the network, enabling the isolated observation of a single rule’s behaviour. This rule is an adaptation of Wolfram’s elementary cellular automata rules, extended to accommodate cells with more than two predecessors while retaining the simplicity of the two-neighbour approach. The chosen method integrates arbitrary network topologies with Wolfram’s elementary cellular automata rules through a straightforward procedure: when a cell has more than two inputs, a voting mechanism determines its new state. Specifically, the rule is applied to all possible pairs of predecessors, and the cell’s new state is decided by a majority vote of these outcomes. Exceptions are made for cells with fewer than two predecessors. If a cell is not connected to any other cell by an incoming edge, its state remains unchanged. If a cell has only one predecessor, it adopts the state of that cell.

For example, consider a cell  $c_1$  that receives input from a list of predecessors  $\mathcal{P} = \{c_2, c_3, \dots, c_n\}$ . It is important to note that for each list of predecessors,

a predetermined, randomized ordering unique to each target cell  $c_1$  is assumed. The Wolfram rule  $f$  is applied to each pair within  $\mathcal{P}$ , generating a list of outcomes  $f(c_i, c_1, c_j)$ , where  $c_i, c_j \in \mathcal{H}$  and  $i \neq j$ . Then, the majority outcome becomes the new state of  $c_1$ .

### 3.4 Readout Layer

The readout layer applies a linear transformation to the evolution of the reservoir. Then, the model employs the sigmoid activation function to convert the linear combination of input features into a value in  $(0, 1)$ . The sigmoid function is defined as follows:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

This function maps any real-valued number into the range  $(0, 1)$ , with  $\sigma(0) = 0.5$ , making it suitable for binary classification tasks. For an input vector  $\mathbf{x}$  with weights  $\mathbf{w}$  and a bias term  $w_{\text{bias}}$ , the prediction  $\hat{y}(\mathbf{x})$  is calculated using the following equation:

$$\hat{y}(\mathbf{x}) = \text{round} \left( \sigma \left( \sum_{i=0}^{n-1} w_i x_i + w_{\text{bias}} \right) \right)$$

Here,  $n$  represents the number of features in  $\mathbf{x}$ . Each weight  $w_i$  corresponds to a feature, and  $w_{\text{bias}}$  is the bias term added to the linear combination to adjust the decision boundary. If the weighted sum is positive the prediction becomes 1, otherwise it is 0. During the training phase, the weights are adjusted using the gradient descent algorithm to minimize the difference between the predictions and the actual labels. The update rule for the weights is given by:

$$w_j \leftarrow w_j + \frac{\alpha}{m} \sum_{i=1}^m (y^{(i)} - \hat{y}(\mathbf{x}^{(i)})) x_j^{(i)}$$

for  $j = 0, 1, \dots, n - 1$ , and

$$w_{\text{bias}} \leftarrow w_{\text{bias}} + \frac{\alpha}{m} \sum_{i=1}^m (y^{(i)} - \hat{y}(\mathbf{x}^{(i)}))$$

Here,  $\alpha$  is the learning rate,  $m$  is the number of training examples,  $\mathbf{x}^{(i)}$  is the input vector of the  $i$ -th example, and  $y^{(i)}$  is the actual label of the  $i$ -th example. Ultimately, the final prediction is determined by computing  $\hat{y}(\mathbf{x}_{\text{test}}^{(i)})$ , where  $\mathbf{x}_{\text{test}}^{(i)}$  is a sample from the test dataset.



# 4

## Methods

This chapter introduces the methods used to investigate the performance and functionality of different reservoirs. The chapter is organized into two sections. The first section details the text classification task, and the second section describes how information theory is applied to assess the system.

### 4.1 Classification

This study consists of multiple experiments, each comprising numerous classification tasks that follow the same procedure but with different combinations of network models, transition rules, and input text structures. Let  $\mathcal{N}_\gamma = \{N_1, N_2, \dots, N_{100}\}$  represent a set of networks generated by a network model  $\gamma$ , where  $\gamma \in \{ER, BA, WS, PD-2, PD-3, PD-4\}$  corresponds to the desired network topology. The target density of the *ER*, *BA*, and *WS* networks is 0.16, while the density of the *PD* networks are 0.11, 0.16, and 0.21 for increasing number of predecessors. Additionally, for the *WS* network a fixed probability  $q = 0.1$  is chosen for the long-distance connections. Moreover, let  $\mathcal{R} = \{R_0, R_1, \dots, R_{255}\}$  denote the set of transition rules, where  $R_i$  is the extension of elementary cellular automata rule  $i$  that is introduced in this study.

The aim is for the classifier to differentiate between input generated by different text structures  $\alpha$  and  $\beta$ . Suppose that  $\mathcal{I}_{\alpha,\beta} = \{I_1^{\alpha,\beta}, I_2^{\alpha,\beta}, \dots, I_n^{\alpha,\beta}\}$  represents a set of input text sequences for the classifier, where  $I_i^{\alpha,\beta}$  is an input sequence of either type  $\alpha$  or  $\beta$ . Here,  $\alpha$  and  $\beta$  define the type of text structure an input instance can be generated by, with  $\alpha, \beta \in \{a, b, c, d\}$ , corresponding to the Markov chains shown in Figure 3.2. Specifically,  $a$  is random text,  $b$  is biased text,  $c$  is alternating text, and  $d$  is alternating text with noise.

To analyse the performance of the various configurations, two stochastic variables are introduced which are associated with the choices of  $\mathcal{N}_\gamma$ ,  $\mathcal{I}_{\alpha,\beta}$ , and  $\mathcal{R}$ . Given the objective to differentiate between rules, each rule  $R_i$  in  $\mathcal{R}$  is considered individually. The first variable,  $\mu(\mathcal{N}_\gamma, \mathcal{I}_{\alpha,\beta}, R_i)$ , denotes the mean classification accuracy for a given combination of network model, input text structure, and transition rule. The second variable,  $\sigma^2(\mathcal{N}_\gamma, \mathcal{I}_{\alpha,\beta}, R_i)$ , represents the standard deviation of the classification accuracy. This variable quantifies the variability or spread of the individual classification accuracies across the experiments, providing insight into the consistency of the performance for each configuration.

To obtain  $\mu(\mathcal{N}_\gamma, \mathcal{I}_{\alpha,\beta}, R_i)$  and  $\sigma^2(\mathcal{N}_\gamma, \mathcal{I}_{\alpha,\beta}, R_i)$ , the following procedure is repeated 100 times:

1. Generate a network based on the network model  $\gamma$ .
2. For each transition rule  $R_i$ , use the network and transition rule as the reservoir in a reservoir computing context.
3. Train the system on a training dataset  $\mathcal{I}_{\alpha,\beta}$ .
4. Test the system on a test dataset  $\mathcal{I}'_{\alpha,\beta}$ .
5. Record the prediction accuracy for  $R_i$  on the network, as the number of correct prediction divided by the number of total predictions.

After creating 100 unique graphs,  $\mu(\mathcal{N}_\gamma, \mathcal{I}_{\alpha,\beta}, R_i)$  is computed as the mean prediction accuracy for the rule  $R_i$ , and  $\sigma^2(\mathcal{N}_\gamma, \mathcal{I}_{\alpha,\beta}, R_i)$  is calculated as the standard deviation of the accuracy across different graphs.

In this framework, the efficiency of rule  $R_i$  is characterized by the metrics  $\mu(\mathcal{N}_\gamma, \mathcal{I}_{\alpha,\beta}, R_i)$  and  $\sigma^2(\mathcal{N}_\gamma, \mathcal{I}_{\alpha,\beta}, R_i)$ , which indicate average performance and performance consistency, respectively, across the network ensemble within the specified topology and input type. For simplicity, the notation  $\mathcal{I}$  will be used instead of  $\mathcal{I}_{\alpha,\beta}$  when representing the mean and standard deviation for a network model and rule across all input types.

The scope of the experiments can be summarized as follows. Numerous small models are created. In a single experiment,  $256 \cdot 100 = 25600$  reservoir computing models are trained and evaluated. In this study, a total of  $6 \cdot 3 = 18$  experiments are conducted. This means that over all experiments, a total of  $25600 \cdot 6 = 460800$  models are create and  $256 \cdot 6 = 1536$  different model configurations are evaluated. The hyperparameters used for the training of the readout layer include a training dataset size of 400, a test dataset size of 100, a learning rate of 0.01, and 1000 passes over the training dataset.

## 4.2 Rule Entropy

In addition to the experiments, the information gain of predecessors in a single cell will also be computed for all configurations of network topology and transition rules. This type of assessment of CAN has been conducted in previous studies [37]. Information gain is defined as the difference between the entropy of a variable  $Y$  and the conditional entropy of  $Y$  given a variable  $X$ . Here,  $Y$  is defined as the next state of a single cell, and  $X$  is defined as the current state of either the cell itself or one of its predecessors. This is also known as *mutual information*. The computation aims to better understand the effects of different configurations when certain parts of the system are known and to highlight important properties of CAN. It is also utilized to categorise transition rules with similar behaviour.

In information theory, the entropy [47] of the binary variable  $Y$  is calculated as:

$$H(Y) = - \sum_{y \in \{0,1\}} p(Y = y) \log_2 p(Y = y)$$

where  $p(Y = y)$  is the probability of  $Y$  being  $y$ . Similarly, the conditional entropy of  $Y$  given the binary variable  $X$  is calculated as:

$$H(Y|X) = \sum_{x \in \{0,1\}} p(X = x) \left( - \sum_{y \in \{0,1\}} p(Y = y|X = x) \log_2 p(Y = y|X = x) \right)$$

where  $p(Y = y|X = x)$  is the conditional probability of  $Y$  being  $y$  given  $X = x$ , and  $p(X = x)$  is the probability of  $X$  being  $x$ . Finally, the information gain from knowing  $X$  is computed as:

$$IG(X) = H(Y) - H(Y|X)$$

This measures the reduction in uncertainty, i.e., the number of bits needed to describe the outcome, for  $Y$  when  $X$  is known. For the purposes of this study, it is assumed that all possible current states are equally likely. This simplification facilitates a more straightforward computation of the conditional entropy and, subsequently, the information gain. Therefore, the calculations can be further simplified as:

$$H(Y|X) = \frac{1}{2}H(Y|X = 0) + \frac{1}{2}H(Y|X = 1)$$

In this study, the outcome consists of a single bit. This means that the maximum possible entropy is  $H(Y) = 1$ , indicating that the outcome is equally likely to be 0 or 1. Conversely, the minimum entropy is  $H(Y) = 0$ , which means that the outcome is always the same. Consequently,  $IG(X) = 1$  implies that both outcomes were equally likely before, but with the additional knowledge of  $X$ , the outcome becomes certain. In contrast,  $IG(X) = 0$  indicates that the knowledge of  $X$  does not provide any additional information about the outcome.



# 5

## Results

This chapter presents the results from the text classification tasks, organized into three distinct sections, each offering a unique perspective on the findings. The first section examines the impact of different network topologies on prediction accuracy. The second section explores how various transition rules affect prediction accuracy. Finally, the third section details the information gain associated with different combinations of network topologies and transition rules.

### 5.1 Network Topology Performance

The overall performance of different network models on the text classification tasks is summarized in Figure 5.1. The average prediction accuracy,  $\mu(\mathcal{N}_\gamma, \mathcal{I}, R_i)$ , is calculated for each combination of network model and transition rule across all input types. The results are then grouped by  $\gamma$ , and within each group, the transition rules are ordered from worst to best, based on their prediction accuracy. This implies that

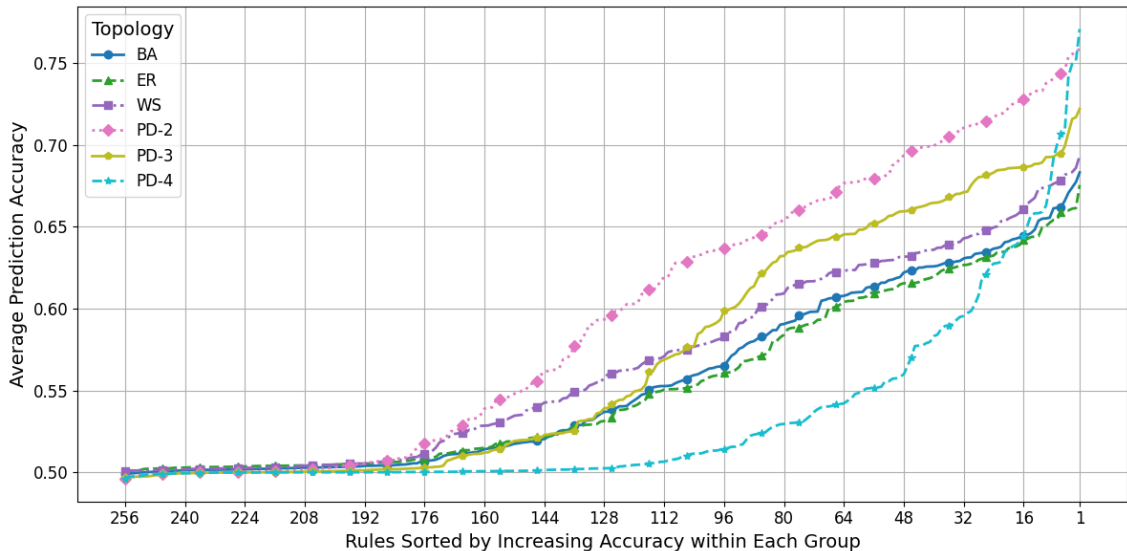


Figure 5.1: Average performance  $\mu(\mathcal{N}_\gamma, \mathcal{I}, R_i)$  over all experiments for each rule  $R_i$ ,  $i \in \{0, 1, \dots, 255\}$ , grouped by network model  $\gamma$ . The rules are sorted from worst to best performance within each group, with the x-axis representing the rank of each rule within its respective group.

the  $i$ -th best-performing rule for each  $\gamma$  does not have to be the same rule across all groups. Instead, the  $i$ -th position within each group represents the rule with the  $i$ -th best performance among rules applied to that specific network model. Moving forward,  $\mathcal{N}_\gamma$  is used instead of  $\gamma$  to emphasize that a set of network instances of the network model  $\gamma$  are utilised in all experiments, rather than a single instance of  $\gamma$ .

The analysis of the graph reveals a distinct hierarchy in the performance of different  $\mathcal{N}_\gamma$ . The highest overall performance is observed for  $\mathcal{N}_{PD-2}$ , followed by  $\mathcal{N}_{PD-3}$ ,  $\mathcal{N}_{WS}$ ,  $\mathcal{N}_{BA}$ ,  $\mathcal{N}_{ER}$ , respectively. An exception is noted for  $\mathcal{N}_{PD-4}$ , where the prediction accuracy does not follow the same trend as the other network models. While  $\mathcal{N}_{PD-4}$  exhibits poor overall performance, it demonstrates excellent peak performance compared to other network models. Additionally, it is evident that for each  $\mathcal{N}_\gamma$ , the number of transition rules  $n$  that fail to provide meaningful computation, indicated by  $\mu(\mathcal{N}_\gamma, \mathcal{I}, R_i) \approx 0.50$ , is approximately the same across each  $\mathcal{N}_\gamma$ , with  $n \approx 75$ .

The observed results indicate that the most promising performance with the proposed transition rules is found in network models with a fixed number of predecessors, specifically  $\mathcal{N}_{PD-2}$ ,  $\mathcal{N}_{PD-3}$ , and  $\mathcal{N}_{PD-4}$ . Therefore, for the sake of simplicity, the primary focus will be on these network models moving forward. The full results for  $\mathcal{N}_{BA}$ ,  $\mathcal{N}_{ER}$ , and  $\mathcal{N}_{WS}$  can instead be found in Appendix A, Table A.2.

Moreover, network models with more than four predecessors do not appear to offer any additional computational capabilities, as demonstrated in Figure 5.2. The experiment focuses exclusively on the best performing rule,  $R_{91}$ , for  $\mathcal{N}_{PD-4}$ , as this rule seems to exhibit properties that are well-suited for network models with more than two predecessors. Additionally, the investigation only considers one type of input,  $\mathcal{I}_{a,d}$ . Given these limitations compared to the original experiments, the

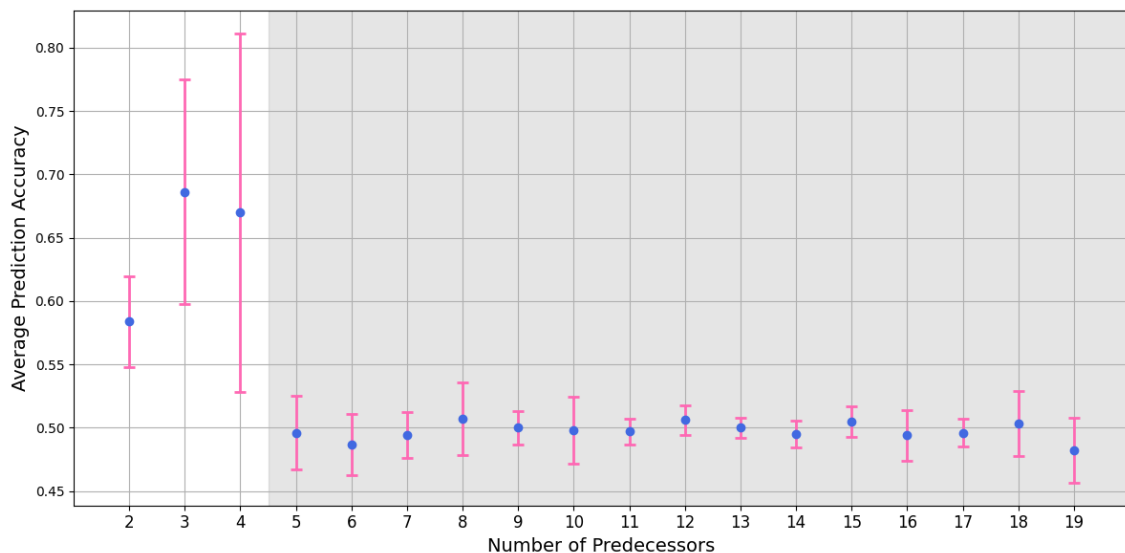


Figure 5.2: Average performance  $\mu(\mathcal{N}_{PD-\delta}, \mathcal{I}_{a,d}, R_{91})$  with error bars representing  $\sigma^2(\mathcal{N}_{PD-\delta}, \mathcal{I}_{a,d}, R_{91})$  over 10 runs for each network model  $PD-\delta$ , where  $\delta \in \{2, 3, \dots, 19\}$ . The shaded gray area highlights the region where a significant drop in performance is observed.

average performance is evaluated over 10 runs for each network model  $\mathcal{N}_{PD-\alpha}$ ,  $\alpha \in \{2, 3, \dots, 19\}$ . The results strongly suggests that for  $\alpha > 4$  there is no meaningful computation performed by the network. Consequently, these types of network models will not be investigated further.

For comparative purposes, a simple network consisting solely of the two driven input nodes was also evaluated. This configuration is equivalent to the readout layer directly processing the input sequence, which violate the fundamental principle of reservoir computing that the readout layer should not perform any computation. Nonetheless, this comparison is crucial as it provides a baseline to determine whether the other network models genuinely enhance the computational capabilities of the system.

Table 5.1 presents the prediction accuracy when this simple network is used as the reservoir. Similar to previous experiments, the readout layer processes the last 20 states of the network evolution. The results indicate that while the readout layer can distinguish between random input and input of type B, it fails to differentiate between inputs of type C or D.

Table 5.1: Mean prediction accuracy  $\mu(-, \mathcal{I}_{a,b}, \mathcal{R})$ ,  $\mu(-, \mathcal{I}_{a,c}, \mathcal{R})$ , and  $\mu(-, \mathcal{I}_{a,d}, \mathcal{R})$  without reservoir, i.e., the readout layer is applied directly to the input sequence.

| Input type    | B    | C    | D    |
|---------------|------|------|------|
| Mean Accuracy | 0.74 | 0.50 | 0.50 |

## 5.2 Transition Rule Performance

In the section on transition rules, the analysis will begin with an overview of the general performance of the rules. Then, a detailed examination will be conducted for each network, aiming to categorize the rules based on their performance and highlight the best performing rules for the network topology. This will specifically involve analysing the distribution of prediction accuracy across different types of input.

### 5.2.1 General Performance

The performance of each transition rule given a network model is summarised in Figure 5.3. The average prediction accuracy is normalised to highlight the relative performance of the rules. Specifically, the prediction accuracy for each input type is normalised over all rules, which means that the new value for the best performing rule for a specific input type is 1, and for the worst performing rule the same value is 0. The normalised average performance is defined as  $\mu(\mathcal{N}_\gamma, \mathcal{I}', R_i)$ , where  $\mathcal{I}'$  indicates that the prediction accuracy for each input type is normalized before combining the averages of the three input types. The results are presented in a grid format, where each square corresponds to the normalised prediction accuracy of a single rule,  $\mu(\mathcal{N}_\gamma, \mathcal{I}', R_i)$ . The rules are displayed in numerical order. Recall that a rule

## 5. Results

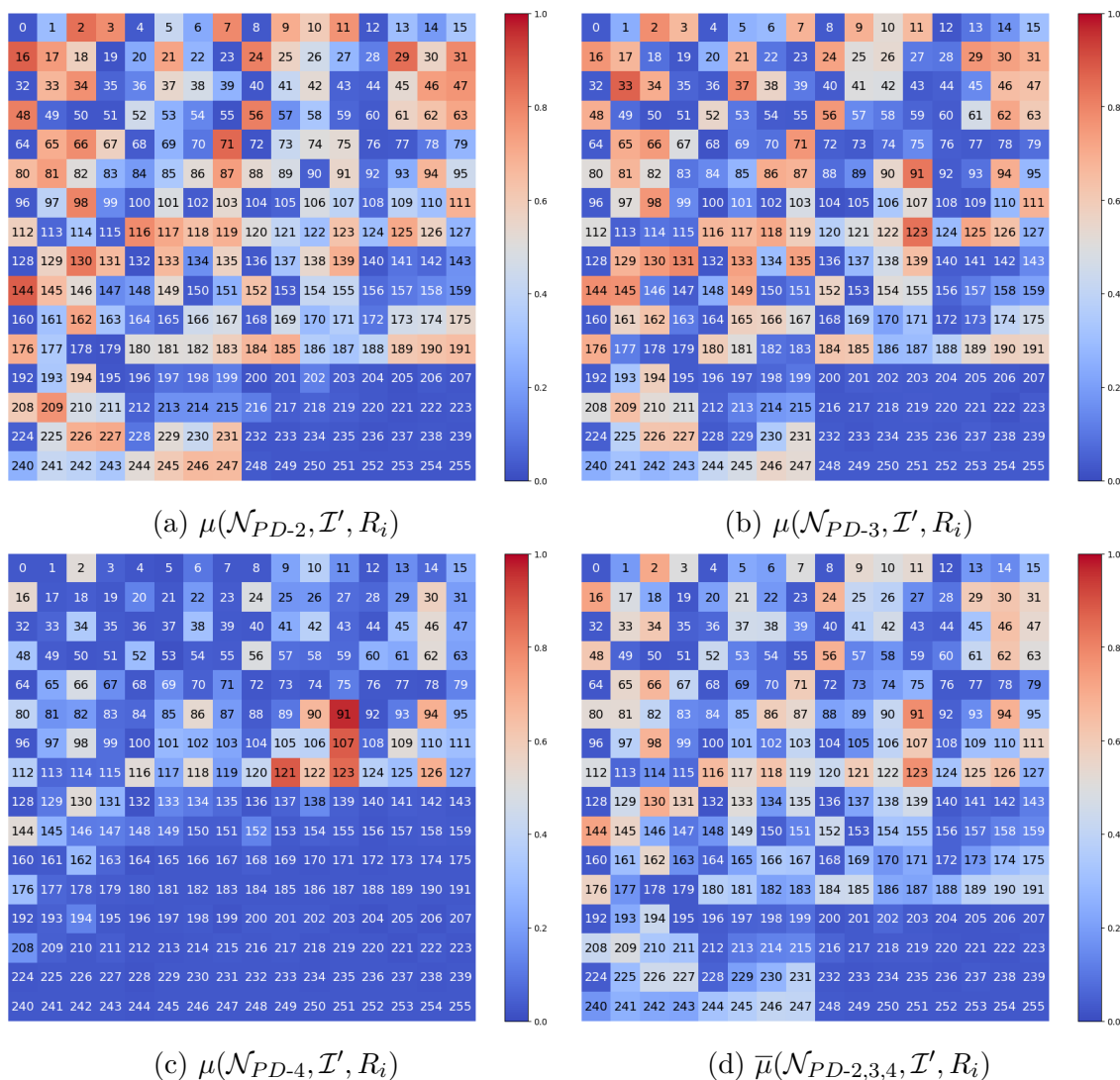


Figure 5.3: Average normalized prediction accuracy  $\mu(\mathcal{N}_{PD-\alpha}, \mathcal{I}', R_i)$  over all three input sequences  $\mathcal{I}'$  for each rule, based on the network models: (a) PD-2, (b) PD-3, (c) PD-4, and (d) the combined average for all three network models. Rules are presented in numerical order

comprises eight mappings from current state to next state and is named after the binary representation of the output cell in these mappings. Consequently, all rules in the same row are identical with respect to the first four mappings, while all rules in the same column share the same last four mappings.

There are four grids presented in total. The first grid corresponds to  $\mathcal{N}_{PD-2}$ , the second to  $\mathcal{N}_{PD-3}$ , and the third to  $\mathcal{N}_{PD-4}$ . The fourth grid is a composite of the previous three, representing the average value of those grids for each rule, denoted as  $\bar{\mu}(\mathcal{N}_{PD-2,3,4}, \mathcal{I}', R_i)$ . While specific rules will be discussed in detail later in this section, some general trends can already be observed across these four grids.

Most notably, the performance of each rule is not random, and certain rules are more effective at processing input than others. Some rules exhibit consistently poor

performance, effectively erasing any information introduced into the network. Conversely, there are rules that consistently demonstrate superior information processing capabilities compared to others. In  $\mathcal{N}_{PD-2}$ , a wide range of rules exhibit decent performance. However, as the number of predecessors increases, only a few rules stand out in terms of performance. Furthermore, there is no universally optimal rule that performs best across all types of network models. Nonetheless, several rules consistently rank among the top performers across all network models considered.

Several general trends can be observed in the grids. Odd-numbered rules tend to perform better than even-numbered ones. This advantage arises from the fact that odd-numbered rules include the mapping  $000 \rightarrow 1$ . Consequently, this configuration prevents networks with many cells in state 0 from becoming stuck in that state. Similarly, rules numbered 127 and below outperform those numbered 128 and above. This distinction is due to the mapping  $111 \rightarrow 0$  present in the lower-numbered rules.

The large rectangle of poorly performing rules in the bottom right corner can be attributed to the fact that all of these rules share the following three mappings:  $111 \rightarrow 1$ ,  $110 \rightarrow 1$ , and  $011 \rightarrow 1$ . This combination of configurations increases the likelihood that the network will converge to an all-1 state. For these rules, the only scenario in which a cell in state 1 can transition to state 0 is if both of its predecessors are in state 0.

As the number of predecessors increases, fewer rules exhibit superior performance relative to others. A factor contributing to this trend is the majority voting mechanism. For  $\mathcal{N}_{PD-2}$ , this is not a concern since there are only two predecessors. In the case of  $\mathcal{N}_{PD-3}$ , there are three predecessors, resulting in three pairs of combinations. Given that the number of pairs is odd, the impact of the majority vote remains relatively insignificant. However, for  $\mathcal{N}_{PD-4}$ , there are four predecessors, leading to a total of six pairwise combinations. With an even number of pairs, situations arise where a tiebreaker is required. In such instances, the tiebreaker rule specifies that the number of votes for state 1 must exceed those for state 0. This requirement can negatively affect the information processing capabilities of certain rules.

The results are presented in Table 5.2. The rules are ordered from the best to the worst performing based on the overall prediction accuracy across all input types and the three network models. Additionally, the table includes the standard deviation for each rule to indicate the consistency of their performance, as well as the rank among all rules for each network model. The highest prediction accuracies are 75.9%, 72.2%, and 77.1% for the three network models with increasing numbers of predecessors, respectively. The standard deviation is relatively high, approximately 10% for most rules, indicating variability in their performance. The exception to this are the rules that consistently exhibit poor performance, which have a standard deviation around 0%.

