B.Sc. Dissertation

# Dynamic Effects of Robotic Process Automation on Processes and Routines in Organisations

By

Wei Yifei

Department of Information System and Analytics

School of Computing

National University of Singapore

2022/2023

B.Sc. Dissertation

# Dynamic Effects of Robotic Process Automation on Processes and Routines in Organisations

By

Wei Yifei

Department of Information System and Analytics

School of Computing

National University of Singapore

2022/2023

Project No: H175490
Supervisor: Prof Hahn Jungpil

Deliverables:
Report: 1 Volume

Abstract

The deployment of Robotic Process Automation (RPA) has grown increasingly popular in industry and organisations whose ambition is to enhance organisational operating efficiency embrace RPA for automating the standard business processes. Efficiency gain as the most outstanding benefit is always highlighted but unintended consequences of RPA implementation, such as reduced process flexibility and trade-off against potential organisation transformation, are overlooked and less discussed. This paper studies the dynamic effects of RPA on routine evolution over time using a network simulation model. The research study reveals the impacts of RPA on organisational processes in terms of process complexity, structural change in process, probability of phase change and efficiency gain by exploring and interpreting important RPA deployment features, including sequence length and frequency threshold qualifying suitable sequences of actions to automate through RPA. With that, we can understand how the process of organisational routines adaptation is impacted by RPA and highlight new insights about the trade-off between efficiency gain and potential digital transformation in organisations. Also, this research provides well-rounded answers on whether and how organisations could implement RPA to achieve higher operational efficiency while maintaining some degree of process flexibility and preserving the potential of process adaptation and digital innovation.

Subject Descriptors:

D.4.1 Process Management
H.4.1 Office Automation
I.2.9 Robotics
I.6.5 Model Development
K.4.3 Organisational Impacts

Key Words:

Robotic process automation, process and routine, automation, organisational routines

Implementation Software and Hardware:

Python 3.9, NUS School of Computing Compute Cluster

## Acknowledgement

I would like to express my sincere gratitude to all those who have contributed throughout my journey of completing my final year research project.

First, I would like to thank my supervisor Professor Hahn Jungpil and my evaluator Dr. Tan Wee Kek for providing invaluable guidance and support throughout this project. Their expertise and insight were instrumental in shaping the direction of this research project and ensuring the quality of research output.

I would also like to extend my appreciation to Zhou Junjie, who generously gave his time and shared valuable feedback with me to fine-tune the details in the simulation model of my research project. Additionally, I would like to thank the members of Garbage Can Lab, Chen Dawei, Li Yonggang and Li Junyi who offered their assistance, encouragement and brilliant suggestions. Besides, I would like to show appreciation to my peers and friends who supported me in this journey of final year project.

Finally, I would like to acknowledge the resources that were made available to me by the university including the NUS School of Computing Compute Cluster as well as the access to multiple research papers. The support from NUS School of Computing was essential in enabling me to carry out this research project.

Table of Contents

# 1 Introduction

## 1.1 Background and Research Motivation

Robotic process automation (RPA) has gained popularity in industry in recent years. RPA leverages software robots to automate the processing of repetitive and standard tasks and thus reduce unnecessary manual work. As a type of digital technology, RPA enhances efficiency in organisational operations by automating and simplifying those business processes qualified for RPA.

The efficiency gain as a result of RPA application creates significant enthusiasm towards implementing RPA for automation in organisations, which gives rise to significant uptake in practice. However, the benefits of RPA have been overemphasised and there is not much discussion on the possible negative consequences of RPA. For example, RPA, which relies on the designed robot to execute processes autonomously, may reduce process flexibility and thus makes it difficult to modify and adapt to meet new demands or change requests (Syed et al., 2020). As a result, implementing RPA for automation may hinder an organisation's ability to be adaptive. Further, process complexity, structural change in process as well as the potential for digital innovation and organisational process transformation are relevant and important factors to organisations, which can be impacted by RPA implementation and lead to unintended outcomes. Therefore, there is a need to understand how the process of organisational routines adaptation is impacted by RPA and study how the important design features of RPA intervention, such as memory, sequence occurrence frequency and length, might impact the above-mentioned factors. The findings will give a better understanding and provide well-rounded answers on whether an organisation should adopt RPA and if so how they might want to implement it if they want to ensure that the process of organisational routines adaptation is well managed and healthy.

Studying the impacts of RPA from the angle of routine evolution enables organisations to have a deeper understanding of the trade-off between potential gains and any possible negative side effects of applying RPA. Further, routine evolution takes place over time and the impacts on organisations in terms of process complexity, structural change in process, the likelihood of digital transformation and efficiency gain vary at different stages of routine evolution under the influence of external disturbance from RPA. The dynamic concept highlights the change in the effects that are spread over time, which provides a more nuanced understanding of how a process responds to changes and how routines evolve in both short and long term. Thus, it is important to study the dynamic impacts of RPA on routine evolution in organisations so that we can obtain a full picture of how routine evolution impacts organisations at different timestamps. With that, organisations can also know whether they should implement RPA for automation at various stages of the organisational life cycle and how to implement RPA to maximise the intended benefits at the current phase.