The results suggest that the exact ranking of rules should be interpreted with caution. Although each input type-network model-rule configuration is run 100 times, this does not eliminate variability in the data. Nevertheless, certain trends are evident and significant. Notably, the performance of rules varies substantially across different network models. For example, the best-performing rule on  $\mathcal{N}_{PD-2}$  is

## 5. Results

Table 5.2: Numerical results from the experiments, including the rank, mean accuracy, and standard deviation for each rule on each network model, as well as the overall results. Only a subset of the rules is shown; complete results can be found in Table A.1 in Appendix A.

| $R_i$      | Rank | $\mu(\mathcal{N}_{PD-2})$ | $\sigma^2(\mathcal{N}_{PD-2})$ | Rank | $\mu(\mathcal{N}_{PD-3})$ | $\sigma^2(\mathcal{N}_{PD-3})$ | Rank | $\mu(\mathcal{N}_{PD-4})$ | $\sigma^2(\mathcal{N}_{PD-4})$ | Rank | $\mu(\cdot)$ | $\sigma^2(\cdot)$ |
|------------|------|---------------------------|--------------------------------|------|---------------------------|--------------------------------|------|---------------------------|--------------------------------|------|--------------|-------------------|
| <b>91</b>  | 61   | 0.678                     | 0.078                          | 2    | 0.717                     | 0.1                            | 1    | 0.771                     | 0.103                          | 1    | 0.722        | 0.093             |
| <b>123</b> | 53   | 0.681                     | 0.105                          | 3    | 0.716                     | 0.098                          | 4    | 0.742                     | 0.096                          | 2    | 0.713        | 0.1               |
| <b>94</b>  | 41   | 0.699                     | 0.124                          | 12   | 0.688                     | 0.105                          | 6    | 0.707                     | 0.097                          | 3    | 0.698        | 0.109             |
| <b>144</b> | 3    | 0.755                     | 0.15                           | 20   | 0.686                     | 0.165                          | 19   | 0.638                     | 0.161                          | 4    | 0.693        | 0.159             |
| <b>16</b>  | 4    | 0.754                     | 0.151                          | 24   | 0.683                     | 0.168                          | 17   | 0.641                     | 0.166                          | 5    | 0.693        | 0.162             |
| <b>56</b>  | 6    | 0.744                     | 0.119                          | 19   | 0.686                     | 0.137                          | 16   | 0.644                     | 0.132                          | 6    | 0.691        | 0.129             |
| <b>2</b>   | 5    | 0.746                     | 0.151                          | 16   | 0.686                     | 0.169                          | 20   | 0.636                     | 0.163                          | 7    | 0.689        | 0.161             |
| <b>98</b>  | 7    | 0.742                     | 0.114                          | 15   | 0.687                     | 0.142                          | 18   | 0.639                     | 0.124                          | 8    | 0.689        | 0.127             |
| <b>126</b> | 59   | 0.679                     | 0.078                          | 30   | 0.675                     | 0.08                           | 5    | 0.707                     | 0.086                          | 9    | 0.687        | 0.081             |
| <b>116</b> | 14   | 0.731                     | 0.125                          | 36   | 0.668                     | 0.135                          | 12   | 0.658                     | 0.094                          | 10   | 0.686        | 0.118             |
| <b>130</b> | 8    | 0.74                      | 0.153                          | 27   | 0.681                     | 0.168                          | 21   | 0.635                     | 0.164                          | 11   | 0.685        | 0.162             |
| <b>30</b>  | 56   | 0.679                     | 0.092                          | 7    | 0.694                     | 0.088                          | 9    | 0.674                     | 0.095                          | 12   | 0.683        | 0.092             |
| <b>24</b>  | 10   | 0.738                     | 0.149                          | 29   | 0.678                     | 0.158                          | 22   | 0.628                     | 0.155                          | 13   | 0.681        | 0.154             |
| <b>46</b>  | 21   | 0.72                      | 0.122                          | 45   | 0.661                     | 0.131                          | 11   | 0.659                     | 0.093                          | 14   | 0.68         | 0.115             |
| <b>66</b>  | 15   | 0.73                      | 0.147                          | 28   | 0.68                      | 0.163                          | 25   | 0.625                     | 0.153                          | 15   | 0.679        | 0.154             |
| <b>118</b> | 46   | 0.696                     | 0.102                          | 25   | 0.682                     | 0.13                           | 14   | 0.657                     | 0.103                          | 16   | 0.678        | 0.112             |
| <b>62</b>  | 48   | 0.693                     | 0.109                          | 26   | 0.681                     | 0.139                          | 13   | 0.658                     | 0.095                          | 17   | 0.678        | 0.114             |
| <b>107</b> | 102  | 0.633                     | 0.076                          | 60   | 0.648                     | 0.089                          | 3    | 0.75                      | 0.099                          | 18   | 0.677        | 0.088             |
| <b>121</b> | 100  | 0.635                     | 0.072                          | 68   | 0.642                     | 0.083                          | 2    | 0.751                     | 0.099                          | 19   | 0.676        | 0.085             |
| <b>86</b>  | 73   | 0.664                     | 0.088                          | 17   | 0.686                     | 0.096                          | 10   | 0.663                     | 0.098                          | 20   | 0.671        | 0.094             |
| <b>48</b>  | 9    | 0.739                     | 0.151                          | 33   | 0.67                      | 0.162                          | 35   | 0.59                      | 0.131                          | 21   | 0.666        | 0.148             |
| <b>34</b>  | 16   | 0.728                     | 0.152                          | 34   | 0.67                      | 0.163                          | 34   | 0.593                     | 0.132                          | 22   | 0.664        | 0.149             |
| <b>131</b> | 38   | 0.702                     | 0.106                          | 5    | 0.699                     | 0.136                          | 36   | 0.59                      | 0.063                          | 23   | 0.663        | 0.102             |
| <b>29</b>  | 2    | 0.757                     | 0.106                          | 11   | 0.689                     | 0.129                          | 72   | 0.537                     | 0.08                           | 24   | 0.661        | 0.105             |
| <b>145</b> | 40   | 0.699                     | 0.101                          | 9    | 0.693                     | 0.137                          | 38   | 0.587                     | 0.067                          | 25   | 0.66         | 0.101             |
| <b>71</b>  | 1    | 0.759                     | 0.111                          | 14   | 0.687                     | 0.138                          | 79   | 0.53                      | 0.071                          | 26   | 0.658        | 0.107             |
| <b>125</b> | 26   | 0.715                     | 0.1                            | 8    | 0.693                     | 0.096                          | 49   | 0.558                     | 0.066                          | 27   | 0.655        | 0.088             |
| <b>65</b>  | 30   | 0.712                     | 0.095                          | 18   | 0.686                     | 0.095                          | 47   | 0.565                     | 0.104                          | 28   | 0.654        | 0.098             |
| <b>176</b> | 37   | 0.705                     | 0.147                          | 31   | 0.671                     | 0.159                          | 39   | 0.584                     | 0.105                          | 29   | 0.653        | 0.137             |
| <b>122</b> | 118  | 0.609                     | 0.081                          | 50   | 0.658                     | 0.077                          | 8    | 0.692                     | 0.082                          | 30   | 0.653        | 0.08              |
| ...        | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...          | ...               |
| <b>133</b> | 45   | 0.697                     | 0.121                          | 10   | 0.692                     | 0.109                          | 69   | 0.54                      | 0.05                           | 36   | 0.643        | 0.093             |
| ...        | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...          | ...               |
| <b>33</b>  | 49   | 0.692                     | 0.102                          | 1    | 0.722                     | 0.098                          | 104  | 0.511                     | 0.032                          | 40   | 0.642        | 0.077             |
| ...        | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...          | ...               |
| <b>129</b> | 54   | 0.68                      | 0.082                          | 22   | 0.685                     | 0.084                          | 80   | 0.53                      | 0.046                          | 45   | 0.631        | 0.071             |
| ...        | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...          | ...               |
| <b>135</b> | 58   | 0.679                     | 0.097                          | 6    | 0.694                     | 0.091                          | 92   | 0.517                     | 0.035                          | 48   | 0.63         | 0.074             |
| ...        | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...          | ...               |
| <b>37</b>  | 62   | 0.677                     | 0.074                          | 4    | 0.707                     | 0.091                          | 183  | 0.5                       | 0.035                          | 51   | 0.628        | 0.066             |
| ...        | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...          | ...               |
| <b>90</b>  | 187  | 0.507                     | 0.064                          | 47   | 0.66                      | 0.085                          | 7    | 0.7                       | 0.101                          | 60   | 0.622        | 0.084             |
| ...        | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...          | ...               |
| <b>246</b> | 39   | 0.701                     | 0.162                          | 71   | 0.641                     | 0.159                          | 204  | 0.5                       | 0                              | 75   | 0.614        | 0.107             |
| ...        | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...          | ...               |
| <b>191</b> | 44   | 0.698                     | 0.16                           | 72   | 0.64                      | 0.155                          | 204  | 0.5                       | 0                              | 77   | 0.613        | 0.105             |
| ...        | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...          | ...               |
| <b>247</b> | 42   | 0.699                     | 0.165                          | 75   | 0.637                     | 0.155                          | 204  | 0.5                       | 0                              | 79   | 0.612        | 0.107             |
| ...        | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...          | ...               |
| <b>110</b> | 125  | 0.598                     | 0.099                          | 104  | 0.577                     | 0.085                          | 28   | 0.608                     | 0.091                          | 88   | 0.595        | 0.092             |
| ...        | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...          | ...               |
| <b>15</b>  | 143  | 0.562                     | 0.068                          | 118  | 0.551                     | 0.068                          | 52   | 0.556                     | 0.1                            | 121  | 0.556        | 0.079             |
| ...        | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...          | ...               |
| <b>85</b>  | 142  | 0.562                     | 0.075                          | 125  | 0.543                     | 0.071                          | 55   | 0.552                     | 0.102                          | 128  | 0.552        | 0.083             |
| ...        | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...          | ...               |
| <b>54</b>  | 157  | 0.544                     | 0.063                          | 190  | 0.502                     | 0.034                          | 109  | 0.508                     | 0.042                          | 168  | 0.518        | 0.047             |
| ...        | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...          | ...               |
| <b>0</b>   | 232  | 0.5                       | 0                              | 222  | 0.5                       | 0                              | 204  | 0.5                       | 0                              | 232  | 0.5          | 0                 |
| <b>32</b>  | 232  | 0.5                       | 0                              | 222  | 0.5                       | 0                              | 204  | 0.5                       | 0                              | 232  | 0.5          | 0                 |
| <b>250</b> | 232  | 0.5                       | 0                              | 222  | 0.5                       | 0                              | 204  | 0.5                       | 0                              | 232  | 0.5          | 0                 |
| <b>251</b> | 232  | 0.5                       | 0                              | 222  | 0.5                       | 0                              | 204  | 0.5                       | 0                              | 232  | 0.5          | 0                 |
| <b>254</b> | 232  | 0.5                       | 0                              | 222  | 0.5                       | 0                              | 204  | 0.5                       | 0                              | 232  | 0.5          | 0                 |
| <b>255</b> | 232  | 0.5                       | 0                              | 222  | 0.5                       | 0                              | 204  | 0.5                       | 0                              | 232  | 0.5          | 0                 |
| ...        | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...                       | ...                            | ...  | ...          | ...               |
| <b>222</b> | 245  | 0.499                     | 0.025                          | 256  | 0.497                     | 0.022                          | 146  | 0.501                     | 0.019                          | 252  | 0.499        | 0.022             |
| <b>200</b> | 254  | 0.498                     | 0.025                          | 247  | 0.498                     | 0.024                          | 165  | 0.501                     | 0.022                          | 253  | 0.499        | 0.024             |
| <b>19</b>  | 256  | 0.496                     | 0.031                          | 237  | 0.5                       | 0.021                          | 158  | 0.501                     | 0.021                          | 254  | 0.499        | 0.025             |
| <b>221</b> | 244  | 0.5                       | 0.022                          | 242  | 0.499                     | 0.022                          | 253  | 0.497                     | 0.021                          | 255  | 0.499        | 0.022             |
| <b>132</b> | 251  | 0.499                     | 0.022                          | 253  | 0.497                     | 0.023                          | 254  | 0.497                     | 0.026                          | 256  | 0.498        | 0.024             |

only ranked 26th overall, while the best-performing rule on  $\mathcal{N}_{PD-3}$  is ranked 40th. This indicates that no single rule universally excels across all network models. However, some rules consistently achieve around 70% prediction accuracy across all network models, suggesting a degree of generalization capability. These findings highlight the importance of considering specific network model characteristics when evaluating rule performance.

## 5.2.2 Categorisation based on Accuracy Distribution

A more detailed evaluation and categorization can be achieved by examining the accuracy distribution. This distribution reveals the spread of accuracy for individual instances of configurations with different input type, network model, and transition rule. Based on the location and size of peaks in the distributions, rules can be grouped into four categories with similar patterns by applying a few simple criteria. Figure 5.4 illustrates examples of these distributions using kernel density estimation plots, highlighting the core properties of each group.

*Group 1* consists of rules that offer no valuable computation. An example of such a rule is depicted in Figure 5.4a, where the distributions for all three input types are centered around 0.5. This indicates that, regardless of the network instance, the rule effectively erases any information introduced into the network.

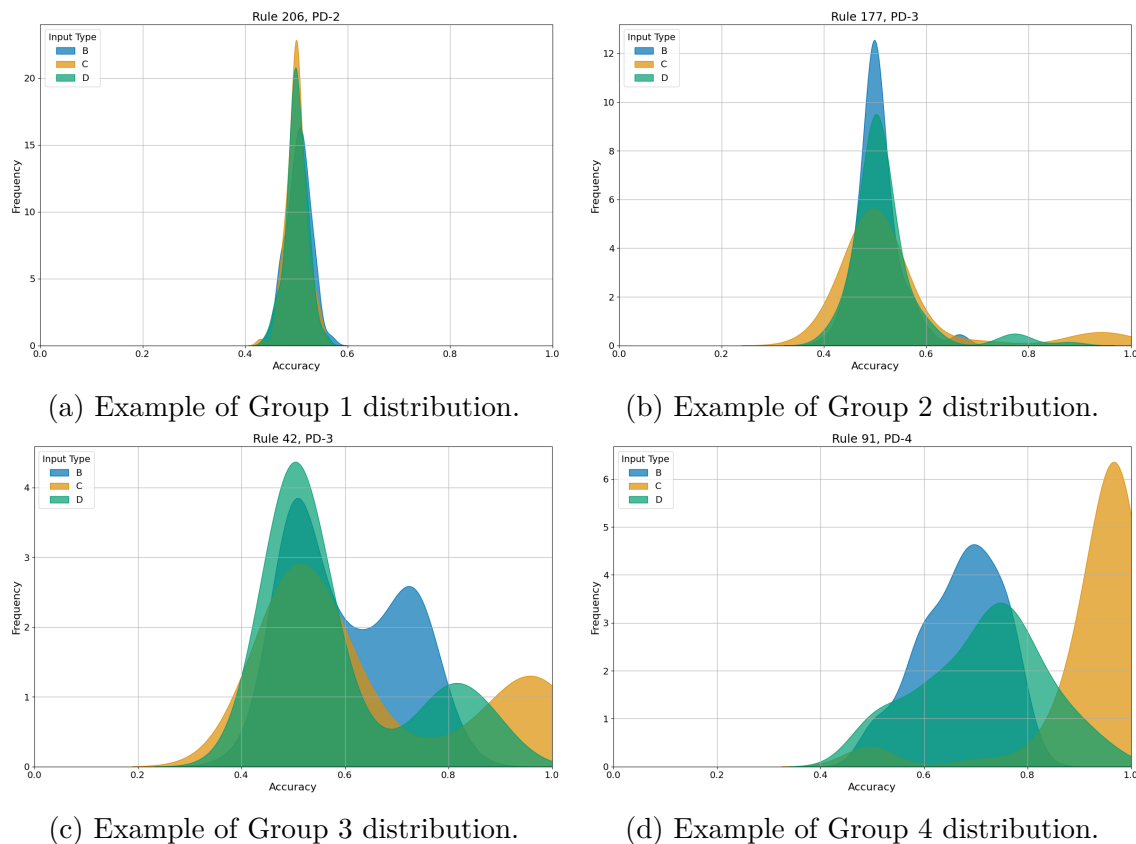


Figure 5.4: Four groups of distribution types.

*Group 2* comprises rules similar to those in Group 1 but with one key difference. These distributions exhibit a clear tail for one or more input types, suggesting that, for a small subset of network instances, the rule can perform computations on the network. An example of this is shown in Figure 5.4c.

*Group 3* includes rules where the distributions demonstrate a clear ability to perform computations on the networks, but this capability is dependent on the network instance. This is indicated by a substantial portion of the distribution’s density being located above 60% accuracy, while there remains a significant peak at 0.5 for at least one input type. An example is provided in Figure 5.4b, where each input type exhibits two peaks: one at 50% accuracy and another at a higher accuracy.

*Group 4* consists of the best-performing rules, characterized by the absence of a significant peak at 50% accuracy for any of the input types. These model configurations display the ability to generalize effectively and represent the best options identified in this study for computation on CAN. An example is illustrated in Figure 5.4d, where the accuracy is consistently high.

The categorization is based on several metrics that differentiate rules across the various groups. In summary, rules where all peaks are below 60% accuracy are allocated to Group 1. Rules where all the highest peaks exceed 60% accuracy, with no significant peak below 60%, are placed in Group 4. Rules with significant peaks both above and below 60% accuracy are assigned to Group 3. The remaining rules, which exhibit a tail in their distribution, are categorized into Group 2. Previous results have suggested that the performance of individual rules is closely tied to the network model. Therefore, the categorization will be conducted separately for each of the three network models, rather than combining them all.

### 5.2.3 Categorisation of Rules for $\mathcal{N}_{PD-2}$

The categorisation of rules for  $\mathcal{N}_{PD-2}$  is depicted in Figure 5.5. There are a total of nine rules in Group 4, specifically rules 29, 56, 65, 71, 98, 118, 125, 131, and 145. Most rules fall into Group 3, while the remaining rules are evenly distributed between Groups 1 and 2. It is evident that a significant number of rules are effective on this network model, although most are dependent on the specific network instance.

The distributions of a selected set of four rules are illustrated in Figure 5.6. These rules were chosen for their superior performance with one or more input types. The top-performing rules for  $\mathcal{N}_{PD-2}$  are rules 71 and 29. These two rules are mirrored versions of each other and exhibit similar behaviour. The two rules are symmetrical and share the same general behaviour. The distribution for Rule 71 is shown in Figure 5.6a, and Rule 29 displays a comparable distribution. These rules perform exceptionally well on alternating input, similarly to a pure readout layer for biased input, and handle alternating input with noise effectively.

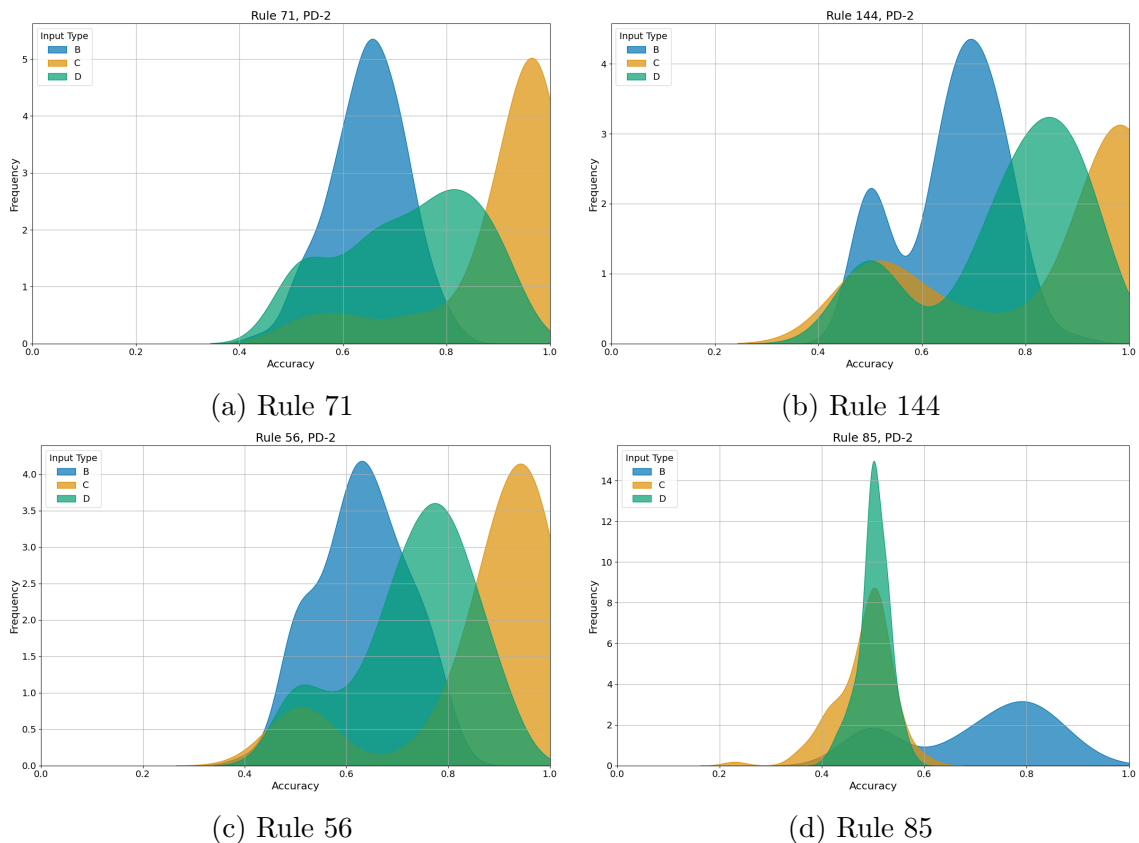
Following rules 71 and 29, the next best rule is Rule 144, with Rules 16 and 2, among other rules, demonstrating similar behaviour and performance. These rules are characterized by incorporating one of the mappings  $100 \rightarrow 1$  and  $001 \rightarrow 1$ , with most other outputs in state 0. The distribution for Rule 144 is depicted in Figure 5.6b.

Figure 5.5: Categorisation of rules on  $\mathcal{N}_{PD-2}$ .

It can be observed that for certain network instances, this model configuration performs exceptionally well, but there is also a significant peak around 50% accuracy, highlighting the rule’s network dependency. Consequently, these types of rules are not included in Group 4.

Next, rules 56 and 98, which are mirrored versions of each other and exhibit similar behaviour, show notable performance. The distribution for Rule 56 is presented in Figure 5.6c. This distribution resembles that of Rule 144, with the rule performing best on alternating text, followed by alternating text with noise, and finally biased text. However, the peaks at 50% accuracy are less pronounced, but the overall performance is slightly lower.

A rule that does not generally perform well but excels with biased text is Rule 85, along with Rule 15, which shares similar behaviour. The distribution for Rule 85 is shown in Figure 5.6d. It is evident that this rule cannot process alternating text effectively; however, it performs exceptionally well with biased text. Although the average performance is still lower than using the readout layer directly on the input, the peak performance is better. The peak at 50% accuracy indicates that the rule’s performance is still dependent on the network instance, meaning it does not work uniformly well across all network instances.

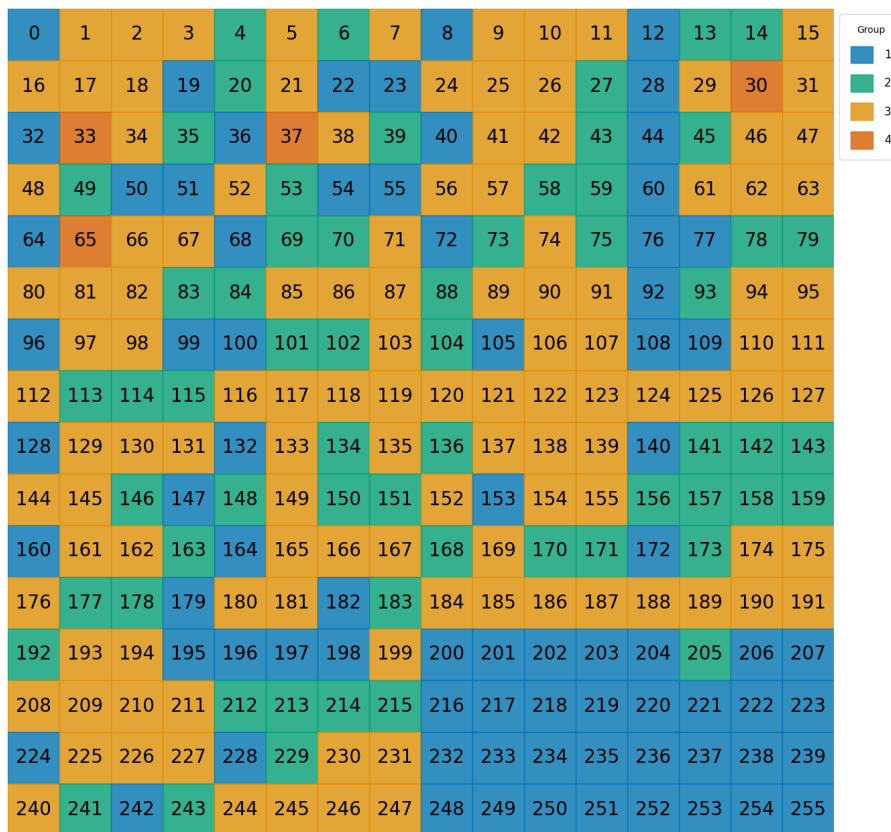
Figure 5.6: Four of the best rules on  $\mathcal{N}_{PD-2}$ .

### 5.2.4 Categorisation of Rules for $\mathcal{N}_{PD-3}$

The categorization of rules for  $\mathcal{N}_{PD-3}$  is illustrated in Figure 5.7. A total of four rules falls into Group 4: rules 30, 33, 37, and 65. The majority of rules are classified into Group 3, while the remaining rules are evenly distributed between Groups 1 and 2. Similar to  $\mathcal{N}_{PD-2}$ , it is evident that many rules are effective on this network model, but their performance is largely dependent on the specific network instance. However, the number of effective rules on  $\mathcal{N}_{PD-3}$  is lower than on  $\mathcal{N}_{PD-2}$ .

The distributions of a select four rules are illustrated in Figure 5.8. These rules were chosen for their exceptional performance with one or more input types. Similar to  $\mathcal{N}_{PD-2}$ , the best rules come in pairs with analogous performance and behavior. Moreover, for the three pairs discussed here, one rule in the pair is classified in Group 3 and the other in Group 4, despite their near-identical characteristics. Interestingly, the rule in Group 4 is not necessarily the one with the highest performance, highlighting the limitations of using purely metric-based criteria for grouping. Nevertheless, the rules in Group 4 are among the best performers, indicating that the groups are still meaningful.

The top-performing rule is Rule 33, which shares similar performance and behaviour with Rule 123. The distribution for Rule 33 is depicted in Figure 5.8a. Unlike the best rules for  $\mathcal{N}_{PD-2}$ , this model configuration performs better on biased text than on text with noise. The overall performance is also approximately 4% lower than the

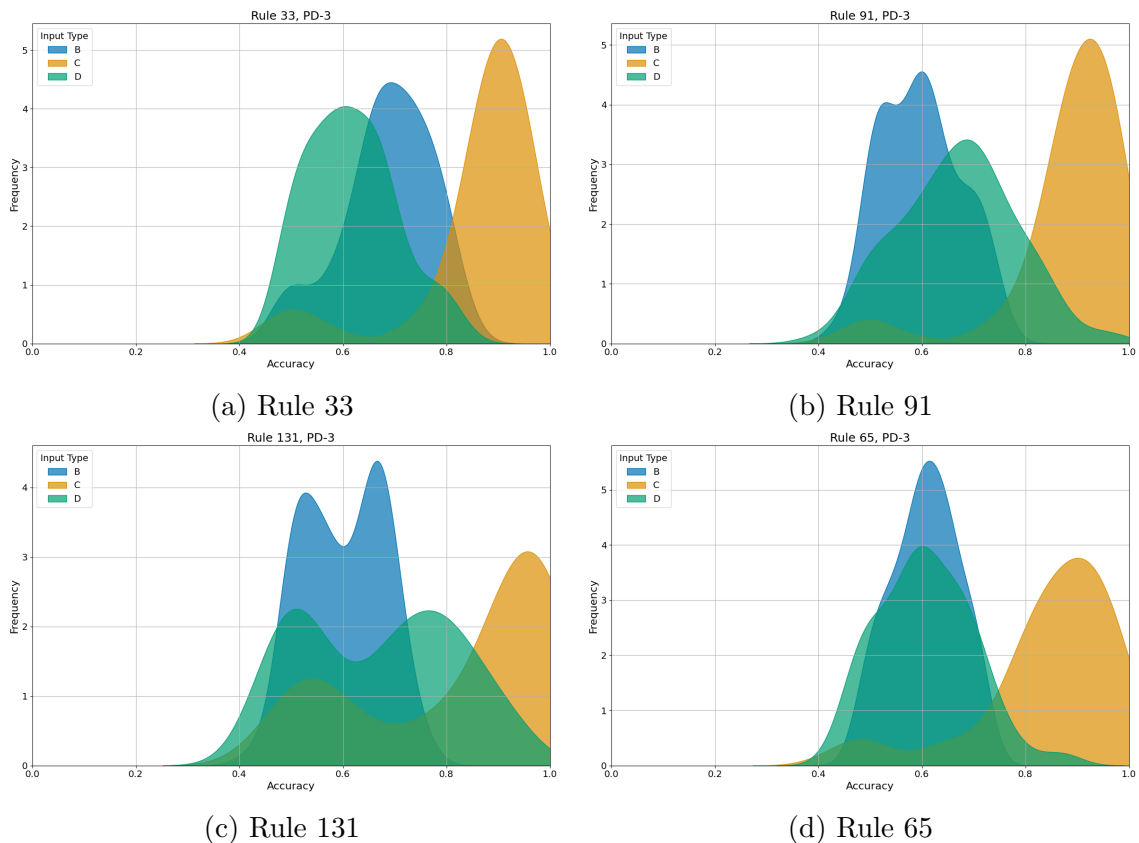
Figure 5.7: Categorisation of rules on  $\mathcal{N}_{PD-3}$ .

best performer on  $\mathcal{N}_{PD-2}$ .

The second best-performing rule is Rule 91, which shares similar performance and behaviour with Rule 37. The distribution for Rule 91 is shown in Figure 5.8b. These rules perform better on text with noise than on biased text. In fact, the performance on biased text is relatively poor, with many runs centred around 50% accuracy. However, this is offset by better performance on alternating text and text with noise compared to Rule 33. Additionally, Rules 30 and 135 exhibit different behaviour but show similar distributions.

A rule that matches the average performance of the aforementioned rules is Rule 131. The distribution for Rule 131 is depicted in Figure 5.8c. It is evident that Rule 131 heavily depends on the network instance, but when the setup is optimal, its performance surpasses that of its counterparts.

Finally, the last rule in Group 4 that has not been mentioned yet is Rule 65, which was also part of Group 4 for  $\mathcal{N}_{PD-2}$ . The distribution for Rule 65 is shown in Figure 5.8d. The peaks just surpass the 60% accuracy threshold, which explains why the average performance is relatively poor despite belonging to Group 4. This trend was also observed for this rule on  $\mathcal{N}_{PD-2}$ , where it was also in Group 4.

Figure 5.8: Four of the best rules on  $\mathcal{N}_{PD-3}$ .

### 5.2.5 Categorisation of Rules for $\mathcal{N}_{PD-4}$

The categorisation of rules for  $\mathcal{N}_{PD-4}$  is illustrated in Figure 5.9. A total of two rules are classified in Group 4: rules 91 and 121. Most rules fall into Group 1, while the remaining rules are evenly split between Groups 2 and 3. It is evident that the number of effective rules for this network model is significantly more limited compared to the previous models.

The distributions of the four best-performing rules are shown in Figure 5.10. For  $\mathcal{N}_{PD-4}$ , the four rules that perform significantly better than the others are, in order: 91, 121, 107, and 123. Additionally, four more rules — 126, 94, 90, and 122 — stand out among the rest. Out of these rules, 91 and 121 belong to group 4.

The distributions for the top four rules exhibit many similarities. They all perform well on alternating text, and their performance on biased text is comparable. The primary difference lies in their performance on text with noise. Compared to rules on other network models, the performance of these rules is slightly lower than the best-performing configurations on  $\mathcal{N}_{PD-2}$ . However, rule 91 on  $\mathcal{N}_{PD-4}$  stands out as the best overall performing rule, with an accuracy of 77.1%, compared to 75.9% for Rule 71 on  $\mathcal{N}_{PD-2}$ . It is important to note that there is some variance in the prediction accuracy that affects the true difference.



Figure 5.9: Categorisation of rules on  $\mathcal{N}_{PD-4}$ .

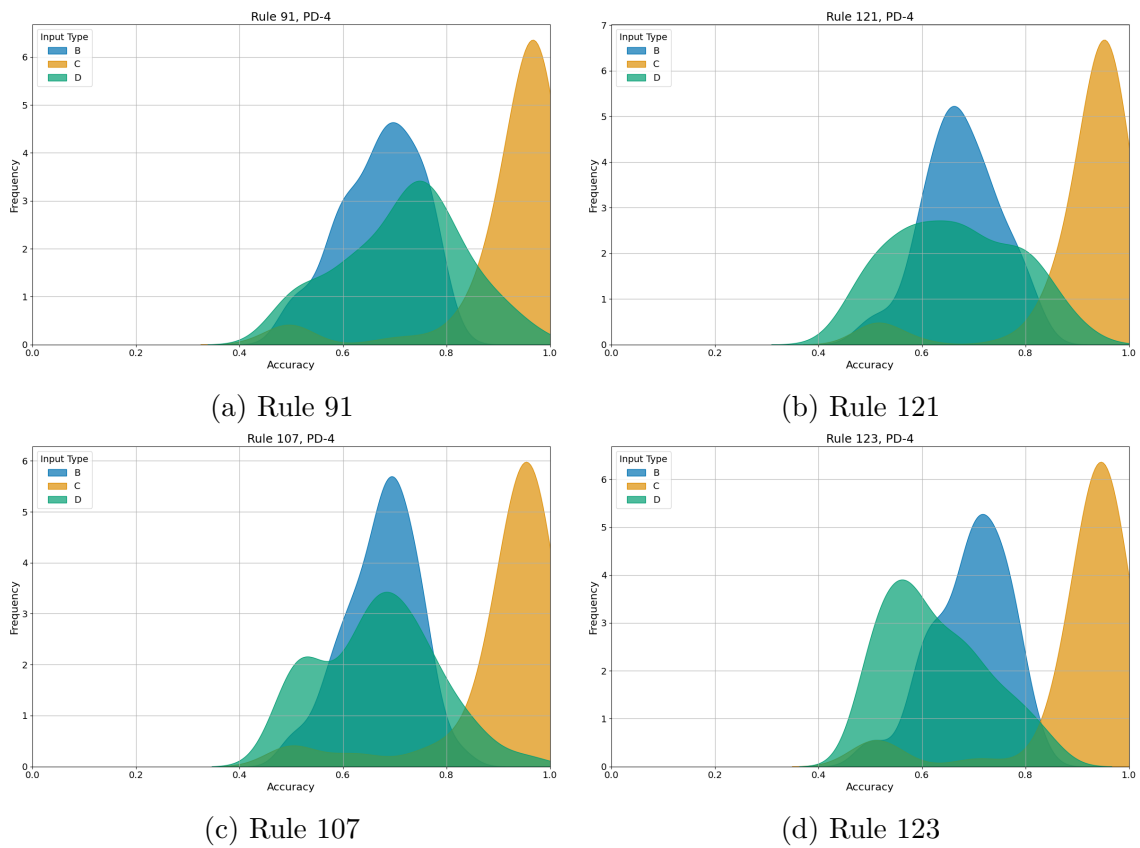


Figure 5.10: Four of the best rules on  $\mathcal{N}_{PD-4}$ .

### 5.3 Information Gain

The information gain of the rules on different network models was also examined. The average information gain  $\overline{IG}(\cdot)$  is presented in Table 5.3, the full results can be viewed in Table A.3 in Appendix A.  $X_{current}$  represents the current cell,  $X_a$  the first predecessor,  $X_b$  the second predecessor, and so on. The observations are as expected. For networks with two predecessors, all three cells, including the current cell, exhibit the same level of information gain. For networks with three predecessors, the current cell generates the most information gain and has the greatest influence on the outcome. In this configuration, the middle predecessor, which is considered once as a left neighbour and once as a right neighbour, has the least influence. A similar trend is observed for networks with four predecessors, where the current cell again exerts the most influence, and the middle cells have the least. Additionally, the average information gain for a single predecessor, mean  $\overline{IG}(X_{a-d})$ , diminishes as the number of predecessors increases. This indicates that predecessors influence the cell less when there are more predecessors, thereby giving the current cell a larger impact on the outcome. Furthermore, the uncertainty  $H(Y)$  when there is no prior information decreases for networks with more predecessors, indicating that denser networks become more predictable.

Information gain is also employed to group similar rules based on behaviour rather than performance. Several important observations can be highlighted. For  $\mathcal{N}_{PD-4}$ , rules 91, 94, 107, 121, and 122 exhibit identical information gain values for the current cell and all predecessors at 0.0084. This is also true for rules 90 and 126; however, in these cases, the information gain from knowledge of either the current cell or a single predecessor is 0. Similarly, for rule 123, which is among the top 8 performers on  $\mathcal{N}_{PD-4}$ , there is no information gain from a single predecessor, but there is information gain from knowledge of the current cell.

Among the best rules for  $\mathcal{N}_{PD-3}$ , there is more variation in information gain. Rules 91 and 37 show consistent information gain of 0.018 from the state of either the current cell or a single predecessor. Rules 123 and 33 exhibit no information gain from a single predecessor but do show information gain from the state of the current cell. However, for rules 30, 131, and 135, the information gain from predecessors depends on their order. In all three cases, knowledge of the state of the first predecessor in order yields significantly higher information gain compared to any other predecessor.

The best rules for  $\mathcal{N}_{PD-2}$  can also be described using information gain. Without prior information, the uncertainty of the outcome when using rules 29 or 71 is 1, meaning the states 0 and 1 are equally likely. Knowledge of the current cell's state

Table 5.3: Average information gain for 2, 3, and 4 predecessors, respectively.

| $\mathcal{N}_\gamma$ | $H(Y)$ | $\overline{IG}(X_{current})$ | $\overline{IG}(X_a)$ | $\overline{IG}(X_b)$ | $\overline{IG}(X_c)$ | $\overline{IG}(X_d)$ | Mean $\overline{IG}(X_{a-d})$ |
|----------------------|--------|------------------------------|----------------------|----------------------|----------------------|----------------------|-------------------------------|
| $\mathcal{N}_{PD-2}$ | 0.902  | 0.122                        | 0.122                | 0.122                | -                    | -                    | 0.122                         |
| $\mathcal{N}_{PD-3}$ | 0.857  | 0.167                        | 0.104                | 0.010                | 0.104                | -                    | 0.072                         |
| $\mathcal{N}_{PD-4}$ | 0.806  | 0.176                        | 0.062                | 0.019                | 0.019                | 0.062                | 0.040                         |

produces no information gain, while knowledge of either predecessor provides an information gain of 0.19. Rules 56 and 98 yield the same information gain from the current cell or a single predecessor at 0.049. This is also true for rules 2 and 16, but with an information gain of 0.14. Rule 144 is asymmetrical, as knowledge of the first predecessor provides an information gain of 0.31, while knowledge of the current cell or second predecessor yields no information gain.

It is important to note that these observations are not unique to the best performing rules. Many rules with the same information gain distribution as the best rules, do not perform well. Additionally, it is generally observed that the average information gain from a single predecessor for the best performing rules is relatively low compared to other rules. This could suggest that the combined values of multiple predecessors are necessary to extract meaningful information from the system. It is this interplay among multiple predecessors that likely creates the complex and interesting computational behaviour observed in the best performing rules.



# 6

## Discussion

This chapter is organized into six sections. The first three sections address the use of cellular automata as a reservoir from various perspectives. Specifically, the performance of different network models and transition rules is discussed, as well as the performance of the overall system. The fourth section discusses reservoir computing as a methodological framework. The fifth section addresses the utilisation of a genetic algorithm to optimise cellular automata networks, while the last section introduces potential directions for future research.

### 6.1 Finding the Correct Network Topology

It appears that a network topology should have a certain amount of structure to function well. This was shown by networks with a fixed amount of predecessors outperforming their counterparts. Similarly, a network with more constraints on the connections, like the small-world network, consistently performs better than a more random network like the Erdős-Rényi network. This is in line with the edge of chaos idea, which emphasizes the importance of balancing the chaotic and deterministic nature of the reservoir. The results indicate that for the transition rules examined in this study, a network cannot be too dense, as this causes the system to deviate from this edge. It is also clear that trivial connections, where a cell has none or only one predecessor, should be avoided, as these provide no additional computation.

The findings suggest that different network topologies do not produce significant differences in performance, unlike factors such as density and a uniform predecessor count. Among the networks examined in this study,  $\mathcal{N}_{PD-2}$ , where all cells have two predecessors, appears to be closest to the edge of chaos. Although other configurations sometimes outperform rules on  $\mathcal{N}_{PD-2}$ , these instances are few and the performance difference is negligible. While these configurations should not be entirely disregarded, it is more sensible to focus on the network model that consistently performs well across a wide range of transition rules.

The  $\mathcal{N}_{PD-2}$  network model was kept simple but can be further optimized. Potential areas for improvement include examining the order of predecessors, allowing for dynamic or varying network sizes, exploring small-world or scale-free  $\mathcal{N}_{PD-2}$  networks, and identifying specific connection patterns that enhance performance. However, caution is necessary when optimizing the system. Overfitting remains a significant

risk, and optimizing for a particular problem or input type might reduce the network's generalization capacity.

From a practical perspective, a network with a sparse and fixed number of predecessors is also advantageous. Each connection and transition logic incurs a cost in a physical system, making simpler systems easier to implement in physical hardware.

## 6.2 Selecting an Appropriate Transition Rule

It is evident that the selection of a suitable transition rule depends on the underlying network. For certain network models, such as  $\mathcal{N}_{PD-4}$ , determining an appropriate rule is comparable to identifying a single rule that actually performs computation effectively, as the options may be limited. In other cases, there are more options and multiple factors to consider when choosing the best transition rule for a specific task.

### 6.2.1 Categorisations of Elementary Cellular Automata

Several properties are consistently observed in the best-performing transition rules. It is essential to examine these properties in greater depth to gain a better understanding of their impact. To facilitate this discussion, Figure 6.1 illustrates a selection of the top-performing rules from the experiments as elementary cellular automata rules.

First and foremost, elementary cellular automata have been extensively studied in prior research, leading to established categorisations of the rules, such as Wolfram's four classes of behaviour. For elementary cellular automata, these categorizations are typically made on a one-dimensional grid rather than the networks studied here. Consequently, they cannot be applied directly to this study but serve as useful guidelines. This also means that the categorisation is the most relevant for  $\mathcal{N}_{PD-2}$

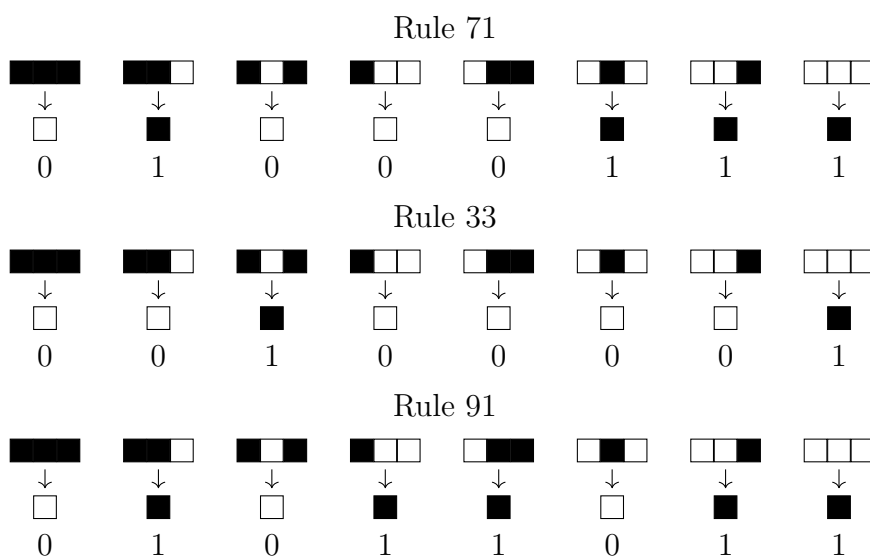


Figure 6.1: Visualisation of the three elementary cellular automata rules with the best performance on network model  $\mathcal{N}_{PD-2}$ ,  $\mathcal{N}_{PD-3}$ , and  $\mathcal{N}_{PD-4}$ , respectively.

networks, as the transition rule on that network model is the most similar to the traditional elementary cellular automata rule, with only two neighbours. That is also why in this section, the underlying elementary cellular automata is considered for the discussion, rather than the extended version.

For example, totalistic cellular automata are rules that depend solely on the total value of the cells in the neighbourhood, i.e., the current cell and its predecessors. This means that the mappings can be divided into four groups: (111; total value 3), (110, 101, 011; 2), (100, 010, 001; 1), and (000; 0), and within each group the outcome should be identical. Among the elementary cellular automata rules classified as totalistic, Rule 126 ranks 9th overall, while Rule 129 is 45th, with other rules performing significantly worse. Rule 126 and 129 can be described as rules where, if all cells in the neighbourhood are in the same state, i.e., mappings 111 and 000, the output is one state, but when the states differ, the output is the opposite state. While this setup appears to compute, other totalistic rules does not. The results indicate that this categorization does not appear to be particularly relevant for this problem.

Another categorisation is outer-totalistic cellular automata, where the output depends on the total value of the predecessors and the state of the current cell separately. This means that the mappings can now be divided into three groups: (111, 101; total value 2 <sup>1</sup>), (110, 100, 011, 001; 1), and (010, 000; 0), and within each group the outcome should either be identical, the state of the current cell should remain the same, or the state of the current cell should be flipped. There are 64 outer-totalistic elementary cellular automata rules. Among the best overall performing rules, those classified as outer-totalistic include rules 33, 37, 90, 91, 94, 122, 123, 126, 129, and 133. Rule 90, ranked 60th overall, is the worst among these, and rule 91, ranked 1st overall, is of course the best. There is significant overlap between these rules and the best-performing rules on  $\mathcal{N}_{PD-3}$  and  $\mathcal{N}_{PD-4}$ , indicating that properties tied to outer-totalistic rules become useful as the number of predecessors increases beyond two. This is not surprising, as the order of predecessors becomes less important due to the extension of the rule, making rules that consider only the density, not the order, more relevant. However, this categorization does not appear to be relevant for  $\mathcal{N}_{PD-2}$ , as the rules listed above are not the best performers on  $\mathcal{N}_{PD-2}$ .

Wolfram's four classes of behaviour is another natural categorisation of elementary cellular automata. The most interesting rules are those belonging to Class 4, which exhibit complex behaviour, in contrast to the predictable patterns in Classes 1 and 2 and the chaotic behaviour of Class 3. On the classic one-directional grid, rules commonly recognized as Class 4 include Rule 54 and Rule 110. These rules are ranked 168th and 88th, respectively, in overall performance in this study's experiments. This suggests that behaviour observed on a one-directional grid does not directly translate to a network context, or alternatively, that a Class 4 rule does not necessarily make a great reservoir.

---

<sup>1</sup>The value is computed from the first and the last digit both being one, the one in the middle is ignored

## 6.2.2 The Ideal Transition Rule

Given the absence of an established categorization that accurately describes the results of this study for all networks, it is necessary to examine the transition rules in more detail. The effectiveness of transition rules is heavily dependent on both the network and the input types. Consequently, it is challenging to determine whether a well-performing rule generalizes effectively or if it benefits significantly from the specific networks and input types used in this study. However, it is evident that the transition rule plays a more critical role than the network topology, as there is a much larger difference in performance between different rules compared to network models.

It is tempting to conclude that outer-totalistic rules perform best for  $\mathcal{N}_{PD-3}$  and  $\mathcal{N}_{PD-4}$ , but when focusing on the transition rules for the  $\mathcal{N}_{PD-2}$  network, two different types of rules emerge. First, there are rules like 71 and 29. These rules feature the mapping  $111 \rightarrow 0$  and  $000 \rightarrow 1$ , which are commonly found in high-performing rules. Additionally, they include the mapping  $101 \rightarrow 0$  and  $010 \rightarrow 1$ , which has the opposite effect to previous mappings, since here the state of the current cell remains unchanged. These rules are also balanced, with four mappings with output 0, and four mappings with output 1. So, when the states of the predecessors differ, the current cell either flips or remains the same, depending on whether the first or second predecessor is in state 1. In terms of information gain, all predecessors produce the same information gain regardless of their position, which is also frequently observed in high-performing transition rules. In summary, the rules appear to do a little bit of everything, yet they are designed in a manner that enables the network to perform meaningful computations.

There is also another type of transition rule that, surprisingly, works well on  $\mathcal{N}_{PD-2}$ . These rules produce a behaviour that can equal a shock wave, due to either of the mappings  $001 \rightarrow 1$  and  $100 \rightarrow 1$ , and most of the remaining mappings outputting state 0. Examples of such rules include 2, 16, 24, 48, 56, 98, 130, and 144, all of which perform well on  $\mathcal{N}_{PD-2}$ . The idea is that most cells in the network are in state 0, and the network reacts to the input nodes by sending a shock wave throughout the network. The shock wave can multiply while moving through the network and will only stop if it interacts with another wave or if it encounters certain network structures. Fundamentally, this type of transition rule is intuitive and relatively easy to understand, especially compared to the previously mentioned rules.

Among the two types of successful transition rules for  $\mathcal{N}_{PD-2}$  networks, the most promising ones are rules 29 and 71. These rules show the best average results and are the least dependent on the specific network instance, indicating superior generalization capabilities. Unfortunately, it is not easy to understand why these rules are the top-performers. Then, for denser networks like  $\mathcal{N}_{PD-3}$  and  $\mathcal{N}_{PD-4}$ , outer-totalistic rules, such as rule 91, should be considered.

### 6.2.3 The Inverse Rules Problem

An observed phenomenon was that inverse rules, which are pairs of rules where all mappings are inverted compared to each other, i.e.,  $111 \rightarrow 0$  becomes  $000 \rightarrow 1$  and so on, did not generally exhibit the same performance. Intuitively, they should perform similarly since they are essentially mirror images of each other, particularly given the setup of this study. While there are a few possible explanations for this discrepancy, none seem applicable to this study's specific conditions.

When working with alphabets containing more than two letters, it is natural that after one-hot encoding, transition rules that primarily respond to predecessors in state 1 would perform differently than those that react to state 0. In this study, however, an alphabet of only two letters was used, implying that when one input node is in state 0, the other is in state 1, and vice versa. This suggests that inverse rules should exhibit similar performance. Nevertheless, this was not observed.

For instance, Rule 2 sends a pulse of cells in state 1 through the network, while Rule 191 sends a pulse of cells in state 0. Intuitively, these rules should have similar performance given the experimental setup. However, in the ranking for  $\mathcal{N}_{PD-2}$ , Rule 2 ranks 5th, whereas Rule 191 ranks 44th. If this were an isolated observation, it could be attributed to chance due to the variance in the results, but this pattern is observed in other cases as well. Rule 144 and its counterpart Rule 246 are ranked 3rd and 39th, respectively, while Rule 16 and its counterpart Rule 247 are ranked 4th and 42nd, and the list can be continued.

There are also possible explanations that has been discussed in previous chapters. One reason for the difference in performance with a larger number of predecessors, as seen in  $\mathcal{N}_{PD-4}$ , is that a tie can occur in the majority vote. In the tiebreaker, there must be more votes for state 1 than state 0, potentially impacting performance. However, for  $\mathcal{N}_{PD-2}$  and  $\mathcal{N}_{PD-3}$ , ties cannot occur, yet the pattern persists. Another reason could be the handling of the biased text input type, where one letter appears more frequently than the other. Again, since only two letters are considered, this should not constitute such a difference. Thus, the actual reason for this discrepancy remains unknown.

## 6.3 Cellular Automata Networks as a Reservoir

To complement the discussion of an optimal CAN for reservoir computing, it is crucial to address the overall performance of the system when CAN is used as a reservoir. Admittedly, the results on these seemingly simple tasks are not particularly impressive in isolation. However, it is essential to consider the context of the experiments, as the primary objective was to compare configurations rather than optimize performance. Notably, only one network size was considered, and multiple constraints were placed on network topology and transition rules. Additionally, the system was evaluated solely on a text classification task. So, the fact that the results demonstrate CAN's ability to compute warrants further investigation into this type of reservoir, especially given CAN's versatility and its ability to model a wide range of systems while being

cost-effective and simple to simulate.

This study indicates that when using CAN, after the general network topology is decided, it may be beneficial to impose more constraints on the random network generation. While randomness is important to avoid overfitting, it should be balanced to ensure robust computational capabilities. Ideally, specific topological structures within the network can be exploited for a given transition rule, and the network should be constructed around these structures. The placement of input node connections can also be optimized, as can the method by which input interacts with a CAN. Overall, there are numerous small details that can be fine-tuned to create better networks, and this study provides direction for future investigations.

The echo state property is also essential for a reservoir. For example, Rule 204, where the state of each cell remains unchanged regardless of the predecessors' states, does not exhibit this property as there is no fading memory. Similarly, loops in the network, combined with specific transition rules, can create structures with repeating behaviour and no fading memory. This indicates that the echo state property is not guaranteed for cellular automata networks. Of course, there are also CAN configurations that do possess the echo state property, but demonstrating this is not as straightforward [48].

## 6.4 Reservoir Computing as a Methodological Framework

The final objective was to develop a robust methodological framework for addressing complex and open-ended questions about CAN in a reservoir computing context. This study achieved this by exploiting the modularity of reservoir computing and employing different CAN configurations as the reservoir. Given the ability to draw meaningful conclusions about CAN based on the study's results, it is fair to say that this objective was met. However, certain limitations of the method must be acknowledged.

One significant issue with the method is the high uncertainty in prediction accuracy, as evidenced by the high standard deviation observed in the experiments. To mitigate this, one could use larger training and test datasets or increase the number of experiments conducted. However, these changes would make the process more time-consuming, potentially requiring a further limitation of the study's scope. This problem becomes especially pronounced as the network size increases, presenting the same challenge that reservoir computing aims to address. It is important to note that in this context, each experiment involved training 25,600 models rather than a single one, but this remains a concern.

On the other end, the primary advantage of this approach is the potential to produce benchmarks centred around a common interface, where the only variable is the reservoir. By doing so, it becomes easier to address complex questions about the computation capabilities of dynamical systems. This is similar to standard benchmarks for machine learning models, but with an important difference that

it is not a trainable system that is benchmarked, rather the benchmark addresses randomly generated fixed reservoirs.

## 6.5 Genetic Algorithm

In the pursuit of an optimal network topology and transition rule, a genetic algorithm was also tried in this experiment, but was not expanded upon due to the study's limited time frame. This is an approach that has been employed on cellular automata before [49] and has been extended to cellular automata networks [50]. The general idea is that the best-performing model configurations would become elites and remain unchanged between episodes. The next best configurations would undergo crossover and mutations, while the worst-performing configurations would be replaced by randomly generated new configurations. When run over enough episodes, the algorithm is expected to converge to the optimal network and transition rule.

The prediction accuracy was used to rank the configurations in a single episode. However, it was quickly discovered that the excessive variance in the system would prove a problem, leading to highly performing configurations being replaced due to an unfavourable input sequence for a single episode. This observation also stems from overfitting issues, where the best-performing configurations do not generalize sufficiently. This limitation reduced their practical applicability, and as a result, they were not explored further.

Despite these challenges, several valuable observations emerged from the experiment. The  $\mathcal{N}_{PD-2}$  network demonstrated the highest peak performance. In particular, when the classifier was tasked with distinguishing between completely random text and any of the three structured input types combined in the same input dataset, there were multiple instances where configurations using the  $\mathcal{N}_{PD-2}$  network achieved an accuracy above 90%. This further suggests that, with the correct setup, CAN does showcase promising computation capabilities.

This experiment highlights the potential of the  $\mathcal{N}_{PD-2}$  network and underscores the importance of further research to refine and optimize these models for better performance in reservoir computing contexts.

## 6.6 Future Work

There are several potential directions for future research. One avenue involves revising the logic for the transition rule. Instead of employing a pair-wise calculation and majority vote, an alternative approach could use outer-totalistic rules, focusing on the density of predecessors as in [37]. Moreover, this rule does not have to be confined to elementary cellular automata rules; it can map each density to a possible output or introduce density thresholds for outputs. The outputs can be dependent on the state of the current cell, meaning the state is flipped or remains unchanged, or be independent, where the output is directly mapped to state 0 or 1. Such a rule allows for more options in network topologies, enhances scalability due to the absence of

pair-wise calculations, and creates a more explainable system that solely depends on the density. However, a primary drawback of the approach is that the transition rule no longer considers the order of predecessors. Nevertheless, in a network context, ordered predecessors are not inherently intuitive, which mitigates this concern.

Additionally, this type of model could be applied to different classification problems, using this study as a foundation. Alternatively, the same classification problem could be addressed by considering different input types, input connections into the reservoir, or reservoirs to gain deeper insights into the dynamics of the network models and transition rules. In the latter case, this study could act as a benchmark.

More broadly, the application of reservoir computing to evaluate dynamical systems for various problems holds significant promise. It can be applied to a wider range of natural language processing tasks to better understand how dynamical systems can interact with language, but one can also look at other domains. For instance, since reservoir computing excels at processing sequential data, it can be used to model and predict complex systems in fields such as finance, climate science, and healthcare. In finance, it could enhance predictive models for stock market trends. In climate science, it could improve the accuracy of weather and climate predictions. In healthcare, it could be applied to analyse time-series data from medical sensors to predict patient outcomes or disease progression. Exploring these areas could provide valuable insights and further validate the effectiveness of CAN as a versatile and powerful tool in reservoir computing.

# 7

## Conclusion

This thesis explored the application of reservoir computing with cellular automata networks for text classification tasks. The primary focus was to understand how different network topologies and transition rules impact the computational capabilities of the cellular automata network.

The results demonstrated that cellular automata networks could effectively serve as a reservoir, with certain configurations yielding better performance. Among the various topologies tested, a low-density random network where each cell has exactly two predecessors emerged as the best-performing configuration. This finding suggests that even simple and sparse networks can function as highly effective reservoirs.

An extension of elementary cellular automata rules was applied as the transition rule. Although there was no universally optimal rule across all considered network topologies, rules 29 and 71 showed notable efficacy on the network with two predecessors. Moreover, rule 91 showed the best overall performance with all network topologies considered.

In conclusion, this study contributes to the growing body of knowledge on reservoir computing by demonstrating the viability of cellular automata networks as reservoirs and identifying key factors that influence their performance. Future work could further explore the configuration space of transition rules and network topologies, as well as investigate the applicability of these findings to other types of computational tasks. The flexibility and adaptability of cellular automata networks in reservoir computing highlights their potential as a powerful tool for complex system evaluations, and there is certainly reason to investigate such systems further.



# Bibliography

- [1] A. M. Turing *et al.*, “On computable numbers, with an application to the entscheidungsproblem,” *J. of Math*, vol. 58, no. 345-363, p. 5, 1936.
- [2] Z. Konkoli, S. Stepney, H. Broersma, P. Dini, C. L. Nehaniv, and S. Nichele, “Philosophy of computation,” in *Computational Matter*, S. Stepney, S. Rasmussen, and M. Amos, Eds. Cham: Springer International Publishing, 2018, pp. 153–184, ISBN: 978-3-319-65826-1. DOI: 10.1007/978-3-319-65826-1\_10. [Online]. Available: [https://doi.org/10.1007/978-3-319-65826-1\\_10](https://doi.org/10.1007/978-3-319-65826-1_10).
- [3] J. von Neumann, D. Godfrey, D. F. Hendry, M. D. Godfrey, and S. Smith, “First draft of a report on the edvac,” 1998. [Online]. Available: <https://api.semanticscholar.org/CorpusID:267407823>.
- [4] M. D. Godfrey, “Introduction to “the first draft report on the edvac” by john von neumann,” *Annals of the History of Computing*, vol. 15, no. 1, pp. 11–21, 1993. [Online]. Available: [https://people.csail.mit.edu/brooks/idocs/VonNeumann\\_EDVAC.pdf](https://people.csail.mit.edu/brooks/idocs/VonNeumann_EDVAC.pdf).
- [5] R. Eigenmann and D. J. Lilja, “Von neumann computers,” in *Wiley Encyclopedia of Electrical and Electronics Engineering*. John Wiley & Sons, Ltd, 1999, ISBN: 9780471346081. DOI: <https://doi.org/10.1002/047134608X.W1704>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/047134608X.W1704>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/047134608X.W1704>.
- [6] P. M. Kogge, “Function-based computing and parallelism: A review,” *Parallel Computing*, vol. 2, no. 3, pp. 243–253, 1985, Pararell processing, ISSN: 0167-8191. DOI: [https://doi.org/10.1016/0167-8191\(85\)90006-7](https://doi.org/10.1016/0167-8191(85)90006-7). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0167819185900067>.
- [7] A. Adamatzky, *Advances in unconventional computing: Volume 1: Theory*. Springer, 2016, vol. 22.
- [8] M. Lukoševičius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009, ISSN: 1574-0137. DOI: <https://doi.org/10.1016/j.cosrev.2009.03.005>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1574013709000173>.
- [9] Z. Konkoli, “Reservoir computing,” in *Encyclopedia of Complexity and Systems Science*, R. A. Meyers, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, pp. 1–12, ISBN: 978-3-642-27737-5. DOI: 10.1007/978-3-642-27737-5\_683-1. [Online]. Available: [https://doi.org/10.1007/978-3-642-27737-5\\_683-1](https://doi.org/10.1007/978-3-642-27737-5_683-1).

- [10] Z. Konkoli, S. Nichele, M. Dale, and S. Stepney, “Reservoir computing with computational matter,” in *Computational Matter*. Springer International Publishing, 2018, pp. 269–293, ISBN: 978-3-319-65826-1. DOI: 10.1007/978-3-319-65826-1\_14. [Online]. Available: [https://doi.org/10.1007/978-3-319-65826-1\\_14](https://doi.org/10.1007/978-3-319-65826-1_14).
- [11] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, “Information processing capacity of dynamical systems,” *Scientific Reports*, vol. 2, no. 1, p. 514, 2012, ISSN: 2045-2322. DOI: 10.1038/srep00514. [Online]. Available: <https://doi.org/10.1038/srep00514>.
- [12] S. Stepney, “Nonclassical computation: A dynamical systems perspective,” in *Handbook of Natural Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 1979–2025, ISBN: 978-3-540-92910-9. DOI: 10.1007/978-3-540-92910-9\_59. [Online]. Available: [https://doi.org/10.1007/978-3-540-92910-9\\_59](https://doi.org/10.1007/978-3-540-92910-9_59).
- [13] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *Proceedings of the 30th International Conference on Machine Learning*, S. Dasgupta and D. McAllester, Eds., ser. Proceedings of Machine Learning Research, vol. 28, Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1310–1318. [Online]. Available: <https://proceedings.mlr.press/v28/pascanu13.html>.
- [14] I. Sutskever, *Training recurrent neural networks*. University of Toronto Toronto, ON, Canada, 2013.
- [15] H. Jaeger and H. Haas, “Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication,” *science*, vol. 304, no. 5667, pp. 78–80, 2004.
- [16] H. Jaeger, “The “echo state” approach to analysing and training recurrent neural networks-with an erratum note,” *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, no. 34, p. 13, 2001.
- [17] W. Maass, T. Natschläger, and H. Markram, “Real-time computing without stable states: A new framework for neural computation based on perturbations,” *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.
- [18] T. Natschläger, W. Maass, and H. Markram, “The “liquid computer”: A novel strategy for real-time computing on time series,” *Telematik*, vol. 8, no. 1, pp. 39–43, 2002.
- [19] N. Bertschinger and T. Natschläger, “Real-time computation at the edge of chaos in recurrent neural networks,” *Neural Computation*, vol. 16, no. 7, pp. 1413–1436, Jul. 2004, ISSN: 0899-7667. DOI: 10.1162/089976604323057443. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/15165396>.
- [20] C. Fernando and S. Sojakka, “Pattern recognition in a bucket,” in *European conference on artificial life*, Springer, 2003, pp. 588–597.
- [21] G. Tanaka, T. Yamane, J. B. Héroux, *et al.*, “Recent advances in physical reservoir computing: A review,” *Neural Networks*, vol. 115, pp. 100–123, 2019, ISSN: 0893-6080. DOI: <https://doi.org/10.1016/j.neunet.2019.03.005>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608019300784>.

- 
- [22] C. Du, F. Cai, M. A. Zidan, W. Ma, S. H. Lee, and W. D. Lu, “Reservoir computing using dynamic memristors for temporal information processing,” *Nature communications*, vol. 8, no. 1, p. 2204, 2017.
- [23] G. Van der Sande, D. Brunner, and M. C. Soriano, “Advances in photonic reservoir computing,” *Nanophotonics*, vol. 6, no. 3, pp. 561–576, 2017.
- [24] J. Torrejon, M. Riou, F. A. Araujo, *et al.*, “Neuromorphic computing with nanoscale spintronic oscillators,” *Nature*, vol. 547, no. 7664, pp. 428–431, 2017.
- [25] P. F. Dominey, “Recurrent temporal networks and language acquisition—from corticostriatal neurophysiology to reservoir computing,” *Frontiers in psychology*, vol. 4, p. 500, 2013.
- [26] R. Pfeifer and J. Bongard, *How the body shapes the way we think: a new view of intelligence*. MIT press, 2006.
- [27] H. Hauser, A. J. Ijspeert, R. M. Füchslin, R. Pfeifer, and W. Maass, “Towards a theoretical foundation for morphological computation with compliant bodies,” *Biological Cybernetics*, vol. 105, no. 5, pp. 355–370, 2011, ISSN: 1432-0770. DOI: 10.1007/s00422-012-0471-0. [Online]. Available: <https://doi.org/10.1007/s00422-012-0471-0>.
- [28] J. von Neumann, “The general and logical theory of automata,” 1963. [Online]. Available: <https://api.semanticscholar.org/CorpusID:14588544>.
- [29] S. Wolfram, “Cellular automata as simple self-organizing systems,” 1982.
- [30] S. Wolfram, “Computation theory of cellular automata,” *Communications in Mathematical Physics*, vol. 96, no. 1, pp. 15–57, 1984, ISSN: 1432-0916. DOI: 10.1007/BF01217347. [Online]. Available: <https://doi.org/10.1007/BF01217347>.
- [31] A. Adamatzky, *Game of life cellular automata*. Springer, 2010, vol. 1.
- [32] P. Rendell, “Turing universality of the game of life,” in *Collision-Based Computing*, A. Adamatzky, Ed. London: Springer London, 2002, pp. 513–539, ISBN: 978-1-4471-0129-1. DOI: 10.1007/978-1-4471-0129-1\_18. [Online]. Available: [https://doi.org/10.1007/978-1-4471-0129-1\\_18](https://doi.org/10.1007/978-1-4471-0129-1_18).
- [33] S. Wolfram, *A New Kind of Science*. Champaign, IL: Wolfram Media, 2002.
- [34] E. W. Weisstein, *Elementary cellular automaton*, <https://mathworld.wolfram.com/ElementaryCellularAutomaton.html>, From MathWorld—A Wolfram Web Resource. [Online]. Available: <https://mathworld.wolfram.com/ElementaryCellularAutomaton.html> (visited on 05/14/2024).
- [35] M. Cook *et al.*, “Universality in elementary cellular automata,” *Complex systems*, vol. 15, no. 1, pp. 1–40, 2004.
- [36] D. O’Sullivan, “Graph-cellular automata: A generalised discrete urban and regional model,” *Environment and planning B: planning and design*, vol. 28, no. 5, pp. 687–705, 2001.
- [37] C. Marr and M.-T. Hütt, “Outer-totalistic cellular automata on graphs,” *Physics Letters A*, vol. 373, no. 5, pp. 546–549, 2009.
- [38] X.-S. Yang and Y. Z. Yang, “Cellular automata networks,” *Proceedings of unconventional computing*, pp. 280–302, 2007.
- [39] O. Yilmaz, “Reservoir computing using cellular automata,” *arXiv preprint arXiv:1410.0162*, 2014.

- [40] O. Yilmaz, “Connectionist-symbolic machine intelligence using cellular automata based reservoir-hyperdimensional computing,” *arXiv preprint arXiv:1503.00851*, 2015.
- [41] D. Kleyko, S. Khan, E. Osipov, and S.-P. Yong, “Modality classification of medical images with distributed representations based on cellular automata reservoir computing,” in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, IEEE, 2017, pp. 1053–1056.
- [42] S. Nichele and A. Molund, “Deep reservoir computing using cellular automata,” *arXiv preprint arXiv:1703.02806*, 2017.
- [43] I. Del Amo and Z. Konkoli, “On applications of cellular automata memristor networks for reservoir computing: Classifying protein toxicity,” *International Journal of Unconventional Computing*, vol. 17, 2022.
- [44] P. Erdős, A. Rényi, *et al.*, “On the evolution of random graphs,” *Publ. math. inst. hung. acad. sci.*, vol. 5, no. 1, pp. 17–60, 1960.
- [45] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [46] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [47] C. E. Shannon, “A mathematical theory of communication,” *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948. DOI: 10.1002/j.1538-7305.1948.tb01338.x. [Online]. Available: <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>.
- [48] I. B. Yildiz, H. Jaeger, and S. J. Kiebel, “Re-visiting the echo state property,” *Neural Networks*, vol. 35, pp. 1–9, 2012, ISSN: 0893-6080. DOI: <https://doi.org/10.1016/j.neunet.2012.07.005>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608012001852>.
- [49] M. Mitchell, J. P. Crutchfield, R. Das, *et al.*, “Evolving cellular automata with genetic algorithms: A review of recent work,” in *Proceedings of the First international conference on evolutionary computation and its applications (EvCA ’96)*, Moscow, vol. 8, 1996, pp. 23–30.
- [50] M. SIPPER, “Evolving uniform and non-uniform cellular automata networks,” in *Annual Reviews of Computational Physics V*, pp. 243–285. DOI: 10.1142/9789812819444\_0006. [Online]. Available: [https://www.worldscientific.com/doi/abs/10.1142/9789812819444\\_0006](https://www.worldscientific.com/doi/abs/10.1142/9789812819444_0006).

# A

## Appendix 1

Table A.1: Full results for  $\mathcal{N}_{PD-2}$ ,  $\mathcal{N}_{PD-3}$ , and  $\mathcal{N}_{PD-4}$  from experiments, including the rank, mean accuracy, and standard deviation for each rule and network pair. Additionally, average results over all three networks is also included. The rules are ordered based on the overall results.

| $R_i$      | Rank | $\mu(\mathcal{N}_{PD2})$ | $\sigma^2(\mathcal{N}_{PD2})$ | Rank | $\mu(\mathcal{N}_{PD3})$ | $\sigma^2(\mathcal{N}_{PD3})$ | Rank | $\mu(\mathcal{N}_{PD4})$ | $\sigma^2(\mathcal{N}_{PD4})$ | Rank | $\mu(\cdot)$ | $\sigma^2(\cdot)$ |
|------------|------|--------------------------|-------------------------------|------|--------------------------|-------------------------------|------|--------------------------|-------------------------------|------|--------------|-------------------|
| <b>91</b>  | 61   | 0.678                    | 0.078                         | 2    | 0.717                    | 0.1                           | 1    | 0.771                    | 0.103                         | 1    | 0.722        | 0.093             |
| <b>123</b> | 53   | 0.681                    | 0.105                         | 3    | 0.716                    | 0.098                         | 4    | 0.742                    | 0.096                         | 2    | 0.713        | 0.1               |
| <b>94</b>  | 41   | 0.699                    | 0.124                         | 12   | 0.688                    | 0.105                         | 6    | 0.707                    | 0.097                         | 3    | 0.698        | 0.109             |
| <b>144</b> | 3    | 0.755                    | 0.15                          | 20   | 0.686                    | 0.165                         | 19   | 0.638                    | 0.161                         | 4    | 0.693        | 0.159             |
| <b>16</b>  | 4    | 0.754                    | 0.151                         | 24   | 0.683                    | 0.168                         | 17   | 0.641                    | 0.166                         | 5    | 0.693        | 0.162             |
| <b>56</b>  | 6    | 0.744                    | 0.119                         | 19   | 0.686                    | 0.137                         | 16   | 0.644                    | 0.132                         | 6    | 0.691        | 0.129             |
| <b>2</b>   | 5    | 0.746                    | 0.151                         | 16   | 0.686                    | 0.169                         | 20   | 0.636                    | 0.163                         | 7    | 0.689        | 0.161             |
| <b>98</b>  | 7    | 0.742                    | 0.114                         | 15   | 0.687                    | 0.142                         | 18   | 0.639                    | 0.124                         | 8    | 0.689        | 0.127             |
| <b>126</b> | 59   | 0.679                    | 0.078                         | 30   | 0.675                    | 0.08                          | 5    | 0.707                    | 0.086                         | 9    | 0.687        | 0.081             |
| <b>116</b> | 14   | 0.731                    | 0.125                         | 36   | 0.668                    | 0.135                         | 12   | 0.658                    | 0.094                         | 10   | 0.686        | 0.118             |
| <b>130</b> | 8    | 0.74                     | 0.153                         | 27   | 0.681                    | 0.168                         | 21   | 0.635                    | 0.164                         | 11   | 0.685        | 0.162             |
| <b>30</b>  | 56   | 0.679                    | 0.092                         | 7    | 0.694                    | 0.088                         | 9    | 0.674                    | 0.095                         | 12   | 0.683        | 0.092             |
| <b>24</b>  | 10   | 0.738                    | 0.149                         | 29   | 0.678                    | 0.158                         | 22   | 0.628                    | 0.155                         | 13   | 0.681        | 0.154             |
| <b>46</b>  | 21   | 0.72                     | 0.122                         | 45   | 0.661                    | 0.131                         | 11   | 0.659                    | 0.093                         | 14   | 0.68         | 0.115             |
| <b>66</b>  | 15   | 0.73                     | 0.147                         | 28   | 0.68                     | 0.163                         | 25   | 0.625                    | 0.153                         | 15   | 0.679        | 0.154             |
| <b>118</b> | 46   | 0.696                    | 0.102                         | 25   | 0.682                    | 0.13                          | 14   | 0.657                    | 0.103                         | 16   | 0.678        | 0.112             |
| <b>62</b>  | 48   | 0.693                    | 0.109                         | 26   | 0.681                    | 0.139                         | 13   | 0.658                    | 0.095                         | 17   | 0.678        | 0.114             |
| <b>107</b> | 102  | 0.633                    | 0.076                         | 60   | 0.648                    | 0.089                         | 3    | 0.75                     | 0.099                         | 18   | 0.677        | 0.088             |
| <b>121</b> | 100  | 0.635                    | 0.072                         | 68   | 0.642                    | 0.083                         | 2    | 0.751                    | 0.099                         | 19   | 0.676        | 0.085             |
| <b>86</b>  | 73   | 0.664                    | 0.088                         | 17   | 0.686                    | 0.096                         | 10   | 0.663                    | 0.098                         | 20   | 0.671        | 0.094             |
| <b>48</b>  | 9    | 0.739                    | 0.151                         | 33   | 0.67                     | 0.162                         | 35   | 0.59                     | 0.131                         | 21   | 0.666        | 0.148             |
| <b>34</b>  | 16   | 0.728                    | 0.152                         | 34   | 0.67                     | 0.163                         | 34   | 0.593                    | 0.132                         | 22   | 0.664        | 0.149             |
| <b>131</b> | 38   | 0.702                    | 0.106                         | 5    | 0.699                    | 0.136                         | 36   | 0.59                     | 0.063                         | 23   | 0.663        | 0.102             |
| <b>29</b>  | 2    | 0.757                    | 0.106                         | 11   | 0.689                    | 0.129                         | 72   | 0.537                    | 0.08                          | 24   | 0.661        | 0.105             |
| <b>145</b> | 40   | 0.699                    | 0.101                         | 9    | 0.693                    | 0.137                         | 38   | 0.587                    | 0.067                         | 25   | 0.66         | 0.101             |
| <b>71</b>  | 1    | 0.759                    | 0.111                         | 14   | 0.687                    | 0.138                         | 79   | 0.53                     | 0.071                         | 26   | 0.658        | 0.107             |
| <b>125</b> | 26   | 0.715                    | 0.1                           | 8    | 0.693                    | 0.096                         | 49   | 0.558                    | 0.066                         | 27   | 0.655        | 0.088             |
| <b>65</b>  | 30   | 0.712                    | 0.095                         | 18   | 0.686                    | 0.095                         | 47   | 0.565                    | 0.104                         | 28   | 0.654        | 0.098             |
| <b>176</b> | 37   | 0.705                    | 0.147                         | 31   | 0.671                    | 0.159                         | 39   | 0.584                    | 0.105                         | 29   | 0.653        | 0.137             |
| <b>122</b> | 118  | 0.609                    | 0.081                         | 50   | 0.658                    | 0.077                         | 8    | 0.692                    | 0.082                         | 30   | 0.653        | 0.08              |
| <b>111</b> | 32   | 0.71                     | 0.094                         | 23   | 0.684                    | 0.094                         | 48   | 0.56                     | 0.064                         | 31   | 0.652        | 0.084             |
| <b>9</b>   | 29   | 0.712                    | 0.092                         | 21   | 0.685                    | 0.093                         | 51   | 0.557                    | 0.108                         | 32   | 0.651        | 0.098             |
| <b>31</b>  | 12   | 0.733                    | 0.154                         | 32   | 0.671                    | 0.168                         | 65   | 0.542                    | 0.09                          | 33   | 0.649        | 0.137             |
| <b>162</b> | 33   | 0.71                     | 0.152                         | 53   | 0.655                    | 0.158                         | 44   | 0.577                    | 0.102                         | 34   | 0.647        | 0.137             |
| <b>47</b>  | 19   | 0.725                    | 0.145                         | 51   | 0.657                    | 0.154                         | 58   | 0.551                    | 0.1                           | 35   | 0.644        | 0.133             |
| <b>133</b> | 45   | 0.697                    | 0.121                         | 10   | 0.692                    | 0.109                         | 69   | 0.54                     | 0.05                          | 36   | 0.643        | 0.093             |
| <b>11</b>  | 18   | 0.726                    | 0.141                         | 44   | 0.662                    | 0.154                         | 68   | 0.54                     | 0.095                         | 37   | 0.643        | 0.13              |
| <b>117</b> | 31   | 0.711                    | 0.143                         | 41   | 0.664                    | 0.154                         | 54   | 0.552                    | 0.101                         | 38   | 0.642        | 0.133             |
| <b>87</b>  | 22   | 0.719                    | 0.156                         | 38   | 0.665                    | 0.163                         | 64   | 0.542                    | 0.095                         | 39   | 0.642        | 0.138             |
| <b>33</b>  | 49   | 0.692                    | 0.102                         | 1    | 0.722                    | 0.098                         | 104  | 0.511                    | 0.032                         | 40   | 0.642        | 0.077             |
| <b>81</b>  | 23   | 0.718                    | 0.147                         | 39   | 0.665                    | 0.151                         | 67   | 0.541                    | 0.103                         | 41   | 0.641        | 0.134             |
| <b>63</b>  | 27   | 0.714                    | 0.155                         | 59   | 0.648                    | 0.152                         | 62   | 0.545                    | 0.088                         | 42   | 0.636        | 0.132             |
| <b>7</b>   | 11   | 0.733                    | 0.156                         | 35   | 0.668                    | 0.165                         | 147  | 0.501                    | 0.034                         | 43   | 0.634        | 0.118             |
| <b>3</b>   | 13   | 0.733                    | 0.15                          | 42   | 0.663                    | 0.159                         | 114  | 0.506                    | 0.034                         | 44   | 0.634        | 0.114             |
| <b>129</b> | 54   | 0.68                     | 0.082                         | 22   | 0.685                    | 0.084                         | 80   | 0.53                     | 0.046                         | 45   | 0.631        | 0.071             |
| <b>119</b> | 34   | 0.706                    | 0.154                         | 65   | 0.644                    | 0.146                         | 66   | 0.541                    | 0.089                         | 46   | 0.631        | 0.13              |
| <b>209</b> | 17   | 0.726                    | 0.126                         | 49   | 0.659                    | 0.133                         | 117  | 0.505                    | 0.026                         | 47   | 0.63         | 0.095             |
| <b>135</b> | 58   | 0.679                    | 0.097                         | 6    | 0.694                    | 0.091                         | 92   | 0.517                    | 0.035                         | 48   | 0.63         | 0.074             |
| <b>17</b>  | 20   | 0.723                    | 0.153                         | 43   | 0.662                    | 0.153                         | 126  | 0.503                    | 0.034                         | 49   | 0.629        | 0.114             |
| <b>38</b>  | 87   | 0.645                    | 0.101                         | 70   | 0.642                    | 0.116                         | 30   | 0.598                    | 0.082                         | 50   | 0.628        | 0.1               |
| <b>37</b>  | 62   | 0.677                    | 0.074                         | 4    | 0.707                    | 0.091                         | 183  | 0.5                      | 0.035                         | 51   | 0.628        | 0.066             |
| <b>21</b>  | 25   | 0.716                    | 0.154                         | 37   | 0.666                    | 0.163                         | 166  | 0.5                      | 0.036                         | 52   | 0.627        | 0.118             |
| <b>80</b>  | 64   | 0.677                    | 0.17                          | 90   | 0.609                    | 0.157                         | 33   | 0.595                    | 0.144                         | 53   | 0.627        | 0.157             |
| <b>149</b> | 67   | 0.668                    | 0.084                         | 13   | 0.688                    | 0.094                         | 88   | 0.523                    | 0.038                         | 54   | 0.626        | 0.072             |
| <b>112</b> | 77   | 0.66                     | 0.158                         | 87   | 0.618                    | 0.15                          | 31   | 0.597                    | 0.133                         | 55   | 0.625        | 0.147             |
| <b>227</b> | 24   | 0.717                    | 0.135                         | 52   | 0.657                    | 0.142                         | 204  | 0.5                      | 0                             | 56   | 0.625        | 0.092             |
| <b>52</b>  | 94   | 0.639                    | 0.109                         | 73   | 0.639                    | 0.119                         | 32   | 0.595                    | 0.079                         | 57   | 0.624        | 0.102             |
| <b>139</b> | 35   | 0.705                    | 0.133                         | 46   | 0.66                     | 0.137                         | 118  | 0.505                    | 0.029                         | 58   | 0.623        | 0.1               |
| <b>120</b> | 89   | 0.642                    | 0.132                         | 94   | 0.6                      | 0.099                         | 23   | 0.628                    | 0.111                         | 59   | 0.623        | 0.114             |
| <b>90</b>  | 187  | 0.507                    | 0.064                         | 47   | 0.66                     | 0.085                         | 7    | 0.7                      | 0.101                         | 60   | 0.622        | 0.084             |
| <b>185</b> | 28   | 0.713                    | 0.132                         | 54   | 0.653                    | 0.141                         | 204  | 0.5                      | 0                             | 61   | 0.622        | 0.091             |

# A. Appendix 1

|     |     |       |       |     |       |       |     |       |       |     |       |       |
|-----|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|-------|-------|
| 82  | 72  | 0.665 | 0.085 | 62  | 0.646 | 0.1   | 53  | 0.556 | 0.085 | 62  | 0.622 | 0.09  |
| 41  | 93  | 0.639 | 0.077 | 66  | 0.644 | 0.087 | 41  | 0.58  | 0.068 | 63  | 0.621 | 0.077 |
| 42  | 85  | 0.645 | 0.154 | 89  | 0.612 | 0.149 | 29  | 0.603 | 0.139 | 64  | 0.62  | 0.147 |
| 10  | 80  | 0.655 | 0.17  | 88  | 0.614 | 0.155 | 37  | 0.589 | 0.152 | 65  | 0.619 | 0.159 |
| 226 | 36  | 0.705 | 0.129 | 56  | 0.652 | 0.131 | 149 | 0.501 | 0.008 | 66  | 0.619 | 0.089 |
| 25  | 63  | 0.677 | 0.094 | 81  | 0.632 | 0.107 | 60  | 0.549 | 0.099 | 67  | 0.619 | 0.1   |
| 97  | 103 | 0.632 | 0.074 | 64  | 0.645 | 0.083 | 42  | 0.579 | 0.07  | 68  | 0.619 | 0.076 |
| 184 | 43  | 0.699 | 0.127 | 57  | 0.652 | 0.131 | 161 | 0.501 | 0.009 | 69  | 0.617 | 0.089 |
| 61  | 65  | 0.676 | 0.1   | 85  | 0.623 | 0.109 | 56  | 0.552 | 0.096 | 70  | 0.617 | 0.101 |
| 26  | 84  | 0.649 | 0.087 | 58  | 0.649 | 0.101 | 57  | 0.551 | 0.086 | 71  | 0.616 | 0.091 |
| 152 | 52  | 0.682 | 0.136 | 74  | 0.638 | 0.137 | 82  | 0.529 | 0.066 | 72  | 0.616 | 0.113 |
| 67  | 60  | 0.678 | 0.095 | 82  | 0.629 | 0.108 | 70  | 0.54  | 0.09  | 73  | 0.615 | 0.098 |
| 106 | 98  | 0.636 | 0.13  | 101 | 0.589 | 0.096 | 27  | 0.62  | 0.106 | 74  | 0.615 | 0.111 |
| 246 | 39  | 0.701 | 0.162 | 71  | 0.641 | 0.159 | 204 | 0.5   | 0     | 75  | 0.614 | 0.107 |
| 194 | 66  | 0.671 | 0.133 | 76  | 0.637 | 0.138 | 77  | 0.53  | 0.062 | 76  | 0.613 | 0.111 |
| 191 | 44  | 0.698 | 0.16  | 72  | 0.64  | 0.155 | 204 | 0.5   | 0     | 77  | 0.613 | 0.105 |
| 103 | 70  | 0.667 | 0.098 | 86  | 0.622 | 0.105 | 59  | 0.55  | 0.097 | 78  | 0.613 | 0.1   |
| 247 | 42  | 0.699 | 0.165 | 75  | 0.637 | 0.155 | 204 | 0.5   | 0     | 79  | 0.612 | 0.107 |
| 190 | 50  | 0.688 | 0.165 | 67  | 0.644 | 0.158 | 204 | 0.5   | 0     | 80  | 0.611 | 0.108 |
| 231 | 47  | 0.694 | 0.153 | 78  | 0.635 | 0.154 | 185 | 0.5   | 0.001 | 81  | 0.61  | 0.102 |
| 189 | 51  | 0.688 | 0.155 | 80  | 0.632 | 0.154 | 204 | 0.5   | 0     | 82  | 0.607 | 0.103 |
| 181 | 71  | 0.666 | 0.085 | 63  | 0.645 | 0.102 | 161 | 0.501 | 0.005 | 83  | 0.604 | 0.064 |
| 180 | 82  | 0.652 | 0.104 | 55  | 0.652 | 0.116 | 133 | 0.502 | 0.014 | 84  | 0.602 | 0.078 |
| 208 | 76  | 0.66  | 0.163 | 91  | 0.606 | 0.148 | 73  | 0.534 | 0.072 | 85  | 0.6   | 0.128 |
| 167 | 83  | 0.652 | 0.083 | 69  | 0.642 | 0.097 | 174 | 0.5   | 0.002 | 86  | 0.598 | 0.061 |
| 124 | 130 | 0.593 | 0.097 | 105 | 0.577 | 0.092 | 26  | 0.621 | 0.095 | 87  | 0.597 | 0.095 |
| 110 | 125 | 0.598 | 0.099 | 104 | 0.577 | 0.085 | 28  | 0.608 | 0.091 | 88  | 0.595 | 0.092 |
| 166 | 96  | 0.636 | 0.105 | 61  | 0.646 | 0.11  | 154 | 0.501 | 0.009 | 89  | 0.594 | 0.075 |
| 109 | 105 | 0.631 | 0.083 | 208 | 0.5   | 0.032 | 15  | 0.65  | 0.1   | 90  | 0.594 | 0.072 |
| 210 | 86  | 0.645 | 0.107 | 77  | 0.635 | 0.113 | 185 | 0.5   | 0.021 | 91  | 0.593 | 0.08  |
| 45  | 69  | 0.667 | 0.092 | 131 | 0.533 | 0.059 | 43  | 0.577 | 0.12  | 92  | 0.592 | 0.09  |
| 161 | 117 | 0.611 | 0.08  | 48  | 0.659 | 0.084 | 137 | 0.502 | 0.007 | 93  | 0.591 | 0.057 |
| 154 | 107 | 0.628 | 0.109 | 79  | 0.635 | 0.113 | 115 | 0.506 | 0.021 | 94  | 0.589 | 0.081 |
| 138 | 92  | 0.64  | 0.153 | 96  | 0.598 | 0.146 | 81  | 0.529 | 0.07  | 95  | 0.589 | 0.123 |
| 211 | 104 | 0.632 | 0.11  | 83  | 0.628 | 0.117 | 122 | 0.504 | 0.022 | 96  | 0.588 | 0.083 |
| 245 | 78  | 0.658 | 0.169 | 92  | 0.603 | 0.153 | 204 | 0.5   | 0     | 97  | 0.587 | 0.107 |
| 101 | 79  | 0.656 | 0.088 | 133 | 0.532 | 0.05  | 46  | 0.57  | 0.118 | 98  | 0.586 | 0.085 |
| 155 | 106 | 0.628 | 0.104 | 84  | 0.626 | 0.117 | 123 | 0.504 | 0.023 | 99  | 0.586 | 0.082 |
| 175 | 90  | 0.642 | 0.168 | 95  | 0.599 | 0.147 | 204 | 0.5   | 0     | 100 | 0.58  | 0.105 |
| 225 | 101 | 0.634 | 0.127 | 93  | 0.601 | 0.099 | 178 | 0.5   | 0.002 | 101 | 0.578 | 0.076 |
| 75  | 68  | 0.667 | 0.093 | 134 | 0.531 | 0.054 | 84  | 0.526 | 0.053 | 102 | 0.575 | 0.066 |
| 244 | 99  | 0.635 | 0.156 | 100 | 0.589 | 0.14  | 225 | 0.5   | 0.003 | 103 | 0.575 | 0.099 |
| 169 | 109 | 0.628 | 0.131 | 98  | 0.591 | 0.09  | 181 | 0.5   | 0.002 | 104 | 0.573 | 0.075 |
| 230 | 108 | 0.628 | 0.139 | 99  | 0.59  | 0.127 | 174 | 0.5   | 0.004 | 105 | 0.573 | 0.09  |
| 146 | 74  | 0.663 | 0.08  | 140 | 0.524 | 0.038 | 75  | 0.531 | 0.043 | 106 | 0.572 | 0.054 |
| 95  | 113 | 0.617 | 0.107 | 119 | 0.55  | 0.049 | 61  | 0.548 | 0.042 | 107 | 0.572 | 0.066 |
| 188 | 111 | 0.619 | 0.133 | 97  | 0.593 | 0.13  | 179 | 0.5   | 0.003 | 108 | 0.571 | 0.088 |
| 89  | 81  | 0.652 | 0.087 | 132 | 0.533 | 0.048 | 89  | 0.522 | 0.045 | 109 | 0.569 | 0.06  |
| 174 | 112 | 0.619 | 0.15  | 102 | 0.586 | 0.136 | 188 | 0.5   | 0.003 | 110 | 0.568 | 0.096 |
| 18  | 57  | 0.679 | 0.092 | 154 | 0.517 | 0.033 | 120 | 0.505 | 0.027 | 111 | 0.567 | 0.051 |
| 193 | 119 | 0.604 | 0.098 | 103 | 0.584 | 0.09  | 112 | 0.506 | 0.031 | 112 | 0.565 | 0.073 |
| 183 | 55  | 0.68  | 0.092 | 158 | 0.513 | 0.033 | 204 | 0.5   | 0     | 113 | 0.564 | 0.042 |
| 241 | 114 | 0.612 | 0.133 | 107 | 0.574 | 0.111 | 204 | 0.5   | 0     | 114 | 0.562 | 0.081 |
| 137 | 121 | 0.603 | 0.103 | 106 | 0.576 | 0.085 | 111 | 0.507 | 0.028 | 115 | 0.562 | 0.072 |
| 182 | 75  | 0.661 | 0.085 | 149 | 0.52  | 0.036 | 204 | 0.5   | 0     | 116 | 0.56  | 0.04  |
| 165 | 180 | 0.509 | 0.065 | 40  | 0.665 | 0.086 | 176 | 0.5   | 0.002 | 117 | 0.558 | 0.051 |
| 242 | 116 | 0.612 | 0.109 | 116 | 0.561 | 0.085 | 204 | 0.5   | 0     | 118 | 0.558 | 0.065 |
| 22  | 127 | 0.593 | 0.076 | 147 | 0.521 | 0.037 | 50  | 0.557 | 0.065 | 119 | 0.557 | 0.059 |
| 186 | 122 | 0.603 | 0.106 | 112 | 0.569 | 0.092 | 204 | 0.5   | 0     | 120 | 0.557 | 0.066 |
| 15  | 143 | 0.562 | 0.068 | 118 | 0.551 | 0.068 | 52  | 0.556 | 0.1   | 121 | 0.556 | 0.079 |
| 171 | 126 | 0.596 | 0.124 | 110 | 0.571 | 0.112 | 204 | 0.5   | 0     | 122 | 0.556 | 0.079 |
| 20  | 138 | 0.57  | 0.089 | 114 | 0.566 | 0.092 | 78  | 0.53  | 0.072 | 123 | 0.555 | 0.084 |
| 6   | 139 | 0.569 | 0.08  | 109 | 0.572 | 0.095 | 87  | 0.523 | 0.057 | 124 | 0.555 | 0.077 |
| 243 | 129 | 0.593 | 0.095 | 113 | 0.568 | 0.091 | 204 | 0.5   | 0     | 125 | 0.554 | 0.062 |
| 187 | 131 | 0.59  | 0.097 | 111 | 0.57  | 0.093 | 204 | 0.5   | 0     | 126 | 0.553 | 0.063 |
| 88  | 88  | 0.643 | 0.159 | 160 | 0.512 | 0.035 | 129 | 0.502 | 0.022 | 127 | 0.553 | 0.072 |
| 85  | 142 | 0.562 | 0.075 | 125 | 0.543 | 0.071 | 55  | 0.552 | 0.102 | 128 | 0.552 | 0.083 |
| 5   | 115 | 0.612 | 0.103 | 124 | 0.544 | 0.049 | 251 | 0.498 | 0.028 | 129 | 0.551 | 0.06  |
| 74  | 97  | 0.636 | 0.157 | 162 | 0.511 | 0.035 | 150 | 0.501 | 0.025 | 130 | 0.55  | 0.072 |
| 229 | 91  | 0.64  | 0.161 | 168 | 0.509 | 0.032 | 227 | 0.5   | 0.003 | 130 | 0.55  | 0.065 |
| 127 | 146 | 0.556 | 0.055 | 123 | 0.544 | 0.045 | 63  | 0.544 | 0.04  | 132 | 0.548 | 0.047 |
| 134 | 153 | 0.546 | 0.075 | 108 | 0.572 | 0.094 | 85  | 0.524 | 0.051 | 133 | 0.548 | 0.073 |
| 73  | 95  | 0.638 | 0.082 | 193 | 0.501 | 0.034 | 132 | 0.502 | 0.023 | 134 | 0.547 | 0.046 |
| 177 | 120 | 0.603 | 0.129 | 144 | 0.523 | 0.083 | 101 | 0.512 | 0.04  | 135 | 0.546 | 0.084 |
| 58  | 124 | 0.599 | 0.123 | 141 | 0.524 | 0.075 | 95  | 0.515 | 0.067 | 136 | 0.546 | 0.088 |
| 148 | 147 | 0.554 | 0.077 | 115 | 0.562 | 0.093 | 90  | 0.52  | 0.054 | 137 | 0.545 | 0.075 |
| 173 | 110 | 0.624 | 0.156 | 167 | 0.509 | 0.031 | 221 | 0.5   | 0.001 | 138 | 0.545 | 0.063 |
| 105 | 191 | 0.506 | 0.053 | 244 | 0.499 | 0.031 | 24  | 0.627 | 0.083 | 139 | 0.544 | 0.056 |
| 114 | 123 | 0.6   | 0.128 | 153 | 0.518 | 0.076 | 97  | 0.514 | 0.065 | 140 | 0.544 | 0.09  |
| 163 | 127 | 0.593 | 0.125 | 143 | 0.523 | 0.072 | 107 | 0.51  | 0.031 | 141 | 0.542 | 0.076 |
| 27  | 133 | 0.585 | 0.124 | 138 | 0.525 | 0.082 | 94  | 0.516 | 0.069 | 142 | 0.542 | 0.091 |
| 13  | 137 | 0.572 | 0.098 | 156 | 0.514 | 0.038 | 71  | 0.539 | 0.073 | 143 | 0.541 | 0.07  |
| 151 | 132 | 0.59  | 0.075 | 151 | 0.519 | 0.037 | 121 | 0.504 | 0.02  | 144 | 0.538 | 0.044 |
| 83  | 136 | 0.577 | 0.124 | 152 | 0.519 | 0.078 | 99  | 0.513 | 0.066 | 145 | 0.537 | 0.089 |
| 69  | 141 | 0.563 | 0.095 | 156 | 0.514 | 0.041 | 74  | 0.532 | 0.063 | 146 | 0.536 | 0.066 |
| 39  | 134 | 0.581 | 0.121 | 139 | 0.524 | 0.08  | 130 | 0.502 | 0.032 | 147 | 0.536 | 0.077 |
| 57  | 149 | 0.55  | 0.072 | 127 | 0.539 | 0.066 | 91  | 0.517 | 0.066 | 148 | 0.536 | 0.068 |
| 1   | 144 | 0.562 | 0.054 | 126 | 0.542 | 0.04  | 168 | 0.5   | 0.021 | 149 | 0.534 | 0.039 |
| 159 | 151 | 0.549 | 0.07  | 117 | 0.553 | 0.083 | 144 | 0.501 | 0.016 | 150 | 0.534 | 0.056 |
| 79  | 140 | 0.564 | 0.091 | 163 | 0.511 | 0.036 | 83  | 0.528 | 0.048 | 151 | 0.534 | 0.058 |
| 215 | 150 | 0.55  | 0.082 | 120 | 0.549 | 0.081 | 138 | 0.502 | 0.02  | 152 | 0.534 | 0.061 |
| 99  | 155 | 0.546 | 0.072 | 129 | 0.537 | 0.07  | 93  | 0.516 | 0.065 | 153 | 0.533 | 0.069 |
| 93  | 145 | 0.559 | 0.092 | 159 | 0.512 | 0.038 | 86  | 0.524 | 0.047 | 154 | 0.532 | 0.059 |
| 53  | 135 | 0.578 | 0.119 | 146 | 0.521 | 0.083 | 256 | 0.496 | 0.036 | 155 | 0.532 | 0.079 |

|     |     |       |       |     |       |       |     |       |       |     |       |       |
|-----|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|-------|-------|
| 214 | 152 | 0.548 | 0.079 | 122 | 0.546 | 0.082 | 182 | 0.5   | 0.021 | 156 | 0.532 | 0.061 |
| 170 | 148 | 0.554 | 0.072 | 130 | 0.536 | 0.063 | 229 | 0.5   | 0.01  | 157 | 0.53  | 0.048 |
| 158 | 159 | 0.539 | 0.066 | 121 | 0.547 | 0.083 | 134 | 0.502 | 0.019 | 158 | 0.529 | 0.056 |
| 102 | 204 | 0.502 | 0.046 | 186 | 0.502 | 0.042 | 40  | 0.583 | 0.085 | 159 | 0.529 | 0.058 |
| 240 | 154 | 0.546 | 0.068 | 128 | 0.539 | 0.069 | 157 | 0.501 | 0.008 | 160 | 0.529 | 0.048 |
| 60  | 202 | 0.503 | 0.046 | 172 | 0.503 | 0.043 | 45  | 0.577 | 0.084 | 161 | 0.528 | 0.058 |
| 14  | 158 | 0.54  | 0.066 | 135 | 0.531 | 0.06  | 103 | 0.511 | 0.04  | 162 | 0.527 | 0.055 |
| 147 | 156 | 0.545 | 0.065 | 189 | 0.502 | 0.034 | 76  | 0.531 | 0.045 | 163 | 0.526 | 0.048 |
| 84  | 160 | 0.539 | 0.065 | 136 | 0.525 | 0.056 | 106 | 0.51  | 0.044 | 164 | 0.525 | 0.055 |
| 28  | 163 | 0.533 | 0.058 | 145 | 0.521 | 0.051 | 108 | 0.509 | 0.039 | 165 | 0.521 | 0.049 |
| 143 | 161 | 0.535 | 0.068 | 137 | 0.525 | 0.054 | 156 | 0.501 | 0.021 | 166 | 0.52  | 0.047 |
| 157 | 166 | 0.529 | 0.05  | 142 | 0.524 | 0.05  | 140 | 0.502 | 0.025 | 167 | 0.518 | 0.041 |
| 54  | 157 | 0.544 | 0.063 | 190 | 0.502 | 0.034 | 109 | 0.508 | 0.042 | 168 | 0.518 | 0.047 |
| 213 | 162 | 0.533 | 0.059 | 148 | 0.52  | 0.049 | 233 | 0.5   | 0.019 | 169 | 0.518 | 0.043 |
| 199 | 164 | 0.533 | 0.057 | 150 | 0.52  | 0.048 | 249 | 0.499 | 0.024 | 170 | 0.517 | 0.043 |
| 70  | 168 | 0.525 | 0.054 | 155 | 0.515 | 0.05  | 113 | 0.506 | 0.036 | 171 | 0.516 | 0.047 |
| 115 | 190 | 0.506 | 0.048 | 161 | 0.511 | 0.057 | 96  | 0.514 | 0.066 | 172 | 0.51  | 0.057 |
| 36  | 165 | 0.531 | 0.039 | 231 | 0.5   | 0.024 | 223 | 0.5   | 0.03  | 173 | 0.51  | 0.031 |
| 198 | 174 | 0.519 | 0.047 | 166 | 0.51  | 0.044 | 145 | 0.501 | 0.028 | 174 | 0.51  | 0.04  |
| 59  | 181 | 0.509 | 0.05  | 170 | 0.507 | 0.052 | 98  | 0.513 | 0.062 | 175 | 0.51  | 0.055 |
| 202 | 167 | 0.526 | 0.056 | 215 | 0.5   | 0.024 | 139 | 0.502 | 0.022 | 176 | 0.509 | 0.034 |
| 156 | 176 | 0.517 | 0.045 | 165 | 0.51  | 0.046 | 236 | 0.5   | 0.024 | 177 | 0.509 | 0.038 |
| 216 | 169 | 0.525 | 0.053 | 239 | 0.5   | 0.023 | 161 | 0.501 | 0.022 | 178 | 0.508 | 0.033 |
| 108 | 179 | 0.513 | 0.039 | 191 | 0.501 | 0.03  | 105 | 0.511 | 0.036 | 179 | 0.508 | 0.035 |
| 78  | 170 | 0.522 | 0.041 | 188 | 0.502 | 0.025 | 220 | 0.5   | 0.026 | 180 | 0.508 | 0.031 |
| 141 | 175 | 0.519 | 0.046 | 182 | 0.502 | 0.024 | 125 | 0.503 | 0.024 | 181 | 0.508 | 0.031 |
| 113 | 185 | 0.507 | 0.038 | 173 | 0.503 | 0.03  | 102 | 0.512 | 0.059 | 182 | 0.507 | 0.042 |
| 49  | 182 | 0.509 | 0.048 | 164 | 0.511 | 0.057 | 135 | 0.502 | 0.024 | 183 | 0.507 | 0.043 |
| 228 | 171 | 0.521 | 0.052 | 197 | 0.501 | 0.023 | 247 | 0.499 | 0.019 | 184 | 0.507 | 0.031 |
| 172 | 172 | 0.52  | 0.052 | 198 | 0.501 | 0.023 | 240 | 0.499 | 0.019 | 185 | 0.507 | 0.031 |
| 164 | 173 | 0.52  | 0.038 | 204 | 0.5   | 0.024 | 237 | 0.5   | 0.025 | 186 | 0.507 | 0.029 |
| 197 | 177 | 0.515 | 0.044 | 178 | 0.502 | 0.023 | 170 | 0.5   | 0.022 | 187 | 0.506 | 0.03  |
| 43  | 194 | 0.505 | 0.036 | 237 | 0.5   | 0.029 | 100 | 0.513 | 0.059 | 188 | 0.506 | 0.041 |
| 35  | 184 | 0.508 | 0.052 | 169 | 0.508 | 0.058 | 159 | 0.501 | 0.021 | 189 | 0.505 | 0.044 |
| 92  | 178 | 0.515 | 0.037 | 206 | 0.5   | 0.022 | 188 | 0.5   | 0.025 | 190 | 0.505 | 0.028 |
| 142 | 183 | 0.508 | 0.036 | 184 | 0.502 | 0.027 | 128 | 0.502 | 0.024 | 191 | 0.504 | 0.029 |
| 150 | 208 | 0.502 | 0.052 | 171 | 0.504 | 0.032 | 119 | 0.505 | 0.026 | 192 | 0.504 | 0.036 |
| 44  | 186 | 0.507 | 0.034 | 177 | 0.503 | 0.026 | 161 | 0.501 | 0.029 | 193 | 0.503 | 0.029 |
| 212 | 192 | 0.505 | 0.033 | 174 | 0.503 | 0.029 | 246 | 0.499 | 0.019 | 194 | 0.503 | 0.027 |
| 104 | 206 | 0.502 | 0.021 | 250 | 0.498 | 0.028 | 110 | 0.508 | 0.03  | 195 | 0.502 | 0.026 |
| 178 | 250 | 0.499 | 0.023 | 176 | 0.503 | 0.023 | 116 | 0.505 | 0.025 | 196 | 0.502 | 0.023 |
| 218 | 189 | 0.506 | 0.029 | 202 | 0.501 | 0.021 | 239 | 0.5   | 0.018 | 197 | 0.502 | 0.023 |
| 224 | 193 | 0.505 | 0.028 | 181 | 0.502 | 0.02  | 243 | 0.499 | 0.024 | 198 | 0.502 | 0.024 |
| 219 | 196 | 0.504 | 0.021 | 195 | 0.501 | 0.022 | 151 | 0.501 | 0.02  | 199 | 0.502 | 0.021 |
| 168 | 198 | 0.504 | 0.025 | 234 | 0.5   | 0.021 | 130 | 0.502 | 0.026 | 200 | 0.502 | 0.024 |
| 201 | 199 | 0.504 | 0.035 | 196 | 0.501 | 0.032 | 154 | 0.501 | 0.024 | 201 | 0.502 | 0.03  |
| 196 | 201 | 0.503 | 0.024 | 179 | 0.502 | 0.024 | 230 | 0.5   | 0.022 | 202 | 0.502 | 0.024 |
| 153 | 200 | 0.503 | 0.041 | 241 | 0.499 | 0.043 | 126 | 0.503 | 0.025 | 203 | 0.502 | 0.036 |
| 217 | 197 | 0.504 | 0.026 | 245 | 0.499 | 0.022 | 151 | 0.501 | 0.022 | 204 | 0.501 | 0.024 |
| 100 | 188 | 0.507 | 0.033 | 254 | 0.497 | 0.025 | 232 | 0.5   | 0.029 | 205 | 0.501 | 0.029 |
| 195 | 237 | 0.5   | 0.042 | 179 | 0.502 | 0.043 | 143 | 0.501 | 0.027 | 206 | 0.501 | 0.037 |
| 203 | 195 | 0.504 | 0.025 | 230 | 0.5   | 0.022 | 242 | 0.499 | 0.02  | 207 | 0.501 | 0.022 |
| 179 | 247 | 0.499 | 0.022 | 205 | 0.5   | 0.025 | 124 | 0.504 | 0.028 | 208 | 0.501 | 0.025 |
| 140 | 212 | 0.501 | 0.022 | 183 | 0.502 | 0.021 | 223 | 0.5   | 0.021 | 209 | 0.501 | 0.021 |
| 248 | 207 | 0.502 | 0.014 | 217 | 0.5   | 0.01  | 172 | 0.5   | 0.003 | 210 | 0.501 | 0.009 |
| 237 | 232 | 0.5   | 0.012 | 187 | 0.502 | 0.019 | 173 | 0.5   | 0.003 | 211 | 0.501 | 0.012 |
| 223 | 216 | 0.501 | 0.021 | 184 | 0.502 | 0.02  | 238 | 0.5   | 0.015 | 212 | 0.501 | 0.019 |
| 234 | 205 | 0.502 | 0.015 | 232 | 0.5   | 0.012 | 225 | 0.5   | 0.003 | 213 | 0.501 | 0.01  |
| 23  | 248 | 0.499 | 0.036 | 193 | 0.501 | 0.021 | 136 | 0.502 | 0.034 | 213 | 0.501 | 0.03  |
| 77  | 210 | 0.501 | 0.024 | 227 | 0.5   | 0.02  | 164 | 0.501 | 0.02  | 215 | 0.501 | 0.021 |
| 192 | 215 | 0.501 | 0.017 | 203 | 0.501 | 0.022 | 177 | 0.5   | 0.025 | 216 | 0.5   | 0.021 |
| 4   | 211 | 0.501 | 0.023 | 199 | 0.501 | 0.026 | 240 | 0.499 | 0.026 | 217 | 0.5   | 0.025 |
| 76  | 240 | 0.5   | 0.021 | 192 | 0.501 | 0.021 | 168 | 0.5   | 0.022 | 218 | 0.5   | 0.021 |
| 220 | 213 | 0.501 | 0.022 | 211 | 0.5   | 0.02  | 185 | 0.5   | 0.021 | 218 | 0.5   | 0.021 |
| 40  | 224 | 0.5   | 0.006 | 201 | 0.501 | 0.008 | 179 | 0.5   | 0.024 | 220 | 0.5   | 0.013 |
| 68  | 203 | 0.502 | 0.022 | 255 | 0.497 | 0.023 | 148 | 0.501 | 0.021 | 221 | 0.5   | 0.022 |
| 160 | 232 | 0.5   | 0     | 222 | 0.5   | 0     | 167 | 0.5   | 0.009 | 222 | 0.5   | 0.003 |
| 8   | 217 | 0.501 | 0.007 | 213 | 0.5   | 0.009 | 235 | 0.5   | 0.023 | 223 | 0.5   | 0.013 |
| 235 | 220 | 0.5   | 0.003 | 227 | 0.5   | 0.005 | 204 | 0.5   | 0     | 224 | 0.5   | 0.003 |
| 239 | 227 | 0.5   | 0.004 | 210 | 0.5   | 0.004 | 204 | 0.5   | 0     | 224 | 0.5   | 0.003 |
| 51  | 238 | 0.5   | 0.022 | 240 | 0.5   | 0.022 | 154 | 0.501 | 0.021 | 226 | 0.5   | 0.022 |
| 136 | 214 | 0.501 | 0.018 | 212 | 0.5   | 0.022 | 244 | 0.499 | 0.025 | 227 | 0.5   | 0.022 |
| 253 | 224 | 0.5   | 0.005 | 222 | 0.5   | 0.002 | 204 | 0.5   | 0     | 228 | 0.5   | 0.002 |
| 72  | 252 | 0.498 | 0.02  | 175 | 0.503 | 0.024 | 250 | 0.498 | 0.025 | 229 | 0.5   | 0.023 |
| 250 | 232 | 0.5   | 0     | 222 | 0.5   | 0     | 204 | 0.5   | 0     | 232 | 0.5   | 0     |
| 251 | 232 | 0.5   | 0     | 222 | 0.5   | 0     | 204 | 0.5   | 0     | 232 | 0.5   | 0     |
| 254 | 232 | 0.5   | 0     | 222 | 0.5   | 0     | 204 | 0.5   | 0     | 232 | 0.5   | 0     |
| 0   | 232 | 0.5   | 0     | 222 | 0.5   | 0     | 204 | 0.5   | 0     | 232 | 0.5   | 0     |
| 255 | 232 | 0.5   | 0     | 222 | 0.5   | 0     | 204 | 0.5   | 0     | 232 | 0.5   | 0     |
| 32  | 232 | 0.5   | 0     | 222 | 0.5   | 0     | 204 | 0.5   | 0     | 232 | 0.5   | 0     |
| 128 | 226 | 0.5   | 0     | 222 | 0.5   | 0     | 227 | 0.5   | 0.002 | 236 | 0.5   | 0.001 |
| 252 | 222 | 0.5   | 0.012 | 235 | 0.5   | 0.01  | 185 | 0.5   | 0.001 | 236 | 0.5   | 0.008 |
| 96  | 219 | 0.5   | 0.006 | 216 | 0.5   | 0.006 | 245 | 0.499 | 0.017 | 238 | 0.5   | 0.01  |
| 249 | 243 | 0.5   | 0.003 | 233 | 0.5   | 0.005 | 204 | 0.5   | 0     | 239 | 0.5   | 0.003 |
| 232 | 246 | 0.499 | 0.024 | 213 | 0.5   | 0.024 | 204 | 0.5   | 0.021 | 239 | 0.5   | 0.023 |
| 207 | 239 | 0.5   | 0.021 | 252 | 0.497 | 0.022 | 142 | 0.502 | 0.024 | 241 | 0.5   | 0.022 |
| 236 | 249 | 0.499 | 0.02  | 236 | 0.5   | 0.02  | 170 | 0.5   | 0.02  | 242 | 0.5   | 0.02  |
| 205 | 253 | 0.498 | 0.021 | 200 | 0.501 | 0.024 | 204 | 0.5   | 0.021 | 243 | 0.5   | 0.022 |
| 12  | 218 | 0.501 | 0.023 | 248 | 0.498 | 0.022 | 230 | 0.5   | 0.024 | 244 | 0.5   | 0.023 |
| 206 | 209 | 0.502 | 0.022 | 243 | 0.499 | 0.022 | 252 | 0.498 | 0.022 | 244 | 0.5   | 0.022 |
| 55  | 255 | 0.496 | 0.031 | 207 | 0.5   | 0.021 | 140 | 0.502 | 0.036 | 246 | 0.499 | 0.029 |
| 238 | 232 | 0.5   | 0.012 | 246 | 0.499 | 0.011 | 234 | 0.5   | 0.004 | 247 | 0.499 | 0.009 |
| 64  | 241 | 0.5   | 0.009 | 229 | 0.5   | 0.008 | 248 | 0.499 | 0.018 | 248 | 0.499 | 0.012 |
| 204 | 221 | 0.5   | 0.021 | 209 | 0.5   | 0.022 | 255 | 0.497 | 0.02  | 249 | 0.499 | 0.021 |

## A. Appendix 1

|            |     |       |       |     |       |       |     |       |       |     |       |       |
|------------|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|-------|-------|
| <b>233</b> | 223 | 0.5   | 0.012 | 251 | 0.497 | 0.021 | 204 | 0.5   | 0.002 | 250 | 0.499 | 0.012 |
| <b>50</b>  | 241 | 0.5   | 0.021 | 249 | 0.498 | 0.024 | 221 | 0.5   | 0.023 | 251 | 0.499 | 0.023 |
| <b>222</b> | 245 | 0.499 | 0.025 | 256 | 0.497 | 0.022 | 146 | 0.501 | 0.019 | 252 | 0.499 | 0.022 |
| <b>200</b> | 254 | 0.498 | 0.025 | 247 | 0.498 | 0.024 | 165 | 0.501 | 0.022 | 253 | 0.499 | 0.024 |
| <b>19</b>  | 256 | 0.496 | 0.031 | 237 | 0.5   | 0.021 | 158 | 0.501 | 0.021 | 254 | 0.499 | 0.025 |
| <b>221</b> | 244 | 0.5   | 0.022 | 242 | 0.499 | 0.022 | 253 | 0.497 | 0.021 | 255 | 0.499 | 0.022 |
| <b>132</b> | 251 | 0.499 | 0.022 | 253 | 0.497 | 0.023 | 254 | 0.497 | 0.026 | 256 | 0.498 | 0.024 |

Table A.2: Full results for  $\mathcal{N}_{BA}$ ,  $\mathcal{N}_{ER}$ , and  $\mathcal{N}_{WS}$  from experiments, including the rank, mean accuracy, and standard deviation for each rule and network pair. Additionally, average results over all three networks is also included. The rules are ordered based on the overall results.

| $R_i$      | Rank | $\mu(\mathcal{N}_{BA})$ | $\sigma^2(\mathcal{N}_{BA})$ | Rank | $\mu(\mathcal{N}_{ER})$ | $\sigma^2(\mathcal{N}_{ER})$ | Rank | $\mu(\mathcal{N}_{WS})$ | $\sigma^2(\mathcal{N}_{WS})$ | Rank | $\mu(\cdot)$ | $\sigma^2(\cdot)$ |
|------------|------|-------------------------|------------------------------|------|-------------------------|------------------------------|------|-------------------------|------------------------------|------|--------------|-------------------|
| <b>9</b>   | 1    | 0.683                   | 0.11                         | 1    | 0.675                   | 0.11                         | 11   | 0.674                   | 0.097                        | 1    | 0.677        | 0.106             |
| <b>91</b>  | 2    | 0.677                   | 0.127                        | 6    | 0.659                   | 0.124                        | 1    | 0.693                   | 0.123                        | 2    | 0.676        | 0.125             |
| <b>65</b>  | 3    | 0.674                   | 0.107                        | 2    | 0.661                   | 0.106                        | 8    | 0.676                   | 0.102                        | 3    | 0.671        | 0.105             |
| <b>37</b>  | 6    | 0.662                   | 0.125                        | 3    | 0.661                   | 0.112                        | 3    | 0.683                   | 0.113                        | 4    | 0.669        | 0.117             |
| <b>111</b> | 4    | 0.671                   | 0.131                        | 5    | 0.66                    | 0.114                        | 9    | 0.675                   | 0.109                        | 5    | 0.668        | 0.118             |
| <b>125</b> | 5    | 0.666                   | 0.115                        | 4    | 0.66                    | 0.125                        | 12   | 0.673                   | 0.11                         | 6    | 0.666        | 0.116             |
| <b>107</b> | 7    | 0.662                   | 0.104                        | 7    | 0.654                   | 0.105                        | 20   | 0.655                   | 0.104                        | 7    | 0.657        | 0.105             |
| <b>30</b>  | 8    | 0.661                   | 0.096                        | 9    | 0.652                   | 0.099                        | 18   | 0.657                   | 0.11                         | 8    | 0.657        | 0.102             |
| <b>126</b> | 10   | 0.655                   | 0.109                        | 10   | 0.65                    | 0.096                        | 15   | 0.664                   | 0.107                        | 9    | 0.656        | 0.104             |
| <b>90</b>  | 17   | 0.643                   | 0.113                        | 8    | 0.654                   | 0.106                        | 13   | 0.668                   | 0.118                        | 10   | 0.655        | 0.112             |
| <b>71</b>  | 15   | 0.645                   | 0.123                        | 20   | 0.637                   | 0.126                        | 5    | 0.682                   | 0.128                        | 11   | 0.654        | 0.126             |
| <b>29</b>  | 19   | 0.642                   | 0.121                        | 18   | 0.638                   | 0.121                        | 7    | 0.677                   | 0.126                        | 12   | 0.652        | 0.123             |
| <b>121</b> | 11   | 0.655                   | 0.104                        | 14   | 0.644                   | 0.106                        | 22   | 0.653                   | 0.1                          | 13   | 0.651        | 0.103             |
| <b>122</b> | 9    | 0.655                   | 0.104                        | 12   | 0.644                   | 0.098                        | 23   | 0.65                    | 0.106                        | 14   | 0.65         | 0.102             |
| <b>86</b>  | 12   | 0.654                   | 0.095                        | 17   | 0.639                   | 0.103                        | 19   | 0.656                   | 0.108                        | 15   | 0.65         | 0.102             |
| <b>133</b> | 48   | 0.622                   | 0.114                        | 19   | 0.637                   | 0.125                        | 2    | 0.686                   | 0.117                        | 16   | 0.648        | 0.119             |
| <b>116</b> | 13   | 0.648                   | 0.108                        | 25   | 0.632                   | 0.113                        | 17   | 0.659                   | 0.112                        | 17   | 0.646        | 0.111             |
| <b>129</b> | 14   | 0.646                   | 0.095                        | 13   | 0.644                   | 0.088                        | 26   | 0.648                   | 0.095                        | 18   | 0.646        | 0.093             |
| <b>94</b>  | 36   | 0.628                   | 0.125                        | 31   | 0.627                   | 0.122                        | 4    | 0.683                   | 0.135                        | 19   | 0.646        | 0.127             |
| <b>46</b>  | 23   | 0.637                   | 0.111                        | 30   | 0.627                   | 0.107                        | 14   | 0.666                   | 0.123                        | 20   | 0.643        | 0.114             |
| <b>62</b>  | 20   | 0.641                   | 0.115                        | 11   | 0.65                    | 0.115                        | 37   | 0.639                   | 0.129                        | 21   | 0.643        | 0.12              |
| <b>56</b>  | 18   | 0.643                   | 0.144                        | 33   | 0.626                   | 0.141                        | 27   | 0.648                   | 0.124                        | 22   | 0.639        | 0.136             |
| <b>181</b> | 26   | 0.634                   | 0.101                        | 21   | 0.635                   | 0.106                        | 28   | 0.646                   | 0.097                        | 23   | 0.638        | 0.101             |
| <b>118</b> | 24   | 0.637                   | 0.113                        | 24   | 0.632                   | 0.116                        | 29   | 0.645                   | 0.127                        | 24   | 0.638        | 0.118             |
| <b>131</b> | 16   | 0.644                   | 0.1                          | 22   | 0.635                   | 0.106                        | 49   | 0.632                   | 0.107                        | 25   | 0.637        | 0.104             |
| <b>167</b> | 32   | 0.631                   | 0.103                        | 15   | 0.642                   | 0.107                        | 38   | 0.637                   | 0.095                        | 26   | 0.637        | 0.102             |
| <b>123</b> | 31   | 0.631                   | 0.119                        | 43   | 0.617                   | 0.121                        | 16   | 0.66                    | 0.132                        | 27   | 0.636        | 0.124             |
| <b>41</b>  | 21   | 0.64                    | 0.092                        | 16   | 0.642                   | 0.092                        | 64   | 0.623                   | 0.085                        | 28   | 0.635        | 0.09              |
| <b>145</b> | 22   | 0.638                   | 0.097                        | 27   | 0.631                   | 0.1                          | 45   | 0.634                   | 0.1                          | 29   | 0.634        | 0.099             |
| <b>47</b>  | 33   | 0.629                   | 0.141                        | 23   | 0.634                   | 0.147                        | 43   | 0.635                   | 0.152                        | 30   | 0.633        | 0.147             |
| <b>97</b>  | 25   | 0.635                   | 0.092                        | 26   | 0.631                   | 0.09                         | 48   | 0.632                   | 0.088                        | 31   | 0.633        | 0.09              |
| <b>98</b>  | 30   | 0.631                   | 0.145                        | 42   | 0.618                   | 0.135                        | 25   | 0.648                   | 0.124                        | 32   | 0.633        | 0.135             |
| <b>176</b> | 28   | 0.634                   | 0.147                        | 35   | 0.625                   | 0.144                        | 39   | 0.637                   | 0.151                        | 33   | 0.632        | 0.148             |
| <b>48</b>  | 27   | 0.634                   | 0.152                        | 29   | 0.628                   | 0.146                        | 46   | 0.632                   | 0.147                        | 34   | 0.631        | 0.149             |
| <b>161</b> | 29   | 0.633                   | 0.095                        | 37   | 0.624                   | 0.09                         | 44   | 0.634                   | 0.087                        | 35   | 0.63         | 0.091             |
| <b>135</b> | 34   | 0.629                   | 0.082                        | 40   | 0.62                    | 0.081                        | 32   | 0.643                   | 0.095                        | 36   | 0.63         | 0.086             |
| <b>165</b> | 40   | 0.626                   | 0.094                        | 28   | 0.63                    | 0.099                        | 56   | 0.628                   | 0.099                        | 37   | 0.628        | 0.097             |
| <b>82</b>  | 42   | 0.626                   | 0.102                        | 39   | 0.621                   | 0.105                        | 47   | 0.632                   | 0.11                         | 38   | 0.626        | 0.106             |
| <b>149</b> | 45   | 0.623                   | 0.087                        | 41   | 0.619                   | 0.086                        | 42   | 0.635                   | 0.099                        | 39   | 0.626        | 0.091             |
| <b>117</b> | 51   | 0.618                   | 0.141                        | 57   | 0.61                    | 0.144                        | 24   | 0.65                    | 0.151                        | 40   | 0.626        | 0.145             |
| <b>162</b> | 58   | 0.613                   | 0.141                        | 55   | 0.61                    | 0.14                         | 21   | 0.653                   | 0.148                        | 41   | 0.625        | 0.143             |
| <b>61</b>  | 38   | 0.627                   | 0.108                        | 34   | 0.625                   | 0.111                        | 61   | 0.624                   | 0.118                        | 42   | 0.625        | 0.113             |
| <b>5</b>   | 66   | 0.607                   | 0.122                        | 75   | 0.589                   | 0.117                        | 6    | 0.678                   | 0.142                        | 43   | 0.625        | 0.127             |
| <b>26</b>  | 44   | 0.625                   | 0.103                        | 50   | 0.613                   | 0.114                        | 40   | 0.636                   | 0.105                        | 44   | 0.625        | 0.107             |
| <b>34</b>  | 52   | 0.617                   | 0.145                        | 52   | 0.612                   | 0.14                         | 30   | 0.644                   | 0.154                        | 45   | 0.624        | 0.146             |
| <b>67</b>  | 41   | 0.626                   | 0.1                          | 36   | 0.624                   | 0.104                        | 68   | 0.621                   | 0.102                        | 46   | 0.624        | 0.102             |
| <b>144</b> | 37   | 0.628                   | 0.16                         | 46   | 0.616                   | 0.157                        | 59   | 0.627                   | 0.152                        | 47   | 0.623        | 0.157             |
| <b>103</b> | 35   | 0.628                   | 0.11                         | 51   | 0.612                   | 0.11                         | 55   | 0.628                   | 0.113                        | 48   | 0.623        | 0.111             |
| <b>31</b>  | 50   | 0.619                   | 0.147                        | 32   | 0.627                   | 0.153                        | 66   | 0.622                   | 0.153                        | 49   | 0.623        | 0.151             |
| <b>95</b>  | 74   | 0.597                   | 0.118                        | 71   | 0.593                   | 0.114                        | 10   | 0.674                   | 0.146                        | 50   | 0.621        | 0.126             |
| <b>63</b>  | 39   | 0.626                   | 0.156                        | 38   | 0.623                   | 0.154                        | 77   | 0.615                   | 0.15                         | 51   | 0.621        | 0.153             |
| <b>11</b>  | 49   | 0.619                   | 0.139                        | 45   | 0.616                   | 0.147                        | 60   | 0.627                   | 0.147                        | 52   | 0.62         | 0.144             |
| <b>81</b>  | 57   | 0.613                   | 0.145                        | 58   | 0.607                   | 0.141                        | 33   | 0.641                   | 0.148                        | 53   | 0.62         | 0.145             |
| <b>16</b>  | 43   | 0.625                   | 0.161                        | 54   | 0.611                   | 0.153                        | 66   | 0.622                   | 0.155                        | 54   | 0.619        | 0.156             |
| <b>45</b>  | 55   | 0.614                   | 0.099                        | 48   | 0.615                   | 0.097                        | 58   | 0.627                   | 0.106                        | 55   | 0.619        | 0.101             |
| <b>24</b>  | 46   | 0.623                   | 0.16                         | 56   | 0.61                    | 0.152                        | 65   | 0.623                   | 0.15                         | 56   | 0.619        | 0.154             |
| <b>101</b> | 59   | 0.61                    | 0.102                        | 44   | 0.616                   | 0.103                        | 54   | 0.629                   | 0.102                        | 57   | 0.618        | 0.102             |
| <b>130</b> | 56   | 0.613                   | 0.156                        | 66   | 0.601                   | 0.152                        | 34   | 0.64                    | 0.168                        | 58   | 0.618        | 0.159             |
| <b>2</b>   | 62   | 0.61                    | 0.153                        | 61   | 0.605                   | 0.152                        | 35   | 0.639                   | 0.169                        | 59   | 0.618        | 0.158             |
| <b>25</b>  | 47   | 0.623                   | 0.1                          | 53   | 0.611                   | 0.101                        | 71   | 0.618                   | 0.094                        | 60   | 0.617        | 0.099             |
| <b>33</b>  | 60   | 0.61                    | 0.113                        | 63   | 0.605                   | 0.112                        | 41   | 0.636                   | 0.118                        | 61   | 0.617        | 0.114             |
| <b>180</b> | 53   | 0.616                   | 0.101                        | 65   | 0.602                   | 0.097                        | 50   | 0.631                   | 0.098                        | 62   | 0.616        | 0.099             |
| <b>66</b>  | 65   | 0.607                   | 0.154                        | 67   | 0.601                   | 0.143                        | 52   | 0.63                    | 0.156                        | 63   | 0.613        | 0.151             |
| <b>119</b> | 68   | 0.606                   | 0.153                        | 60   | 0.607                   | 0.152                        | 61   | 0.624                   | 0.152                        | 64   | 0.612        | 0.152             |
| <b>106</b> | 69   | 0.605                   | 0.099                        | 47   | 0.616                   | 0.106                        | 75   | 0.615                   | 0.102                        | 65   | 0.612        | 0.102             |
| <b>3</b>   | 61   | 0.61                    | 0.154                        | 49   | 0.615                   | 0.155                        | 82   | 0.609                   | 0.152                        | 66   | 0.611        | 0.154             |
| <b>87</b>  | 63   | 0.608                   | 0.15                         | 69   | 0.594                   | 0.142                        | 53   | 0.629                   | 0.152                        | 67   | 0.611        | 0.148             |
| <b>120</b> | 54   | 0.616                   | 0.101                        | 59   | 0.607                   | 0.101                        | 81   | 0.609                   | 0.102                        | 68   | 0.61         | 0.101             |
| <b>7</b>   | 67   | 0.606                   | 0.144                        | 62   | 0.605                   | 0.141                        | 73   | 0.617                   | 0.148                        | 69   | 0.609        | 0.144             |
| <b>166</b> | 78   | 0.592                   | 0.087                        | 64   | 0.604                   | 0.094                        | 63   | 0.623                   | 0.1                          | 70   | 0.606        | 0.094             |
| <b>127</b> | 82   | 0.587                   | 0.11                         | 81   | 0.582                   | 0.119                        | 31   | 0.643                   | 0.141                        | 71   | 0.604        | 0.123             |
| <b>17</b>  | 70   | 0.605                   | 0.153                        | 76   | 0.588                   | 0.143                        | 72   | 0.617                   | 0.15                         | 72   | 0.603        | 0.149             |

|     |     |       |       |     |       |       |     |       |       |     |       |       |
|-----|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|-------|-------|
| 112 | 76  | 0.596 | 0.124 | 68  | 0.6   | 0.136 | 78  | 0.613 | 0.143 | 73  | 0.603 | 0.134 |
| 21  | 77  | 0.593 | 0.143 | 76  | 0.588 | 0.14  | 57  | 0.628 | 0.148 | 74  | 0.603 | 0.144 |
| 124 | 64  | 0.608 | 0.091 | 70  | 0.593 | 0.093 | 84  | 0.603 | 0.1   | 75  | 0.601 | 0.095 |
| 183 | 71  | 0.598 | 0.098 | 72  | 0.591 | 0.092 | 80  | 0.609 | 0.104 | 76  | 0.6   | 0.098 |
| 42  | 85  | 0.583 | 0.119 | 80  | 0.584 | 0.126 | 51  | 0.63  | 0.149 | 77  | 0.599 | 0.132 |
| 15  | 75  | 0.596 | 0.136 | 79  | 0.587 | 0.137 | 79  | 0.613 | 0.145 | 78  | 0.599 | 0.139 |
| 52  | 79  | 0.591 | 0.094 | 78  | 0.588 | 0.1   | 74  | 0.617 | 0.098 | 79  | 0.599 | 0.097 |
| 182 | 72  | 0.598 | 0.096 | 74  | 0.59  | 0.093 | 85  | 0.602 | 0.094 | 80  | 0.597 | 0.094 |
| 110 | 73  | 0.598 | 0.098 | 73  | 0.59  | 0.092 | 86  | 0.601 | 0.104 | 81  | 0.597 | 0.098 |
| 1   | 90  | 0.578 | 0.111 | 85  | 0.572 | 0.107 | 36  | 0.639 | 0.144 | 82  | 0.596 | 0.121 |
| 38  | 84  | 0.585 | 0.105 | 82  | 0.58  | 0.095 | 69  | 0.619 | 0.1   | 83  | 0.595 | 0.1   |
| 85  | 80  | 0.591 | 0.137 | 84  | 0.574 | 0.127 | 70  | 0.618 | 0.144 | 84  | 0.594 | 0.136 |
| 209 | 87  | 0.582 | 0.095 | 91  | 0.568 | 0.093 | 83  | 0.606 | 0.118 | 85  | 0.585 | 0.102 |
| 152 | 83  | 0.587 | 0.128 | 86  | 0.571 | 0.12  | 92  | 0.591 | 0.122 | 86  | 0.583 | 0.123 |
| 194 | 88  | 0.58  | 0.124 | 87  | 0.57  | 0.122 | 89  | 0.594 | 0.125 | 87  | 0.582 | 0.123 |
| 151 | 91  | 0.576 | 0.075 | 83  | 0.58  | 0.068 | 95  | 0.585 | 0.075 | 88  | 0.58  | 0.073 |
| 139 | 97  | 0.565 | 0.089 | 98  | 0.559 | 0.093 | 76  | 0.615 | 0.115 | 89  | 0.58  | 0.099 |
| 75  | 81  | 0.59  | 0.108 | 89  | 0.569 | 0.096 | 102 | 0.578 | 0.102 | 90  | 0.579 | 0.102 |
| 89  | 94  | 0.571 | 0.096 | 90  | 0.568 | 0.103 | 90  | 0.594 | 0.109 | 91  | 0.578 | 0.103 |
| 22  | 93  | 0.573 | 0.08  | 88  | 0.569 | 0.075 | 93  | 0.587 | 0.079 | 92  | 0.576 | 0.078 |
| 243 | 86  | 0.583 | 0.135 | 94  | 0.561 | 0.122 | 104 | 0.577 | 0.139 | 93  | 0.574 | 0.132 |
| 242 | 89  | 0.579 | 0.131 | 96  | 0.561 | 0.12  | 101 | 0.578 | 0.136 | 94  | 0.573 | 0.129 |
| 80  | 100 | 0.563 | 0.126 | 97  | 0.559 | 0.126 | 88  | 0.595 | 0.15  | 95  | 0.573 | 0.134 |
| 227 | 92  | 0.575 | 0.13  | 92  | 0.564 | 0.128 | 109 | 0.574 | 0.123 | 96  | 0.571 | 0.127 |
| 10  | 111 | 0.553 | 0.119 | 103 | 0.554 | 0.119 | 87  | 0.6   | 0.159 | 97  | 0.569 | 0.132 |
| 105 | 102 | 0.561 | 0.072 | 95  | 0.561 | 0.079 | 98  | 0.581 | 0.086 | 98  | 0.568 | 0.079 |
| 187 | 98  | 0.564 | 0.125 | 101 | 0.556 | 0.122 | 99  | 0.58  | 0.141 | 99  | 0.567 | 0.129 |
| 186 | 105 | 0.559 | 0.114 | 99  | 0.559 | 0.119 | 97  | 0.582 | 0.138 | 100 | 0.566 | 0.124 |
| 184 | 114 | 0.552 | 0.089 | 104 | 0.553 | 0.091 | 91  | 0.591 | 0.118 | 101 | 0.565 | 0.099 |
| 185 | 103 | 0.56  | 0.118 | 111 | 0.551 | 0.11  | 100 | 0.579 | 0.127 | 102 | 0.563 | 0.118 |
| 225 | 108 | 0.555 | 0.087 | 100 | 0.558 | 0.092 | 110 | 0.574 | 0.098 | 103 | 0.562 | 0.092 |
| 226 | 96  | 0.565 | 0.101 | 105 | 0.552 | 0.089 | 113 | 0.57  | 0.108 | 104 | 0.562 | 0.099 |
| 137 | 104 | 0.559 | 0.077 | 108 | 0.551 | 0.081 | 111 | 0.573 | 0.084 | 105 | 0.561 | 0.081 |
| 247 | 95  | 0.568 | 0.128 | 106 | 0.552 | 0.118 | 120 | 0.563 | 0.126 | 106 | 0.561 | 0.124 |
| 193 | 107 | 0.556 | 0.077 | 102 | 0.556 | 0.077 | 116 | 0.568 | 0.087 | 107 | 0.56  | 0.08  |
| 208 | 109 | 0.554 | 0.113 | 114 | 0.549 | 0.109 | 105 | 0.577 | 0.134 | 108 | 0.56  | 0.119 |
| 169 | 124 | 0.54  | 0.084 | 112 | 0.551 | 0.088 | 94  | 0.586 | 0.109 | 109 | 0.559 | 0.093 |
| 190 | 112 | 0.553 | 0.118 | 115 | 0.548 | 0.116 | 107 | 0.575 | 0.14  | 110 | 0.559 | 0.125 |
| 150 | 113 | 0.553 | 0.07  | 93  | 0.562 | 0.068 | 126 | 0.56  | 0.069 | 111 | 0.558 | 0.069 |
| 231 | 99  | 0.564 | 0.123 | 109 | 0.551 | 0.114 | 127 | 0.56  | 0.125 | 112 | 0.558 | 0.121 |
| 246 | 101 | 0.562 | 0.122 | 110 | 0.551 | 0.118 | 125 | 0.561 | 0.131 | 113 | 0.558 | 0.124 |
| 191 | 110 | 0.553 | 0.119 | 113 | 0.55  | 0.118 | 112 | 0.571 | 0.138 | 114 | 0.558 | 0.125 |
| 189 | 115 | 0.552 | 0.115 | 116 | 0.548 | 0.113 | 115 | 0.569 | 0.13  | 115 | 0.556 | 0.119 |
| 138 | 119 | 0.546 | 0.104 | 124 | 0.538 | 0.102 | 96  | 0.583 | 0.137 | 116 | 0.555 | 0.114 |
| 241 | 117 | 0.548 | 0.102 | 122 | 0.539 | 0.098 | 108 | 0.575 | 0.139 | 117 | 0.554 | 0.113 |
| 210 | 118 | 0.547 | 0.085 | 123 | 0.538 | 0.081 | 106 | 0.575 | 0.099 | 118 | 0.553 | 0.088 |
| 109 | 121 | 0.542 | 0.067 | 107 | 0.551 | 0.072 | 118 | 0.566 | 0.086 | 119 | 0.553 | 0.075 |
| 230 | 106 | 0.557 | 0.117 | 120 | 0.54  | 0.101 | 133 | 0.55  | 0.111 | 120 | 0.549 | 0.109 |
| 146 | 116 | 0.55  | 0.068 | 118 | 0.543 | 0.066 | 132 | 0.553 | 0.071 | 121 | 0.549 | 0.068 |
| 154 | 135 | 0.529 | 0.073 | 125 | 0.537 | 0.079 | 103 | 0.577 | 0.095 | 122 | 0.548 | 0.082 |
| 171 | 123 | 0.54  | 0.097 | 121 | 0.54  | 0.099 | 124 | 0.562 | 0.125 | 123 | 0.548 | 0.107 |
| 188 | 122 | 0.541 | 0.108 | 119 | 0.542 | 0.104 | 129 | 0.555 | 0.116 | 124 | 0.546 | 0.109 |
| 211 | 130 | 0.534 | 0.075 | 126 | 0.533 | 0.077 | 117 | 0.568 | 0.1   | 125 | 0.545 | 0.084 |
| 240 | 120 | 0.544 | 0.094 | 134 | 0.528 | 0.079 | 122 | 0.562 | 0.119 | 126 | 0.545 | 0.097 |
| 245 | 127 | 0.537 | 0.099 | 136 | 0.528 | 0.082 | 123 | 0.562 | 0.132 | 127 | 0.542 | 0.104 |
| 155 | 138 | 0.526 | 0.067 | 128 | 0.531 | 0.08  | 114 | 0.57  | 0.099 | 128 | 0.542 | 0.082 |
| 244 | 125 | 0.539 | 0.1   | 137 | 0.527 | 0.084 | 131 | 0.554 | 0.118 | 129 | 0.54  | 0.101 |
| 170 | 134 | 0.529 | 0.075 | 127 | 0.532 | 0.083 | 128 | 0.558 | 0.114 | 130 | 0.54  | 0.09  |
| 175 | 129 | 0.536 | 0.102 | 141 | 0.522 | 0.075 | 130 | 0.555 | 0.124 | 131 | 0.538 | 0.1   |
| 57  | 132 | 0.532 | 0.066 | 131 | 0.529 | 0.07  | 136 | 0.549 | 0.074 | 132 | 0.537 | 0.07  |
| 60  | 128 | 0.537 | 0.066 | 117 | 0.544 | 0.073 | 163 | 0.526 | 0.07  | 133 | 0.536 | 0.07  |
| 6   | 136 | 0.529 | 0.069 | 135 | 0.528 | 0.068 | 135 | 0.549 | 0.085 | 134 | 0.535 | 0.074 |
| 20  | 131 | 0.532 | 0.064 | 129 | 0.531 | 0.073 | 142 | 0.543 | 0.077 | 135 | 0.535 | 0.072 |
| 102 | 126 | 0.539 | 0.07  | 130 | 0.53  | 0.063 | 152 | 0.535 | 0.072 | 136 | 0.534 | 0.068 |
| 18  | 133 | 0.531 | 0.062 | 132 | 0.529 | 0.061 | 146 | 0.54  | 0.075 | 137 | 0.533 | 0.066 |
| 99  | 137 | 0.526 | 0.068 | 143 | 0.522 | 0.066 | 134 | 0.55  | 0.072 | 138 | 0.533 | 0.069 |
| 134 | 140 | 0.525 | 0.06  | 133 | 0.529 | 0.071 | 141 | 0.543 | 0.081 | 139 | 0.532 | 0.071 |
| 177 | 163 | 0.512 | 0.056 | 148 | 0.521 | 0.075 | 119 | 0.564 | 0.103 | 140 | 0.532 | 0.078 |
| 174 | 139 | 0.526 | 0.083 | 140 | 0.524 | 0.072 | 139 | 0.545 | 0.111 | 141 | 0.532 | 0.089 |
| 163 | 169 | 0.511 | 0.057 | 150 | 0.519 | 0.071 | 121 | 0.563 | 0.098 | 142 | 0.531 | 0.076 |
| 148 | 141 | 0.524 | 0.066 | 139 | 0.525 | 0.064 | 143 | 0.543 | 0.074 | 143 | 0.53  | 0.068 |
| 114 | 144 | 0.52  | 0.071 | 149 | 0.52  | 0.075 | 144 | 0.543 | 0.092 | 144 | 0.528 | 0.08  |
| 83  | 156 | 0.515 | 0.062 | 141 | 0.522 | 0.078 | 145 | 0.541 | 0.096 | 145 | 0.526 | 0.079 |
| 88  | 142 | 0.522 | 0.062 | 145 | 0.522 | 0.056 | 151 | 0.535 | 0.071 | 146 | 0.526 | 0.063 |
| 69  | 159 | 0.514 | 0.042 | 158 | 0.516 | 0.05  | 137 | 0.548 | 0.093 | 147 | 0.526 | 0.062 |
| 79  | 145 | 0.52  | 0.056 | 153 | 0.519 | 0.059 | 148 | 0.539 | 0.084 | 148 | 0.526 | 0.066 |
| 93  | 162 | 0.512 | 0.041 | 151 | 0.519 | 0.054 | 138 | 0.545 | 0.091 | 149 | 0.526 | 0.062 |
| 13  | 148 | 0.519 | 0.051 | 152 | 0.519 | 0.051 | 149 | 0.538 | 0.088 | 150 | 0.525 | 0.063 |
| 229 | 143 | 0.522 | 0.065 | 146 | 0.521 | 0.065 | 157 | 0.53  | 0.085 | 151 | 0.524 | 0.072 |
| 74  | 147 | 0.519 | 0.053 | 154 | 0.519 | 0.058 | 150 | 0.535 | 0.075 | 152 | 0.524 | 0.062 |
| 153 | 153 | 0.517 | 0.048 | 147 | 0.521 | 0.058 | 154 | 0.532 | 0.07  | 153 | 0.523 | 0.058 |
| 58  | 165 | 0.512 | 0.062 | 165 | 0.514 | 0.068 | 140 | 0.543 | 0.086 | 154 | 0.523 | 0.072 |
| 195 | 150 | 0.518 | 0.053 | 138 | 0.526 | 0.058 | 165 | 0.524 | 0.061 | 155 | 0.523 | 0.057 |
| 27  | 151 | 0.518 | 0.064 | 157 | 0.517 | 0.062 | 155 | 0.532 | 0.079 | 156 | 0.522 | 0.068 |
| 147 | 154 | 0.516 | 0.054 | 144 | 0.522 | 0.062 | 161 | 0.528 | 0.058 | 157 | 0.522 | 0.058 |
| 159 | 152 | 0.517 | 0.066 | 166 | 0.513 | 0.051 | 153 | 0.534 | 0.075 | 158 | 0.521 | 0.064 |
| 173 | 157 | 0.515 | 0.055 | 156 | 0.518 | 0.058 | 156 | 0.531 | 0.087 | 159 | 0.521 | 0.067 |
| 53  | 172 | 0.509 | 0.06  | 161 | 0.515 | 0.065 | 147 | 0.539 | 0.094 | 160 | 0.521 | 0.073 |
| 215 | 149 | 0.518 | 0.054 | 159 | 0.515 | 0.059 | 159 | 0.529 | 0.071 | 161 | 0.521 | 0.062 |
| 54  | 155 | 0.516 | 0.054 | 155 | 0.518 | 0.06  | 160 | 0.529 | 0.056 | 162 | 0.521 | 0.056 |
| 73  | 158 | 0.515 | 0.047 | 167 | 0.513 | 0.046 | 158 | 0.529 | 0.057 | 163 | 0.519 | 0.05  |
| 14  | 146 | 0.52  | 0.062 | 170 | 0.512 | 0.043 | 168 | 0.523 | 0.06  | 164 | 0.518 | 0.055 |
| 158 | 168 | 0.511 | 0.056 | 160 | 0.515 | 0.059 | 164 | 0.526 | 0.07  | 165 | 0.517 | 0.062 |
| 214 | 161 | 0.513 | 0.056 | 163 | 0.514 | 0.06  | 166 | 0.524 | 0.069 | 166 | 0.517 | 0.061 |

# A. Appendix 1

|            |     |       |       |            |       |       |            |       |       |            |       |       |
|------------|-----|-------|-------|------------|-------|-------|------------|-------|-------|------------|-------|-------|
| <b>39</b>  | 176 | 0.507 | 0.053 | <b>162</b> | 0.514 | 0.071 | <b>162</b> | 0.527 | 0.077 | <b>167</b> | 0.516 | 0.067 |
| <b>84</b>  | 160 | 0.514 | 0.054 | <b>170</b> | 0.512 | 0.045 | <b>169</b> | 0.522 | 0.056 | <b>168</b> | 0.516 | 0.052 |
| <b>28</b>  | 166 | 0.512 | 0.041 | <b>172</b> | 0.511 | 0.05  | <b>167</b> | 0.524 | 0.057 | <b>169</b> | 0.516 | 0.05  |
| <b>115</b> | 164 | 0.512 | 0.052 | <b>164</b> | 0.514 | 0.053 | <b>177</b> | 0.511 | 0.06  | <b>170</b> | 0.512 | 0.055 |
| <b>70</b>  | 181 | 0.505 | 0.041 | <b>173</b> | 0.509 | 0.046 | <b>170</b> | 0.521 | 0.056 | <b>171</b> | 0.512 | 0.047 |
| <b>143</b> | 167 | 0.511 | 0.056 | <b>196</b> | 0.505 | 0.033 | <b>172</b> | 0.518 | 0.058 | <b>172</b> | 0.512 | 0.049 |
| <b>199</b> | 173 | 0.508 | 0.035 | <b>185</b> | 0.506 | 0.037 | <b>171</b> | 0.519 | 0.054 | <b>173</b> | 0.511 | 0.042 |
| <b>59</b>  | 170 | 0.511 | 0.053 | <b>168</b> | 0.512 | 0.052 | <b>179</b> | 0.509 | 0.045 | <b>174</b> | 0.511 | 0.05  |
| <b>213</b> | 171 | 0.51  | 0.056 | <b>186</b> | 0.506 | 0.033 | <b>173</b> | 0.515 | 0.057 | <b>175</b> | 0.51  | 0.048 |
| <b>157</b> | 188 | 0.504 | 0.03  | <b>175</b> | 0.509 | 0.043 | <b>176</b> | 0.511 | 0.046 | <b>176</b> | 0.508 | 0.04  |
| <b>49</b>  | 177 | 0.507 | 0.048 | <b>169</b> | 0.512 | 0.055 | <b>195</b> | 0.505 | 0.043 | <b>177</b> | 0.508 | 0.049 |
| <b>113</b> | 174 | 0.507 | 0.048 | <b>209</b> | 0.504 | 0.037 | <b>175</b> | 0.511 | 0.057 | <b>178</b> | 0.508 | 0.047 |
| <b>36</b>  | 183 | 0.505 | 0.035 | <b>174</b> | 0.509 | 0.038 | <b>180</b> | 0.509 | 0.039 | <b>179</b> | 0.508 | 0.037 |
| <b>35</b>  | 178 | 0.506 | 0.045 | <b>179</b> | 0.508 | 0.045 | <b>187</b> | 0.507 | 0.041 | <b>180</b> | 0.507 | 0.044 |
| <b>198</b> | 211 | 0.503 | 0.038 | <b>226</b> | 0.503 | 0.04  | <b>174</b> | 0.513 | 0.04  | <b>181</b> | 0.506 | 0.04  |
| <b>179</b> | 207 | 0.503 | 0.029 | <b>182</b> | 0.506 | 0.036 | <b>178</b> | 0.509 | 0.047 | <b>182</b> | 0.506 | 0.037 |
| <b>108</b> | 244 | 0.501 | 0.033 | <b>176</b> | 0.508 | 0.035 | <b>183</b> | 0.508 | 0.039 | <b>183</b> | 0.506 | 0.036 |
| <b>164</b> | 205 | 0.503 | 0.033 | <b>177</b> | 0.508 | 0.041 | <b>192</b> | 0.505 | 0.038 | <b>184</b> | 0.506 | 0.037 |
| <b>178</b> | 212 | 0.503 | 0.03  | <b>188</b> | 0.506 | 0.036 | <b>181</b> | 0.508 | 0.047 | <b>185</b> | 0.505 | 0.037 |
| <b>202</b> | 185 | 0.504 | 0.029 | <b>183</b> | 0.506 | 0.031 | <b>190</b> | 0.506 | 0.039 | <b>186</b> | 0.505 | 0.033 |
| <b>43</b>  | 189 | 0.504 | 0.047 | <b>181</b> | 0.507 | 0.041 | <b>204</b> | 0.504 | 0.041 | <b>187</b> | 0.505 | 0.043 |
| <b>156</b> | 209 | 0.503 | 0.032 | <b>195</b> | 0.505 | 0.036 | <b>182</b> | 0.508 | 0.041 | <b>188</b> | 0.505 | 0.037 |
| <b>172</b> | 175 | 0.507 | 0.031 | <b>224</b> | 0.504 | 0.029 | <b>200</b> | 0.505 | 0.033 | <b>189</b> | 0.505 | 0.031 |
| <b>50</b>  | 197 | 0.503 | 0.028 | <b>189</b> | 0.506 | 0.033 | <b>194</b> | 0.505 | 0.029 | <b>190</b> | 0.505 | 0.03  |
| <b>224</b> | 224 | 0.502 | 0.037 | <b>178</b> | 0.508 | 0.04  | <b>201</b> | 0.505 | 0.033 | <b>191</b> | 0.505 | 0.037 |
| <b>168</b> | 222 | 0.502 | 0.034 | <b>180</b> | 0.507 | 0.034 | <b>195</b> | 0.505 | 0.031 | <b>192</b> | 0.505 | 0.033 |
| <b>92</b>  | 184 | 0.504 | 0.031 | <b>240</b> | 0.503 | 0.031 | <b>186</b> | 0.507 | 0.034 | <b>193</b> | 0.505 | 0.032 |
| <b>104</b> | 180 | 0.505 | 0.033 | <b>253</b> | 0.501 | 0.038 | <b>184</b> | 0.507 | 0.039 | <b>194</b> | 0.505 | 0.037 |
| <b>216</b> | 179 | 0.505 | 0.032 | <b>230</b> | 0.503 | 0.029 | <b>198</b> | 0.505 | 0.031 | <b>195</b> | 0.505 | 0.031 |
| <b>23</b>  | 186 | 0.504 | 0.033 | <b>184</b> | 0.506 | 0.033 | <b>224</b> | 0.503 | 0.039 | <b>196</b> | 0.504 | 0.035 |
| <b>197</b> | 208 | 0.503 | 0.031 | <b>232</b> | 0.503 | 0.034 | <b>185</b> | 0.507 | 0.033 | <b>197</b> | 0.504 | 0.033 |
| <b>55</b>  | 195 | 0.503 | 0.033 | <b>191</b> | 0.505 | 0.034 | <b>209</b> | 0.504 | 0.03  | <b>198</b> | 0.504 | 0.032 |
| <b>222</b> | 228 | 0.502 | 0.023 | <b>205</b> | 0.504 | 0.027 | <b>188</b> | 0.506 | 0.03  | <b>199</b> | 0.504 | 0.027 |
| <b>141</b> | 190 | 0.504 | 0.028 | <b>238</b> | 0.503 | 0.032 | <b>192</b> | 0.505 | 0.032 | <b>200</b> | 0.504 | 0.031 |
| <b>40</b>  | 215 | 0.502 | 0.021 | <b>190</b> | 0.506 | 0.036 | <b>207</b> | 0.504 | 0.036 | <b>201</b> | 0.504 | 0.031 |
| <b>201</b> | 182 | 0.505 | 0.031 | <b>223</b> | 0.504 | 0.035 | <b>210</b> | 0.504 | 0.03  | <b>202</b> | 0.504 | 0.032 |
| <b>100</b> | 194 | 0.504 | 0.035 | <b>222</b> | 0.504 | 0.037 | <b>203</b> | 0.505 | 0.037 | <b>203</b> | 0.504 | 0.036 |
| <b>44</b>  | 234 | 0.502 | 0.031 | <b>197</b> | 0.505 | 0.036 | <b>192</b> | 0.505 | 0.036 | <b>204</b> | 0.504 | 0.034 |
| <b>78</b>  | 216 | 0.502 | 0.029 | <b>242</b> | 0.503 | 0.035 | <b>188</b> | 0.506 | 0.035 | <b>205</b> | 0.504 | 0.033 |
| <b>212</b> | 198 | 0.503 | 0.026 | <b>235</b> | 0.503 | 0.033 | <b>201</b> | 0.505 | 0.028 | <b>206</b> | 0.504 | 0.029 |
| <b>77</b>  | 196 | 0.503 | 0.026 | <b>205</b> | 0.504 | 0.033 | <b>216</b> | 0.503 | 0.029 | <b>207</b> | 0.504 | 0.029 |
| <b>32</b>  | 227 | 0.502 | 0.017 | <b>207</b> | 0.504 | 0.034 | <b>199</b> | 0.505 | 0.031 | <b>208</b> | 0.504 | 0.027 |
| <b>218</b> | 200 | 0.503 | 0.028 | <b>192</b> | 0.505 | 0.033 | <b>238</b> | 0.502 | 0.033 | <b>209</b> | 0.504 | 0.031 |
| <b>219</b> | 187 | 0.504 | 0.029 | <b>202</b> | 0.505 | 0.033 | <b>247</b> | 0.501 | 0.025 | <b>210</b> | 0.503 | 0.029 |
| <b>233</b> | 202 | 0.503 | 0.026 | <b>193</b> | 0.505 | 0.031 | <b>243</b> | 0.502 | 0.028 | <b>211</b> | 0.503 | 0.028 |
| <b>12</b>  | 239 | 0.501 | 0.031 | <b>208</b> | 0.504 | 0.032 | <b>205</b> | 0.504 | 0.028 | <b>212</b> | 0.503 | 0.03  |
| <b>228</b> | 201 | 0.503 | 0.032 | <b>252</b> | 0.501 | 0.027 | <b>197</b> | 0.505 | 0.033 | <b>213</b> | 0.503 | 0.031 |
| <b>19</b>  | 206 | 0.503 | 0.029 | <b>194</b> | 0.505 | 0.035 | <b>250</b> | 0.501 | 0.029 | <b>214</b> | 0.503 | 0.031 |
| <b>203</b> | 251 | 0.5   | 0.026 | <b>187</b> | 0.506 | 0.033 | <b>212</b> | 0.503 | 0.032 | <b>215</b> | 0.503 | 0.031 |
| <b>96</b>  | 219 | 0.502 | 0.024 | <b>225</b> | 0.504 | 0.038 | <b>211</b> | 0.503 | 0.029 | <b>216</b> | 0.503 | 0.03  |
| <b>249</b> | 223 | 0.502 | 0.015 | <b>204</b> | 0.504 | 0.026 | <b>223</b> | 0.503 | 0.017 | <b>217</b> | 0.503 | 0.019 |
| <b>237</b> | 221 | 0.502 | 0.025 | <b>215</b> | 0.504 | 0.03  | <b>225</b> | 0.503 | 0.025 | <b>218</b> | 0.503 | 0.027 |
| <b>234</b> | 242 | 0.501 | 0.018 | <b>201</b> | 0.505 | 0.03  | <b>218</b> | 0.503 | 0.023 | <b>219</b> | 0.503 | 0.023 |
| <b>250</b> | 229 | 0.502 | 0.014 | <b>218</b> | 0.504 | 0.023 | <b>220</b> | 0.503 | 0.021 | <b>220</b> | 0.503 | 0.019 |
| <b>142</b> | 192 | 0.504 | 0.026 | <b>256</b> | 0.5   | 0.035 | <b>208</b> | 0.504 | 0.031 | <b>221</b> | 0.503 | 0.03  |
| <b>239</b> | 232 | 0.502 | 0.02  | <b>200</b> | 0.505 | 0.027 | <b>235</b> | 0.502 | 0.018 | <b>222</b> | 0.503 | 0.022 |
| <b>0</b>   | 233 | 0.502 | 0.019 | <b>214</b> | 0.504 | 0.029 | <b>222</b> | 0.503 | 0.02  | <b>223</b> | 0.503 | 0.023 |
| <b>238</b> | 219 | 0.502 | 0.021 | <b>217</b> | 0.504 | 0.032 | <b>232</b> | 0.502 | 0.024 | <b>224</b> | 0.503 | 0.026 |
| <b>221</b> | 193 | 0.504 | 0.028 | <b>247</b> | 0.502 | 0.037 | <b>230</b> | 0.502 | 0.027 | <b>225</b> | 0.503 | 0.031 |
| <b>252</b> | 218 | 0.502 | 0.024 | <b>219</b> | 0.504 | 0.028 | <b>233</b> | 0.502 | 0.025 | <b>226</b> | 0.503 | 0.026 |
| <b>217</b> | 255 | 0.499 | 0.027 | <b>199</b> | 0.505 | 0.034 | <b>206</b> | 0.504 | 0.028 | <b>227</b> | 0.503 | 0.029 |
| <b>76</b>  | 204 | 0.503 | 0.028 | <b>243</b> | 0.503 | 0.029 | <b>228</b> | 0.502 | 0.031 | <b>228</b> | 0.503 | 0.029 |
| <b>4</b>   | 230 | 0.502 | 0.03  | <b>198</b> | 0.505 | 0.036 | <b>244</b> | 0.502 | 0.036 | <b>229</b> | 0.503 | 0.034 |
| <b>136</b> | 214 | 0.502 | 0.028 | <b>220</b> | 0.504 | 0.038 | <b>242</b> | 0.502 | 0.032 | <b>230</b> | 0.503 | 0.033 |
| <b>223</b> | 231 | 0.502 | 0.031 | <b>212</b> | 0.504 | 0.027 | <b>231</b> | 0.502 | 0.026 | <b>231</b> | 0.503 | 0.028 |
| <b>8</b>   | 247 | 0.501 | 0.023 | <b>203</b> | 0.504 | 0.031 | <b>217</b> | 0.503 | 0.029 | <b>232</b> | 0.503 | 0.028 |
| <b>236</b> | 210 | 0.503 | 0.022 | <b>235</b> | 0.503 | 0.025 | <b>236</b> | 0.502 | 0.024 | <b>233</b> | 0.503 | 0.024 |
| <b>160</b> | 225 | 0.502 | 0.02  | <b>230</b> | 0.503 | 0.025 | <b>218</b> | 0.503 | 0.021 | <b>234</b> | 0.503 | 0.022 |
| <b>128</b> | 217 | 0.502 | 0.02  | <b>229</b> | 0.503 | 0.023 | <b>237</b> | 0.502 | 0.022 | <b>235</b> | 0.503 | 0.022 |
| <b>251</b> | 238 | 0.501 | 0.014 | <b>233</b> | 0.503 | 0.022 | <b>220</b> | 0.503 | 0.018 | <b>236</b> | 0.502 | 0.018 |
| <b>235</b> | 243 | 0.501 | 0.016 | <b>220</b> | 0.504 | 0.027 | <b>226</b> | 0.503 | 0.021 | <b>237</b> | 0.502 | 0.021 |
| <b>200</b> | 191 | 0.504 | 0.033 | <b>255</b> | 0.501 | 0.034 | <b>234</b> | 0.502 | 0.031 | <b>238</b> | 0.502 | 0.032 |
| <b>64</b>  | 245 | 0.501 | 0.023 | <b>211</b> | 0.504 | 0.025 | <b>239</b> | 0.502 | 0.024 | <b>239</b> | 0.502 | 0.024 |
| <b>232</b> | 235 | 0.501 | 0.026 | <b>249</b> | 0.502 | 0.028 | <b>215</b> | 0.503 | 0.025 | <b>240</b> | 0.502 | 0.026 |
| <b>192</b> | 203 | 0.503 | 0.029 | <b>246</b> | 0.502 | 0.037 | <b>249</b> | 0.501 | 0.027 | <b>241</b> | 0.502 | 0.031 |
| <b>72</b>  | 199 | 0.503 | 0.033 | <b>251</b> | 0.502 | 0.037 | <b>248</b> | 0.501 | 0.033 | <b>242</b> | 0.502 | 0.034 |
| <b>253</b> | 226 | 0.502 | 0.021 | <b>244</b> | 0.502 | 0.018 | <b>229</b> | 0.502 | 0.022 | <b>243</b> | 0.502 | 0.02  |
| <b>254</b> | 248 | 0.5   | 0.013 | <b>213</b> | 0.504 | 0.025 | <b>241</b> | 0.502 | 0.018 | <b>244</b> | 0.502 | 0.018 |
| <b>255</b> | 246 | 0.501 | 0.017 | <b>237</b> | 0.503 | 0.022 | <b>227</b> | 0.503 | 0.018 | <b>245</b> | 0.502 | 0.019 |
| <b>51</b>  | 250 | 0.5   | 0.023 | <b>210</b> | 0.504 | 0.032 | <b>240</b> | 0.502 | 0.028 | <b>246</b> | 0.502 | 0.028 |
| <b>207</b> | 253 | 0.5   | 0.023 | <b>238</b> | 0.503 | 0.028 | <b>212</b> | 0.503 | 0.031 | <b>247</b> | 0.502 | 0.027 |
| <b>140</b> | 213 | 0.502 | 0.025 | <b>249</b> | 0.502 | 0.032 | <b>253</b> | 0.501 | 0.029 | <b>248</b> | 0.502 | 0.029 |
| <b>196</b> | 236 | 0.501 | 0.024 | <b>235</b> | 0.503 | 0.027 | <b>254</b> | 0.501 | 0.029 | <b>249</b> | 0.502 | 0.027 |
| <b>206</b> | 241 | 0.501 | 0.025 | <b>228</b> | 0.503 | 0.026 | <b>255</b> | 0.501 | 0.028 | <b>250</b> | 0.502 | 0.026 |
| <b>220</b> | 254 | 0.499 | 0.024 | <b>241</b> | 0.503 | 0.031 | <b>214</b> | 0.503 | 0.028 | <b>251</b> | 0.502 | 0.028 |
| <b>248</b> | 240 | 0.501 | 0.017 | <b>248</b> | 0.502 | 0.025 | <b>244</b> | 0.502 | 0.021 | <b>252</b> | 0.502 | 0.021 |
| <b>204</b> | 237 | 0.501 | 0.027 | <b>244</b> | 0.502 | 0.029 | <b>251</b> | 0.501 | 0.027 | <b>253</b> | 0.502 | 0.028 |
| <b>68</b>  | 252 | 0.5   | 0.028 | <b>215</b> | 0.504 | 0.036 | <b>251</b> | 0.501 | 0.029 |            |       |       |

Table A.3: Full information gain table, including the entropy  $H(Y)$ , the information gain for the current (“middle”) cell  $IG_M$ , and the information gain for predecessors  $IG_A$ ,  $IG_B$ ,  $IG_C$  and  $IG_D$ , corresponding to the order the predecessors appear in the list of predecessors. The information gain is calculated for cells with two, three and four predecessors, separately.

| $R_i$ | Two Predecessors |        |        |        | Three Predecessors |        |        |        |        | Four Predecessors |        |        |        |        |        |
|-------|------------------|--------|--------|--------|--------------------|--------|--------|--------|--------|-------------------|--------|--------|--------|--------|--------|
|       | $H(Y)$           | $IG_M$ | $IG_A$ | $IG_B$ | $H(Y)$             | $IG_M$ | $IG_A$ | $IG_B$ | $IG_C$ | $H(Y)$            | $IG_M$ | $IG_A$ | $IG_B$ | $IG_C$ | $IG_D$ |
| 0     | -0               | 0      | 0      | 0      | -0                 | 0      | 0      | 0      | 0      | -0                | 0      | 0      | 0      | 0      | 0      |
| 1     | 0.544            | 0.138  | 0.138  | 0.138  | 0.337              | 0.066  | 0.066  | 0.066  | 0.066  | 0.625             | 0.177  | 0.051  | 0.051  | 0.051  | 0.051  |
| 2     | 0.544            | 0.138  | 0.138  | 0.138  | 0.544              | 0.138  | 0.138  | 0      | 0.138  | 0.544             | 0.138  | 0.138  | 0      | 0      | 0.138  |
| 3     | 0.811            | 0.311  | 0.311  | 0      | 0.811              | 0.311  | 0.311  | 0      | 0      | 0.896             | 0.419  | 0.124  | 0.013  | 0.013  | 0      |
| 4     | 0.544            | 0.138  | 0.138  | 0.138  | 0.337              | 0.066  | 0.066  | 0.066  | 0.066  | 0.625             | 0.177  | 0.051  | 0.051  | 0.051  | 0.051  |
| 5     | 0.811            | 0      | 0.311  | 0.311  | 0.544              | 0      | 0.138  | 0.138  | 0.138  | 0.896             | 0      | 0.124  | 0.124  | 0.124  | 0.124  |
| 6     | 0.811            | 0      | 0.311  | 0      | 0.696              | 0.019  | 0.219  | 0.019  | 0.019  | 0.857             | 0.003  | 0.189  | 0.032  | 0.032  | 0.003  |
| 7     | 0.954            | 0.049  | 0.549  | 0.049  | 0.896              | 0.124  | 0.419  | 0.013  | 0.013  | 0.997             | 0.072  | 0.243  | 0.072  | 0.072  | 0.026  |
| 8     | 0.544            | 0.138  | 0.138  | 0.138  | 0.544              | 0.138  | 0.138  | 0      | 0.138  | 0.544             | 0.138  | 0.138  | 0      | 0      | 0.138  |
| 9     | 0.811            | 0      | 0.311  | 0      | 0.696              | 0.019  | 0.219  | 0.019  | 0.019  | 0.857             | 0.003  | 0.189  | 0.032  | 0.032  | 0.003  |
| 10    | 0.811            | 0      | 0.311  | 0.311  | 0.811              | 0      | 0.311  | 0      | 0.311  | 0.811             | 0      | 0.311  | 0      | 0      | 0.311  |
| 11    | 0.954            | 0.049  | 0.549  | 0.049  | 0.954              | 0.049  | 0.549  | 0      | 0.049  | 0.989             | 0.106  | 0.311  | 0.011  | 0.011  | 0.046  |
| 12    | 0.811            | 0.311  | 0.311  | 0      | 0.811              | 0.311  | 0.311  | 0      | 0      | 0.896             | 0.419  | 0.124  | 0.013  | 0.013  | 0      |
| 13    | 0.954            | 0.049  | 0.549  | 0.049  | 0.896              | 0.124  | 0.419  | 0.013  | 0.013  | 0.997             | 0.072  | 0.243  | 0.072  | 0.072  | 0.026  |
| 14    | 0.954            | 0.049  | 0.549  | 0.049  | 0.954              | 0.049  | 0.549  | 0      | 0.049  | 0.989             | 0.106  | 0.311  | 0.011  | 0.011  | 0.046  |
| 15    | 1                | 0      | 1      | 0      | 1                  | 0      | 1      | 0      | 0      | 0.954             | 0      | 0.549  | 0.049  | 0.049  | 0      |
| 16    | 0.544            | 0.138  | 0.138  | 0.138  | 0.544              | 0.138  | 0.138  | 0      | 0.138  | 0.544             | 0.138  | 0.138  | 0      | 0      | 0.138  |
| 17    | 0.811            | 0.311  | 0      | 0.311  | 0.811              | 0.311  | 0      | 0      | 0.311  | 0.896             | 0.419  | 0      | 0.013  | 0.013  | 0.124  |
| 18    | 0.811            | 0.311  | 0      | 0      | 0.954              | 0.549  | 0      | 0      | 0      | 0.989             | 0.717  | 0      | 0      | 0      | 0      |
| 19    | 0.954            | 0.549  | 0.049  | 0.049  | 0.989              | 0.717  | 0.011  | 0.011  | 0.011  | 0.997             | 0.829  | 0.003  | 0.003  | 0.003  | 0.003  |
| 20    | 0.811            | 0      | 0      | 0.311  | 0.696              | 0.019  | 0.019  | 0.019  | 0.219  | 0.857             | 0.003  | 0.003  | 0.032  | 0.032  | 0.189  |
| 21    | 0.954            | 0.049  | 0.049  | 0.549  | 0.896              | 0.124  | 0.013  | 0.013  | 0.419  | 0.997             | 0.072  | 0.026  | 0.072  | 0.072  | 0.243  |
| 22    | 0.954            | 0.049  | 0.049  | 0.049  | 0.989              | 0.311  | 0.011  | 0.011  | 0.011  | 0.974             | 0.255  | 0.026  | 0.026  | 0.026  | 0.026  |
| 23    | 1                | 0.189  | 0.189  | 0.189  | 1                  | 0.456  | 0.046  | 0.046  | 0.046  | 0.954             | 0.338  | 0.049  | 0.049  | 0.049  | 0.049  |
| 24    | 0.811            | 0      | 0      | 0      | 0.811              | 0      | 0      | 0      | 0      | 0.811             | 0      | 0      | 0      | 0      | 0      |
| 25    | 0.954            | 0.049  | 0.049  | 0.049  | 0.954              | 0.049  | 0.049  | 0      | 0.049  | 0.989             | 0.106  | 0.046  | 0.011  | 0.011  | 0.011  |
| 26    | 0.954            | 0.049  | 0.049  | 0.049  | 1                  | 0.189  | 0.046  | 0      | 0.046  | 0.989             | 0.311  | 0.046  | 0      | 0      | 0.046  |
| 27    | 1                | 0.189  | 0.189  | 0      | 0.989              | 0.311  | 0.106  | 0.011  | 0.011  | 0.974             | 0.4    | 0.075  | 0.003  | 0.003  | 0.026  |
| 28    | 0.954            | 0.049  | 0.049  | 0.049  | 0.954              | 0.049  | 0.049  | 0      | 0.049  | 0.989             | 0.106  | 0.011  | 0.011  | 0.011  | 0.046  |
| 29    | 1                | 0      | 0.189  | 0.189  | 1                  | 0      | 0.189  | 0      | 0.189  | 0.954             | 0      | 0.112  | 0.049  | 0.049  | 0.112  |
| 30    | 1                | 0      | 0.189  | 0      | 0.954              | 0.049  | 0.205  | 0      | 0      | 0.811             | 0.062  | 0.148  | 0.015  | 0.015  | 0      |
| 31    | 0.954            | 0.049  | 0.549  | 0.049  | 0.896              | 0.124  | 0.419  | 0.013  | 0.013  | 0.758             | 0.112  | 0.264  | 0.038  | 0.038  | 0.004  |
| 32    | 0.544            | 0.138  | 0.138  | 0.138  | 0.337              | 0.066  | 0.066  | 0.066  | 0.066  | 0.625             | 0.177  | 0.051  | 0.051  | 0.051  | 0.051  |
| 33    | 0.811            | 0.311  | 0      | 0      | 0.544              | 0.138  | 0      | 0      | 0      | 0.896             | 0.419  | 0      | 0      | 0      | 0      |
| 34    | 0.811            | 0.311  | 0      | 0.311  | 0.811              | 0.311  | 0      | 0      | 0.311  | 0.896             | 0.419  | 0      | 0.013  | 0.013  | 0.124  |
| 35    | 0.954            | 0.549  | 0.049  | 0.049  | 0.954              | 0.549  | 0.049  | 0      | 0.049  | 0.997             | 0.829  | 0.003  | 0.003  | 0.003  | 0.003  |
| 36    | 0.811            | 0      | 0      | 0      | 0.544              | 0      | 0      | 0      | 0      | 0.896             | 0      | 0      | 0      | 0      | 0      |
| 37    | 0.954            | 0.049  | 0.049  | 0.049  | 0.696              | 0.019  | 0.019  | 0.019  | 0.019  | 0.997             | 0.072  | 0.026  | 0.026  | 0.026  | 0.026  |
| 38    | 0.954            | 0.049  | 0.049  | 0.049  | 0.896              | 0.124  | 0.013  | 0.013  | 0.124  | 0.997             | 0.072  | 0.026  | 0.003  | 0.003  | 0.026  |
| 39    | 1                | 0.189  | 0.189  | 0      | 0.989              | 0.311  | 0.106  | 0.011  | 0.011  | 0.954             | 0.338  | 0.049  | 0.049  | 0.012  | 0.012  |
| 40    | 0.811            | 0      | 0      | 0.311  | 0.696              | 0.019  | 0.019  | 0.019  | 0.219  | 0.857             | 0.003  | 0.003  | 0.032  | 0.032  | 0.189  |
| 41    | 0.954            | 0.049  | 0.049  | 0.049  | 0.811              | 0      | 0.062  | 0      | 0.062  | 0.989             | 0.106  | 0.046  | 0      | 0      | 0.046  |
| 42    | 0.954            | 0.049  | 0.049  | 0.549  | 0.954              | 0.049  | 0.049  | 0      | 0.549  | 0.989             | 0.106  | 0.046  | 0.011  | 0.011  | 0.311  |
| 43    | 1                | 0.189  | 0.189  | 0.189  | 1                  | 0.189  | 0.189  | 0      | 0.189  | 0.974             | 0.4    | 0.075  | 0.003  | 0.003  | 0.075  |
| 44    | 0.954            | 0.049  | 0.049  | 0.049  | 0.896              | 0.124  | 0.124  | 0.013  | 0.013  | 0.997             | 0.072  | 0.026  | 0.003  | 0.003  | 0.026  |
| 45    | 1                | 0      | 0.189  | 0      | 0.954              | 0.049  | 0.205  | 0      | 0      | 0.954             | 0      | 0.112  | 0.012  | 0.012  | 0      |
| 46    | 1                | 0      | 0.189  | 0.189  | 1                  | 0      | 0.189  | 0      | 0.189  | 0.954             | 0      | 0.112  | 0      | 0      | 0.112  |
| 47    | 0.954            | 0.049  | 0.549  | 0.049  | 0.954              | 0.049  | 0.549  | 0      | 0.049  | 0.758             | 0.112  | 0.264  | 0.038  | 0.004  | 0.004  |
| 48    | 0.811            | 0.311  | 0.311  | 0      | 0.811              | 0.311  | 0.311  | 0      | 0      | 0.896             | 0.419  | 0.124  | 0.013  | 0.013  | 0      |
| 49    | 0.954            | 0.549  | 0.049  | 0.049  | 0.954              | 0.549  | 0.049  | 0      | 0.049  | 0.997             | 0.829  | 0.003  | 0.003  | 0.003  | 0.003  |
| 50    | 0.954            | 0.549  | 0.049  | 0.049  | 0.989              | 0.717  | 0.011  | 0.011  | 0.011  | 0.997             | 0.829  | 0.003  | 0.003  | 0.003  | 0.003  |
| 51    | 1                | 1      | 0      | 0      | 1                  | 1      | 0      | 0      | 0      | 1                 | 0      | 0      | 0      | 0      | 0      |
| 52    | 0.954            | 0.049  | 0.049  | 0.049  | 0.896              | 0.124  | 0.124  | 0.013  | 0.013  | 0.997             | 0.072  | 0.026  | 0.003  | 0.003  | 0.026  |
| 53    | 1                | 0.189  | 0      | 0.189  | 0.989              | 0.311  | 0.011  | 0.011  | 0.106  | 0.954             | 0.338  | 0.012  | 0.012  | 0.049  | 0.049  |
| 54    | 1                | 0.189  | 0      | 0      | 1                  | 0.456  | 0      | 0      | 0      | 0.954             | 0.338  | 0.012  | 0.012  | 0.012  | 0.012  |
| 55    | 0.954            | 0.549  | 0.049  | 0.049  | 0.989              | 0.717  | 0.011  | 0.011  | 0.011  | 0.928             | 0.48   | 0.028  | 0.028  | 0.028  | 0.028  |
| 56    | 0.954            | 0.049  | 0.049  | 0.049  | 0.954              | 0.049  | 0.049  | 0      | 0.049  | 0.989             | 0.106  | 0.011  | 0.011  | 0.011  | 0.046  |
| 57    | 1                | 0.189  | 0      | 0      | 1                  | 0.189  | 0      | 0      | 0      | 0.974             | 0.4    | 0.026  | 0.003  | 0.003  | 0.026  |
| 58    | 1                | 0.189  | 0      | 0.189  | 0.989              | 0.311  | 0.011  | 0.011  | 0.106  | 0.974             | 0.4    | 0.026  | 0.003  | 0.003  | 0.075  |
| 59    | 0.954            | 0.549  | 0.049  | 0.049  | 0.954              | 0.549  | 0.049  | 0      | 0.049  | 0.954             | 0.549  | 0.049  | 0      | 0      | 0.049  |
| 60    | 1                | 0      | 0      | 0      | 1                  | 0      | 0      | 0      | 0      | 0.954             | 0      | 0      | 0      | 0      | 0      |
| 61    | 0.954            | 0.049  | 0.049  | 0.049  | 0.954              | 0.049  | 0.049  | 0      | 0.049  | 0.758             | 0.112  | 0.112  | 0.004  | 0.038  | 0.004  |
| 62    | 0.954            | 0.049  | 0.049  | 0.049  | 0.896              | 0.124  | 0.124  | 0.013  | 0.013  | 0.758             | 0.112  | 0.112  | 0.004  | 0.004  | 0.004  |
| 63    | 0.811            | 0.311  | 0.311  | 0      | 0.811              | 0.311  | 0.311  | 0      | 0      | 0.696             | 0.219  | 0.219  | 0.019  | 0.019  | 0      |
| 64    | 0.544            | 0.138  | 0.138  | 0.138  | 0.544              | 0.138  | 0.138  | 0      | 0.138  | 0.544             | 0.138  | 0.138  | 0      | 0      | 0.138  |
| 65    | 0.811            | 0      | 0      | 0.311  | 0.696              | 0.019  | 0.019  | 0.019  | 0.219  | 0.857             | 0.003  | 0.003  | 0.032  | 0.032  | 0.189  |
| 66    | 0.811            | 0      | 0      | 0      | 0.811              | 0      | 0      | 0      | 0      | 0.811             | 0      | 0      | 0      | 0      | 0      |
| 67    | 0.954            | 0.049  | 0.049  | 0.049  | 0.954              | 0.049  | 0.049  | 0      | 0.049  | 0.989             | 0.106  | 0.011  | 0.011  | 0.011  | 0.046  |
| 68    | 0.811            | 0.311  | 0      | 0.311  | 0.811              | 0.311  | 0      | 0      | 0.311  | 0.896             | 0.419  | 0      | 0.013  | 0.013  | 0.124  |
| 69    | 0.954            | 0.049  | 0.049  | 0.549  | 0.896              | 0.124  | 0.013  | 0.013  | 0.419  | 0.997             | 0.072  | 0.026  | 0.072  | 0.072  | 0.243  |
| 70    | 0.954            | 0.049  | 0.049  | 0.049  | 0.954              | 0.049  | 0.049  | 0      | 0.049  | 0.989             | 0.106  | 0.046  | 0.011  | 0.011  | 0.011  |
| 71    | 1                | 0      | 0.189  | 0.189  | 1                  | 0      | 0.189  | 0      | 0.189  | 0.954             | 0      | 0.112  | 0.049  | 0.049  | 0.112  |
| 72    | 0.811            | 0.311  | 0      | 0      | 0.954              | 0.549  | 0      | 0      | 0      | 0.989             | 0.717  | 0      | 0      | 0      | 0      |
| 73    | 0.954            | 0.049  | 0.049  | 0.049  | 0.989              | 0.311  | 0.011  | 0.011  | 0.011  | 0.974             | 0.255  | 0.026  | 0.026  | 0.026  | 0.026  |
| 74    | 0.954            | 0.049  | 0.049  | 0.049  | 1                  | 0.189  | 0.046  | 0      | 0.046  | 0.989             | 0.311  | 0.046  | 0      | 0      | 0.046  |
| 75    | 1                | 0      | 0.189  | 0      | 0.954              | 0.049  | 0.205  | 0      | 0      | 0.811             | 0.062  | 0.148  | 0.015  | 0.015  | 0      |
| 76    | 0.954            | 0.549  | 0.049  | 0.049  | 0.989              | 0.717  | 0.011  | 0.011  | 0.011  | 0.997             | 0.829  | 0.003  | 0.003  | 0.003  | 0.003  |
| 77    | 1                | 0.189  | 0.189  | 0.189  | 1                  | 0.456  | 0.046  | 0.046  | 0.046  | 0.954             | 0.338  | 0.049  | 0.049  | 0.049  | 0.049  |
| 78    | 1                | 0.189  | 0.189  | 0      | 0.989              | 0.311  | 0.106  | 0.011  | 0.011  | 0.974             | 0.4    | 0.075  | 0.003  | 0.003  | 0.026  |
| 79    | 0.954            | 0.049  | 0.549  | 0.049  | 0.896              | 0.124  | 0.419  | 0.013  | 0.013  | 0.758             | 0.112  | 0.264  | 0.038  | 0.038  | 0.004  |
| 80    | 0.811            | 0      | 0.311  | 0.311  | 0.811              | 0      | 0.311  |        |        |                   |        |        |        |        |        |

# A. Appendix 1

|     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 82  | 0.954 | 0.049 | 0.049 | 0.049 | 1     | 0.189 | 0.046 | 0     | 0.046 | 0.989 | 0.311 | 0.046 | 0     | 0     | 0.046 |
| 83  | 1     | 0.189 | 0     | 0.189 | 0.989 | 0.311 | 0.011 | 0.011 | 0.106 | 0.974 | 0.4   | 0.026 | 0.003 | 0.003 | 0.075 |
| 84  | 0.954 | 0.049 | 0.049 | 0.549 | 0.954 | 0.049 | 0.049 | 0     | 0.549 | 0.989 | 0.106 | 0.046 | 0.011 | 0.011 | 0.311 |
| 85  | 1     | 0     | 0     | 1     | 1     | 0     | 0     | 0     | 1     | 0.954 | 0     | 0     | 0.049 | 0.049 | 0.549 |
| 86  | 1     | 0     | 0     | 0.189 | 0.954 | 0.049 | 0     | 0     | 0.205 | 0.811 | 0.062 | 0     | 0.015 | 0.015 | 0.148 |
| 87  | 0.954 | 0.049 | 0.049 | 0.549 | 0.896 | 0.124 | 0.013 | 0.013 | 0.419 | 0.758 | 0.112 | 0.004 | 0.038 | 0.038 | 0.264 |
| 88  | 0.954 | 0.049 | 0.049 | 0.049 | 1     | 0.189 | 0.046 | 0     | 0.046 | 0.989 | 0.311 | 0.046 | 0     | 0     | 0.046 |
| 89  | 1     | 0     | 0     | 0.189 | 0.954 | 0.049 | 0     | 0     | 0.205 | 0.811 | 0.062 | 0     | 0.015 | 0.015 | 0.148 |
| 90  | 1     | 0     | 0     | 0     | 0.811 | 0     | 0     | 0     | 0     | 0.544 | 0     | 0     | 0     | 0     | 0     |
| 91  | 0.954 | 0.049 | 0.049 | 0.049 | 0.696 | 0.019 | 0.019 | 0.019 | 0.019 | 0.449 | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 |
| 92  | 1     | 0.189 | 0     | 0.189 | 0.989 | 0.311 | 0.011 | 0.011 | 0.106 | 0.974 | 0.4   | 0.026 | 0.003 | 0.003 | 0.075 |
| 93  | 0.954 | 0.049 | 0.049 | 0.549 | 0.896 | 0.124 | 0.013 | 0.013 | 0.419 | 0.758 | 0.112 | 0.004 | 0.038 | 0.038 | 0.264 |
| 94  | 0.954 | 0.049 | 0.049 | 0.049 | 0.696 | 0.019 | 0.019 | 0.019 | 0.019 | 0.449 | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 |
| 95  | 0.811 | 0     | 0.311 | 0.311 | 0.544 | 0     | 0.138 | 0.138 | 0.138 | 0.337 | 0     | 0.066 | 0.066 | 0.066 | 0.066 |
| 96  | 0.811 | 0     | 0.311 | 0     | 0.696 | 0.019 | 0.219 | 0.019 | 0.019 | 0.857 | 0.003 | 0.189 | 0.032 | 0.032 | 0.003 |
| 97  | 0.954 | 0.049 | 0.049 | 0.049 | 0.811 | 0     | 0.062 | 0     | 0.062 | 0.989 | 0.106 | 0.046 | 0     | 0     | 0.046 |
| 98  | 0.954 | 0.049 | 0.049 | 0.049 | 0.954 | 0.049 | 0.049 | 0     | 0.049 | 0.989 | 0.106 | 0.046 | 0.011 | 0.011 | 0.011 |
| 99  | 1     | 0.189 | 0     | 0     | 1     | 0.189 | 0     | 0     | 0     | 0.974 | 0.4   | 0.026 | 0.003 | 0.003 | 0.026 |
| 100 | 0.954 | 0.049 | 0.049 | 0.049 | 0.896 | 0.124 | 0.013 | 0.013 | 0.124 | 0.997 | 0.072 | 0.026 | 0.003 | 0.003 | 0.026 |
| 101 | 1     | 0     | 0     | 0.189 | 0.954 | 0.049 | 0     | 0     | 0.205 | 0.954 | 0     | 0     | 0.012 | 0.012 | 0.112 |
| 102 | 1     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0.954 | 0     | 0     | 0     | 0     | 0     |
| 103 | 0.954 | 0.049 | 0.049 | 0.049 | 0.954 | 0.049 | 0.049 | 0     | 0.049 | 0.758 | 0.112 | 0.004 | 0.038 | 0.004 | 0.112 |
| 104 | 0.954 | 0.049 | 0.049 | 0.049 | 0.989 | 0.311 | 0.011 | 0.011 | 0.011 | 0.974 | 0.255 | 0.026 | 0.026 | 0.026 | 0.026 |
| 105 | 1     | 0     | 0     | 0     | 1     | 0.189 | 0     | 0     | 0     | 0.811 | 0.062 | 0     | 0     | 0     | 0     |
| 106 | 1     | 0     | 0     | 0.189 | 0.954 | 0.049 | 0     | 0     | 0.205 | 0.811 | 0.062 | 0     | 0.015 | 0.015 | 0.148 |
| 107 | 0.954 | 0.049 | 0.049 | 0.049 | 0.811 | 0     | 0.062 | 0     | 0.062 | 0.449 | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 |
| 108 | 1     | 0.189 | 0     | 0     | 1     | 0.456 | 0     | 0     | 0     | 0.954 | 0.338 | 0.012 | 0.012 | 0.012 | 0.012 |
| 109 | 0.954 | 0.049 | 0.049 | 0.049 | 0.989 | 0.311 | 0.011 | 0.011 | 0.011 | 0.758 | 0.112 | 0.004 | 0.004 | 0.004 | 0.004 |
| 110 | 0.954 | 0.049 | 0.049 | 0.049 | 0.896 | 0.124 | 0.013 | 0.013 | 0.124 | 0.758 | 0.112 | 0.004 | 0.004 | 0.004 | 0.112 |
| 111 | 0.811 | 0     | 0.311 | 0     | 0.696 | 0.019 | 0.219 | 0.019 | 0.019 | 0.337 | 0     | 0.066 | 0.066 | 0     | 0     |
| 112 | 0.954 | 0.049 | 0.549 | 0.049 | 0.954 | 0.049 | 0.549 | 0     | 0.049 | 0.989 | 0.106 | 0.311 | 0.011 | 0.011 | 0.046 |
| 113 | 1     | 0.189 | 0.189 | 0.189 | 1     | 0.189 | 0.189 | 0     | 0.189 | 0.974 | 0.4   | 0.075 | 0.003 | 0.003 | 0.075 |
| 114 | 1     | 0.189 | 0.189 | 0     | 0.989 | 0.311 | 0.106 | 0.011 | 0.011 | 0.974 | 0.4   | 0.075 | 0.003 | 0.003 | 0.026 |
| 115 | 0.954 | 0.549 | 0.049 | 0.049 | 0.954 | 0.549 | 0.049 | 0     | 0.049 | 0.954 | 0.549 | 0.049 | 0     | 0     | 0.049 |
| 116 | 1     | 0     | 0.189 | 0.189 | 1     | 0     | 0.189 | 0     | 0.189 | 0.954 | 0     | 0.112 | 0     | 0     | 0.112 |
| 117 | 0.954 | 0.049 | 0.049 | 0.549 | 0.954 | 0.049 | 0.049 | 0     | 0.549 | 0.758 | 0.112 | 0.004 | 0.004 | 0.038 | 0.264 |
| 118 | 0.954 | 0.049 | 0.049 | 0.049 | 0.896 | 0.124 | 0.013 | 0.013 | 0.124 | 0.758 | 0.112 | 0.004 | 0.004 | 0.004 | 0.112 |
| 119 | 0.811 | 0.311 | 0     | 0.311 | 0.811 | 0.311 | 0     | 0     | 0.311 | 0.696 | 0.219 | 0     | 0.019 | 0.019 | 0.219 |
| 120 | 1     | 0     | 0.189 | 0     | 0.954 | 0.049 | 0.205 | 0     | 0     | 0.811 | 0.062 | 0.148 | 0.015 | 0.015 | 0     |
| 121 | 0.954 | 0.049 | 0.049 | 0.049 | 0.811 | 0     | 0.062 | 0     | 0.062 | 0.449 | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 |
| 122 | 0.954 | 0.049 | 0.049 | 0.049 | 0.696 | 0.019 | 0.019 | 0.019 | 0.019 | 0.449 | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 |
| 123 | 0.811 | 0.311 | 0     | 0     | 0.544 | 0.138 | 0     | 0     | 0     | 0.337 | 0.066 | 0     | 0     | 0     | 0     |
| 124 | 0.954 | 0.049 | 0.049 | 0.049 | 0.896 | 0.124 | 0.124 | 0.013 | 0.013 | 0.758 | 0.112 | 0.112 | 0.004 | 0.004 | 0.004 |
| 125 | 0.811 | 0     | 0     | 0.311 | 0.696 | 0.019 | 0.019 | 0.019 | 0.219 | 0.337 | 0     | 0     | 0     | 0.066 | 0.066 |
| 126 | 0.811 | 0     | 0     | 0     | 0.544 | 0     | 0     | 0     | 0     | 0.337 | 0     | 0     | 0     | 0     | 0     |
| 127 | 0.544 | 0.138 | 0.138 | 0.138 | 0.337 | 0.066 | 0.066 | 0.066 | 0.066 | 0.201 | 0.032 | 0.032 | 0.032 | 0.032 | 0.032 |
| 128 | 0.544 | 0.138 | 0.138 | 0.138 | 0.337 | 0.066 | 0.066 | 0.066 | 0.066 | 0.625 | 0.177 | 0.051 | 0.051 | 0.051 | 0.051 |
| 129 | 0.811 | 0     | 0     | 0     | 0.544 | 0     | 0     | 0     | 0     | 0.896 | 0     | 0     | 0     | 0     | 0     |
| 130 | 0.811 | 0     | 0     | 0.311 | 0.696 | 0.019 | 0.019 | 0.019 | 0.219 | 0.857 | 0.003 | 0.003 | 0.032 | 0.032 | 0.189 |
| 131 | 0.954 | 0.049 | 0.049 | 0.049 | 0.896 | 0.124 | 0.124 | 0.013 | 0.013 | 0.997 | 0.072 | 0.026 | 0.003 | 0.003 | 0.026 |
| 132 | 0.811 | 0.311 | 0     | 0     | 0.544 | 0.138 | 0     | 0     | 0     | 0.896 | 0.419 | 0     | 0     | 0     | 0     |
| 133 | 0.954 | 0.049 | 0.049 | 0.049 | 0.696 | 0.019 | 0.019 | 0.019 | 0.019 | 0.997 | 0.072 | 0.026 | 0.026 | 0.026 | 0.026 |
| 134 | 0.954 | 0.049 | 0.049 | 0.049 | 0.811 | 0     | 0.062 | 0     | 0.062 | 0.989 | 0.106 | 0.046 | 0     | 0     | 0.046 |
| 135 | 1     | 0     | 0.189 | 0     | 0.954 | 0.049 | 0.205 | 0     | 0     | 0.954 | 0     | 0.112 | 0.012 | 0.012 | 0     |
| 136 | 0.811 | 0.311 | 0     | 0.311 | 0.811 | 0.311 | 0     | 0     | 0.311 | 0.896 | 0.419 | 0     | 0.013 | 0.013 | 0.124 |
| 137 | 0.954 | 0.049 | 0.049 | 0.049 | 0.896 | 0.124 | 0.013 | 0.013 | 0.124 | 0.997 | 0.072 | 0.026 | 0.003 | 0.003 | 0.026 |
| 138 | 0.954 | 0.049 | 0.049 | 0.549 | 0.954 | 0.049 | 0.049 | 0     | 0.549 | 0.989 | 0.106 | 0.046 | 0.011 | 0.011 | 0.311 |
| 139 | 1     | 0     | 0.189 | 0.189 | 1     | 0     | 0.189 | 0     | 0.189 | 0.954 | 0     | 0.112 | 0     | 0     | 0.112 |
| 140 | 0.954 | 0.549 | 0.049 | 0.049 | 0.954 | 0.549 | 0.049 | 0     | 0.049 | 0.997 | 0.829 | 0.003 | 0.003 | 0.003 | 0.003 |
| 141 | 1     | 0.189 | 0.189 | 0     | 0.989 | 0.311 | 0.106 | 0.011 | 0.011 | 0.954 | 0.338 | 0.049 | 0.049 | 0.012 | 0.012 |
| 142 | 1     | 0.189 | 0.189 | 0.189 | 1     | 0.189 | 0.189 | 0     | 0.189 | 0.974 | 0.4   | 0.075 | 0.003 | 0.003 | 0.075 |
| 143 | 0.954 | 0.049 | 0.549 | 0.049 | 0.954 | 0.049 | 0.549 | 0     | 0.049 | 0.758 | 0.112 | 0.264 | 0.038 | 0.004 | 0.004 |
| 144 | 0.811 | 0     | 0.311 | 0     | 0.696 | 0.019 | 0.219 | 0.019 | 0.019 | 0.857 | 0.003 | 0.189 | 0.032 | 0.032 | 0.003 |
| 145 | 0.954 | 0.049 | 0.049 | 0.049 | 0.896 | 0.124 | 0.013 | 0.013 | 0.124 | 0.997 | 0.072 | 0.026 | 0.003 | 0.003 | 0.026 |
| 146 | 0.954 | 0.049 | 0.049 | 0.049 | 0.989 | 0.311 | 0.011 | 0.011 | 0.011 | 0.974 | 0.255 | 0.026 | 0.026 | 0.026 | 0.026 |
| 147 | 1     | 0.189 | 0     | 0     | 1     | 0.456 | 0     | 0     | 0     | 0.954 | 0.338 | 0.012 | 0.012 | 0.012 | 0.012 |
| 148 | 0.954 | 0.049 | 0.049 | 0.049 | 0.811 | 0     | 0.062 | 0     | 0.062 | 0.989 | 0.106 | 0.046 | 0     | 0     | 0.046 |
| 149 | 1     | 0     | 0     | 0.189 | 0.954 | 0.049 | 0     | 0     | 0.205 | 0.954 | 0     | 0     | 0.012 | 0.012 | 0.112 |
| 150 | 1     | 0     | 0     | 0     | 1     | 0.189 | 0     | 0     | 0     | 0.811 | 0.062 | 0     | 0     | 0     | 0     |
| 151 | 0.954 | 0.049 | 0.049 | 0.049 | 0.989 | 0.311 | 0.011 | 0.011 | 0.011 | 0.758 | 0.112 | 0.004 | 0.004 | 0.004 | 0.004 |
| 152 | 0.954 | 0.049 | 0.049 | 0.049 | 0.954 | 0.049 | 0.049 | 0     | 0.049 | 0.989 | 0.106 | 0.046 | 0.011 | 0.011 | 0.011 |
| 153 | 1     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0.954 | 0     | 0     | 0     | 0     | 0     |
| 154 | 1     | 0     | 0     | 0.189 | 0.954 | 0.049 | 0     | 0     | 0.205 | 0.811 | 0.062 | 0     | 0.015 | 0.015 | 0.148 |
| 155 | 0.954 | 0.049 | 0.049 | 0.049 | 0.896 | 0.124 | 0.013 | 0.013 | 0.124 | 0.758 | 0.112 | 0.004 | 0.004 | 0.004 | 0.112 |
| 156 | 1     | 0.189 | 0     | 0     | 1     | 0.189 | 0     | 0     | 0     | 0.974 | 0.4   | 0.026 | 0.003 | 0.003 | 0.026 |
| 157 | 0.954 | 0.049 | 0.049 | 0.049 | 0.954 | 0.049 | 0.049 | 0     | 0.049 | 0.758 | 0.112 | 0.004 | 0.038 | 0.004 | 0.112 |
| 158 | 0.954 | 0.049 | 0.049 | 0.049 | 0.811 | 0     | 0.062 | 0     | 0.062 | 0.449 | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 |
| 159 | 0.811 | 0     | 0.311 | 0     | 0.696 | 0.019 | 0.219 | 0.019 | 0.019 | 0.337 | 0     | 0.066 | 0.066 | 0     | 0     |
| 160 | 0.811 | 0     | 0.311 | 0.311 | 0.544 | 0     | 0.138 | 0.138 | 0.138 | 0.896 | 0     | 0.124 | 0.124 | 0.124 | 0.124 |
| 161 | 0.954 | 0.049 | 0.049 | 0.049 | 0.696 | 0.019 | 0.019 | 0.019 | 0.019 | 0.997 | 0.072 | 0.026 | 0.026 | 0.026 | 0.026 |
| 162 | 0.954 | 0.049 | 0.049 | 0.549 | 0.896 | 0.124 | 0.013 | 0.013 | 0.419 | 0.997 | 0.072 | 0.026 | 0.026 | 0.026 | 0.243 |
| 163 | 1     | 0.189 | 0     | 0.189 | 0.989 | 0.311 | 0.011 | 0.011 | 0.106 | 0.954 | 0.338 | 0.012 | 0.012 | 0.049 | 0.049 |
| 164 | 0.954 | 0.049 | 0.049 | 0.049 | 0.696 | 0.019 | 0.019 | 0.019 | 0.019 | 0.997 | 0.072 | 0.026 | 0.026 | 0.026 | 0.026 |
| 165 | 1     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |

|     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 176 | 0.954 | 0.049 | 0.549 | 0.049 | 0.896 | 0.124 | 0.419 | 0.013 | 0.013 | 0.997 | 0.072 | 0.243 | 0.072 | 0.072 | 0.026 |
| 177 | 1     | 0.189 | 0.189 | 0     | 0.989 | 0.311 | 0.106 | 0.011 | 0.011 | 0.954 | 0.338 | 0.049 | 0.049 | 0.012 | 0.012 |
| 178 | 1     | 0.189 | 0.189 | 0.189 | 1     | 0.456 | 0.046 | 0.046 | 0.046 | 0.954 | 0.338 | 0.049 | 0.049 | 0.049 | 0.049 |
| 179 | 0.954 | 0.549 | 0.049 | 0.049 | 0.989 | 0.717 | 0.011 | 0.011 | 0.011 | 0.928 | 0.48  | 0.028 | 0.028 | 0.028 | 0.028 |
| 180 | 1     | 0     | 0.189 | 0     | 0.954 | 0.049 | 0.205 | 0     | 0     | 0.954 | 0     | 0.112 | 0.012 | 0.012 | 0     |
| 181 | 0.954 | 0.049 | 0.049 | 0.049 | 1     | 0.189 | 0.046 | 0     | 0.046 | 0.758 | 0.112 | 0.004 | 0.004 | 0.004 | 0.004 |
| 182 | 0.954 | 0.049 | 0.049 | 0.049 | 0.989 | 0.311 | 0.011 | 0.011 | 0.011 | 0.758 | 0.112 | 0.004 | 0.004 | 0.004 | 0.004 |
| 183 | 0.811 | 0.311 | 0     | 0     | 0.954 | 0.549 | 0     | 0     | 0     | 0.696 | 0.219 | 0     | 0     | 0     | 0     |
| 184 | 1     | 0     | 0.189 | 0.189 | 1     | 0     | 0.189 | 0     | 0.189 | 0.954 | 0     | 0.112 | 0.049 | 0.049 | 0.112 |
| 185 | 0.954 | 0.049 | 0.049 | 0.049 | 0.954 | 0.049 | 0.049 | 0     | 0.049 | 0.758 | 0.112 | 0.004 | 0.038 | 0.004 | 0.112 |
| 186 | 0.954 | 0.049 | 0.049 | 0.549 | 0.896 | 0.124 | 0.013 | 0.013 | 0.419 | 0.758 | 0.112 | 0.004 | 0.038 | 0.038 | 0.264 |
| 187 | 0.811 | 0.311 | 0     | 0.311 | 0.811 | 0.311 | 0     | 0     | 0.311 | 0.696 | 0.219 | 0     | 0.019 | 0.019 | 0.219 |
| 188 | 0.954 | 0.049 | 0.049 | 0.049 | 0.954 | 0.049 | 0.049 | 0     | 0.049 | 0.758 | 0.112 | 0.112 | 0.004 | 0.038 | 0.004 |
| 189 | 0.811 | 0     | 0     | 0     | 0.811 | 0     | 0     | 0     | 0     | 0.337 | 0     | 0     | 0     | 0     | 0     |
| 190 | 0.811 | 0     | 0     | 0.311 | 0.696 | 0.019 | 0.019 | 0.019 | 0.219 | 0.337 | 0     | 0     | 0     | 0.066 | 0.066 |
| 191 | 0.544 | 0.138 | 0.138 | 0.138 | 0.544 | 0.138 | 0.138 | 0     | 0.138 | 0.201 | 0.032 | 0.032 | 0.032 | 0.032 | 0.032 |
| 192 | 0.811 | 0.311 | 0.311 | 0     | 0.811 | 0.311 | 0.311 | 0     | 0     | 0.896 | 0.419 | 0.124 | 0.013 | 0.013 | 0     |
| 193 | 0.954 | 0.049 | 0.049 | 0.049 | 0.896 | 0.124 | 0.124 | 0.013 | 0.013 | 0.997 | 0.072 | 0.026 | 0.003 | 0.003 | 0.026 |
| 194 | 0.954 | 0.049 | 0.049 | 0.049 | 0.954 | 0.049 | 0.049 | 0     | 0.049 | 0.989 | 0.106 | 0.011 | 0.011 | 0.011 | 0.046 |
| 195 | 1     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0.954 | 0     | 0     | 0     | 0     | 0     |
| 196 | 0.954 | 0.549 | 0.049 | 0.049 | 0.954 | 0.549 | 0.049 | 0     | 0.049 | 0.997 | 0.829 | 0.003 | 0.003 | 0.003 | 0.003 |
| 197 | 1     | 0.189 | 0     | 0.189 | 0.989 | 0.311 | 0.011 | 0.011 | 0.106 | 0.954 | 0.338 | 0.012 | 0.012 | 0.049 | 0.049 |
| 198 | 1     | 0.189 | 0     | 0     | 1     | 0.189 | 0     | 0     | 0     | 0.974 | 0.4   | 0.026 | 0.003 | 0.003 | 0.026 |
| 199 | 0.954 | 0.049 | 0.049 | 0.049 | 0.954 | 0.049 | 0.049 | 0     | 0.049 | 0.758 | 0.112 | 0.112 | 0.004 | 0.038 | 0.004 |
| 200 | 0.954 | 0.549 | 0.049 | 0.049 | 0.989 | 0.717 | 0.011 | 0.011 | 0.011 | 0.997 | 0.829 | 0.003 | 0.003 | 0.003 | 0.003 |
| 201 | 1     | 0.189 | 0     | 0     | 1     | 0.456 | 0     | 0     | 0     | 0.954 | 0.338 | 0.012 | 0.012 | 0.012 | 0.012 |
| 202 | 1     | 0.189 | 0     | 0.189 | 0.989 | 0.311 | 0.011 | 0.011 | 0.106 | 0.974 | 0.4   | 0.026 | 0.003 | 0.003 | 0.075 |
| 203 | 0.954 | 0.049 | 0.049 | 0.049 | 0.896 | 0.124 | 0.124 | 0.013 | 0.013 | 0.758 | 0.112 | 0.112 | 0.004 | 0.004 | 0.004 |
| 204 | 1     | 1     | 0     | 0     | 1     | 1     | 0     | 0     | 0     | 1     | 1     | 0     | 0     | 0     | 0     |
| 205 | 0.954 | 0.549 | 0.049 | 0.049 | 0.989 | 0.717 | 0.011 | 0.011 | 0.011 | 0.928 | 0.48  | 0.028 | 0.028 | 0.028 | 0.028 |
| 206 | 0.954 | 0.549 | 0.049 | 0.049 | 0.954 | 0.549 | 0.049 | 0     | 0.049 | 0.954 | 0.549 | 0.049 | 0     | 0     | 0.049 |
| 207 | 0.811 | 0.311 | 0.311 | 0     | 0.811 | 0.311 | 0.311 | 0     | 0     | 0.696 | 0.219 | 0.219 | 0.019 | 0.019 | 0     |
| 208 | 0.954 | 0.049 | 0.549 | 0.049 | 0.954 | 0.049 | 0.549 | 0     | 0.049 | 0.989 | 0.106 | 0.311 | 0.011 | 0.011 | 0.046 |
| 209 | 1     | 0     | 0.189 | 0.189 | 1     | 0     | 0.189 | 0     | 0.189 | 0.954 | 0     | 0.112 | 0     | 0     | 0.112 |
| 210 | 1     | 0     | 0.189 | 0     | 0.954 | 0.049 | 0.205 | 0     | 0     | 0.811 | 0.062 | 0.148 | 0.015 | 0.015 | 0     |
| 211 | 0.954 | 0.049 | 0.049 | 0.049 | 0.896 | 0.124 | 0.124 | 0.013 | 0.013 | 0.758 | 0.112 | 0.112 | 0.004 | 0.004 | 0.004 |
| 212 | 1     | 0.189 | 0.189 | 0.189 | 1     | 0.189 | 0.189 | 0     | 0.189 | 0.974 | 0.4   | 0.075 | 0.003 | 0.003 | 0.075 |
| 213 | 0.954 | 0.049 | 0.049 | 0.549 | 0.954 | 0.049 | 0.049 | 0     | 0.549 | 0.758 | 0.112 | 0.004 | 0.004 | 0.038 | 0.264 |
| 214 | 0.954 | 0.049 | 0.049 | 0.049 | 0.811 | 0     | 0.062 | 0     | 0.062 | 0.449 | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 |
| 215 | 0.811 | 0     | 0     | 0.311 | 0.696 | 0.019 | 0.019 | 0.019 | 0.219 | 0.337 | 0     | 0     | 0     | 0.066 | 0.066 |
| 216 | 1     | 0.189 | 0.189 | 0     | 0.989 | 0.311 | 0.106 | 0.011 | 0.011 | 0.974 | 0.4   | 0.075 | 0.003 | 0.003 | 0.026 |
| 217 | 0.954 | 0.049 | 0.049 | 0.049 | 0.896 | 0.124 | 0.013 | 0.013 | 0.124 | 0.758 | 0.112 | 0.004 | 0.004 | 0.004 | 0.112 |
| 218 | 0.954 | 0.049 | 0.049 | 0.049 | 0.696 | 0.019 | 0.019 | 0.019 | 0.019 | 0.449 | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 |
| 219 | 0.811 | 0     | 0     | 0     | 0.544 | 0     | 0     | 0     | 0     | 0.337 | 0     | 0     | 0     | 0     | 0     |
| 220 | 0.954 | 0.549 | 0.049 | 0.049 | 0.954 | 0.549 | 0.049 | 0     | 0.049 | 0.954 | 0.549 | 0.049 | 0     | 0     | 0.049 |
| 221 | 0.811 | 0.311 | 0     | 0.311 | 0.811 | 0.311 | 0     | 0     | 0.311 | 0.696 | 0.219 | 0     | 0.019 | 0.019 | 0.219 |
| 222 | 0.811 | 0.311 | 0     | 0     | 0.544 | 0.138 | 0     | 0     | 0     | 0.337 | 0.066 | 0     | 0     | 0     | 0     |
| 223 | 0.544 | 0.138 | 0.138 | 0.138 | 0.337 | 0.066 | 0.066 | 0.066 | 0.066 | 0.201 | 0.032 | 0.032 | 0.032 | 0.032 | 0.032 |
| 224 | 0.954 | 0.049 | 0.549 | 0.049 | 0.896 | 0.124 | 0.419 | 0.013 | 0.013 | 0.997 | 0.072 | 0.243 | 0.072 | 0.072 | 0.026 |
| 225 | 1     | 0     | 0.189 | 0     | 0.954 | 0.049 | 0.205 | 0     | 0     | 0.954 | 0     | 0.112 | 0.012 | 0.012 | 0     |
| 226 | 1     | 0     | 0.189 | 0.189 | 1     | 0     | 0.189 | 0     | 0.189 | 0.954 | 0     | 0.112 | 0.049 | 0.049 | 0.112 |
| 227 | 0.954 | 0.049 | 0.049 | 0.049 | 0.954 | 0.049 | 0.049 | 0     | 0.049 | 0.758 | 0.112 | 0.112 | 0.004 | 0.038 | 0.004 |
| 228 | 1     | 0.189 | 0.189 | 0     | 0.989 | 0.311 | 0.106 | 0.011 | 0.011 | 0.954 | 0.338 | 0.049 | 0.049 | 0.012 | 0.012 |
| 229 | 0.954 | 0.049 | 0.049 | 0.049 | 1     | 0.189 | 0.046 | 0     | 0.046 | 0.758 | 0.112 | 0.004 | 0.004 | 0.004 | 0.004 |
| 230 | 0.954 | 0.049 | 0.049 | 0.049 | 0.954 | 0.049 | 0.049 | 0     | 0.049 | 0.758 | 0.112 | 0.004 | 0.038 | 0.004 | 0.112 |
| 231 | 0.811 | 0     | 0     | 0     | 0.811 | 0     | 0     | 0     | 0     | 0.337 | 0     | 0     | 0     | 0     | 0     |
| 232 | 1     | 0.189 | 0.189 | 0.189 | 1     | 0.456 | 0.046 | 0.046 | 0.046 | 0.954 | 0.338 | 0.049 | 0.049 | 0.049 | 0.049 |
| 233 | 0.954 | 0.049 | 0.049 | 0.049 | 0.989 | 0.311 | 0.011 | 0.011 | 0.011 | 0.758 | 0.112 | 0.004 | 0.004 | 0.004 | 0.004 |
| 234 | 0.954 | 0.049 | 0.049 | 0.549 | 0.896 | 0.124 | 0.013 | 0.013 | 0.419 | 0.758 | 0.112 | 0.004 | 0.038 | 0.038 | 0.264 |
| 235 | 0.811 | 0     | 0     | 0.311 | 0.696 | 0.019 | 0.019 | 0.019 | 0.219 | 0.337 | 0     | 0     | 0     | 0.066 | 0.066 |
| 236 | 0.954 | 0.549 | 0.049 | 0.049 | 0.989 | 0.717 | 0.011 | 0.011 | 0.011 | 0.928 | 0.48  | 0.028 | 0.028 | 0.028 | 0.028 |
| 237 | 0.811 | 0.311 | 0     | 0     | 0.954 | 0.549 | 0     | 0     | 0     | 0.696 | 0.219 | 0     | 0     | 0     | 0     |
| 238 | 0.811 | 0.311 | 0     | 0.311 | 0.811 | 0.311 | 0     | 0     | 0.311 | 0.696 | 0.219 | 0     | 0.019 | 0.019 | 0.219 |
| 239 | 0.544 | 0.138 | 0.138 | 0.138 | 0.544 | 0.138 | 0.138 | 0     | 0.138 | 0.201 | 0.032 | 0.032 | 0.032 | 0.032 | 0.032 |
| 240 | 1     | 0     | 1     | 0     | 1     | 0     | 1     | 0     | 0     | 0.954 | 0     | 0.549 | 0.049 | 0.049 | 0     |
| 241 | 0.954 | 0.049 | 0.549 | 0.049 | 0.954 | 0.049 | 0.549 | 0     | 0.049 | 0.758 | 0.112 | 0.264 | 0.038 | 0.004 | 0.004 |
| 242 | 0.954 | 0.049 | 0.549 | 0.049 | 0.896 | 0.124 | 0.419 | 0.013 | 0.013 | 0.758 | 0.112 | 0.264 | 0.038 | 0.038 | 0.004 |
| 243 | 0.811 | 0.311 | 0.311 | 0     | 0.811 | 0.311 | 0.311 | 0     | 0     | 0.696 | 0.219 | 0.219 | 0.019 | 0.019 | 0     |
| 244 | 0.954 | 0.049 | 0.549 | 0.049 | 0.954 | 0.049 | 0.549 | 0     | 0.049 | 0.758 | 0.112 | 0.264 | 0.038 | 0.004 | 0.004 |
| 245 | 0.811 | 0     | 0.311 | 0.311 | 0.811 | 0     | 0.311 | 0     | 0.311 | 0.337 | 0     | 0.066 | 0.066 | 0.066 | 0.066 |
| 246 | 0.811 | 0     | 0.311 | 0     | 0.696 | 0.019 | 0.219 | 0.019 | 0.019 | 0.337 | 0     | 0.066 | 0.066 | 0     | 0     |
| 247 | 0.544 | 0.138 | 0.138 | 0.138 | 0.544 | 0.138 | 0.138 | 0     | 0.138 | 0.201 | 0.032 | 0.032 | 0.032 | 0.032 | 0.032 |
| 248 | 0.954 | 0.049 | 0.549 | 0.049 | 0.896 | 0.124 | 0.419 | 0.013 | 0.013 | 0.758 | 0.112 | 0.264 | 0.038 | 0.038 | 0.004 |
| 249 | 0.811 | 0     | 0.311 | 0     | 0.696 | 0.019 | 0.219 | 0.019 | 0.019 | 0.337 | 0     | 0.066 | 0.066 | 0     | 0     |
| 250 | 0.811 | 0     | 0.311 | 0.311 | 0.544 | 0     | 0.138 | 0.138 | 0.138 | 0.337 | 0     | 0.066 | 0.066 | 0.066 | 0.066 |
| 251 | 0.544 | 0.138 | 0.138 | 0.138 | 0.337 | 0.066 | 0.066 | 0.066 | 0.066 | 0.201 | 0.032 | 0.032 | 0.032 | 0.032 | 0.032 |
| 252 | 0.811 | 0.311 | 0.311 | 0     | 0.811 | 0.311 | 0.311 | 0     | 0     | 0.696 | 0.219 | 0.219 | 0.019 | 0.019 | 0     |
| 253 | 0.544 | 0.138 | 0.138 | 0.138 | 0.544 | 0.138 | 0.138 | 0     | 0.138 | 0.201 | 0.032 | 0.032 | 0.032 | 0.032 | 0.032 |
| 254 | 0.544 | 0.138 | 0.138 | 0.138 | 0.337 | 0.066 | 0.066 | 0.066 | 0.066 | 0.201 | 0.032 | 0.032 | 0.032 | 0.032 | 0.032 |
| 255 | -0    | 0     | 0     | 0     | -0    | 0     | 0     | 0     | 0     | -0    | 0     | 0     | 0     | 0     | 0     |