## 1.2 Robotic Process Automation (RPA)

RPA is a technology for automation of business processes, which relies on software robots running algorithms that are created with a set of rules and instructions on a physical or virtual machine. RPA mimics human-computer interactions (Wellmann, Stierle, Dunzer & Matzner, 2020), utilises a range of software as tools and leverages various platforms to simplify and automate business processes (Van der Aalst, Bichler & Heinzl, 2018), especially those frequent and repetitive computer-based processes which require much manual effort in the workplace. RPA is implemented to automate mundane tasks or time-consuming manual processes, such as copy-paste actions, file transfer, document upload, etc (Aguirre & Rodriguez, 2017), which enhances efficiency in organisations and reduces human error in executing tasks. Common RPA tools include Automation Anywhere, UiPath, SS&C Blue Prism, etc.

Below is an example of RPA applied to automate mundane processes in the banking industry to reduce manual effort and human error. When a customer opens a new account with a bank, the personal particulars of the account holder will be collected by the banker and complimentary documents will be required for submission to validate his financial status and personal information. His data and copies of those supporting documents will then be uploaded and stored in the client database. Without RPA, these processes have to be manually carried out by the bankers who have to type in the information in the Excel file and upload the documents to the bank’s internal portal. With RPA, the entire process can be as simple as clicking the start button to let the robot run. The robot will copy and paste the required information, consolidate the complimentary documents and finally upload and save the data in the database.

RPA can carry out tasks error-free at a high volume and fast speed with minimal human intervention (Wellmann et al., 2020). RPA promises to boost efficiency gains in organisations as it reduces human attention when the processes are executed by robots. There are a few important deployment features and eligibility criteria of RPA, namely sequence length and sequence frequency threshold, which significantly determine whether adopting RPA for business automation can achieve the expected efficiency gain (Wellmann et al., 2020). More specifically, sequence length refers to the number of actions or steps in a sequence that will be automated via RPA and sequence frequency threshold is a cutoff frequency value used to determine the minimum number of occurrences required for a particular sequence of actions to be automated with RPA. These two factors are important RPA deployment characteristics whose impacts on routine evolution will be studied later using the simulation model.

### 1.3 Processes and Routines under Digitalisation

Process and routines are essential in facilitating operations and accomplishing tasks in organisations (Howard-Grenville & Rerup, 2017). Processes refer to those standardised and well-defined operating procedures which organise the tasks to be completed in order to achieve the goal whereas routines are how work usually gets done in reality, especially repetitive and recognizable patterns of interdependent organisational actions (Feldman & Pentland, 2003). Also, processes typically don't organically change unless the management decides to update them and prepare for the change of goals or standard procedures. However, routines are variations in how work actually gets done, which can organically evolve over time and respond to the changing environment (Brown & Duguid, 2000).

A process can drift, which means to undergo incremental change over time (Bose, van der Aalst, Žliobaitë & Pechenizkiy, 2011; Ciborra et al. 2000). When a process drifts, it learns, responds and adapts (Bickhard and Campbell, 2003; Pentland, Feldman, Becker & Liu, 2012), which can be favourable or harmful depending on the circumstances. Drift changes the way a process is performed, thus affecting the outcome of routine evolution over time. The beneficial outcome is that routine drift and evolution inspire more new paths, lead to drastic structural change in the process and finally settle into a stable state with a new dominant path established (Bak, 1996; Frigg, 2003). This is also known as process self-organisation, which means that a process can evolve as it undergoes constant change and ongoing operation without intervention from external agents (Feldman & Pentland, 2003; Foster, 1997).

However, drift can also lead to disastrous and unexpected consequences, such as drift into complexity catastrophe.

Intended as a mechanism to improve coordination and control (Berente, Lyytinen, Yoo & King, 2016), digital technologies (such as RAP) are often used as tools to improve process execution and business operations. Under the influence of digital technology, the drift behaviour of a process will be impacted significantly as digitisation constrains the way a process is performed (Pentland, Liu, Kremser & Haerem, 2020). Digital technologies possess material properties, such as memorizability which is used to retain and store information (Yoo, 2010) and it can be programmed to guide users on the next action based on past experience and decisions. Besides, the programmability of digital technology will eliminate all variation from the process involved. In the context of RPA, those segments of a process where automation is implemented will leave no room for variation or new workarounds as those sub-parts of a process become automated and thus deterministic. As a result, the structural change and the establishment of new dominant paths can be altered when digital transformation technologies, such as RPA, are implemented, which potentially affects the likelihood of undergoing digital innovation and organisation transformation. Thus, it is meaningful to study how digital technology, such as RPA, interacts with process drift and routine evolution.

Processes and routines are crucial and fundamental factors in organisations to perform and accomplish daily work. RPA, as a type of digital technology widely used in organisations, can change and disrupt how processes evolve. RPA can also reshape how organisations perform processes to deliver business value efficiently to their clients.

# 2 Project Objectives

## 2.1 Project Objectives

The primary objective of this study is to investigate the effects of RPA deployment on business processes in organisations. We seek to explore and identify the impacts of RPA on process evolution over time in terms of process complexity, the magnitude of accumulated change, the probability of transformative phase changes in process structure and efficiency gains. Using a network simulation model, we study how these effects change when RPA deployment characteristics vary, including the length of RPA sequence of actions and the occurrence frequency threshold set for selecting suitable candidates of sequences of actions for RPA. This study seeks to generate some insights about the trade-off between the business value of automation tool applications in terms of efficiency gain and potential digital transformation in organisations. Our results culminate in a framework on how organisations should leverage RPA to not only boost operational efficiency but also allow for healthy and effective digital transformation in organisations.

### 2.2 Research Questions

Research Question 1: How does introducing RPA influence process evolution over time in terms of process complexity, the probability of transformative phase changes in process structure, the magnitude of accumulated change and efficiency gain?

Research Question 2: How are those effects, such as process complexity, drift probability, the magnitude of change and efficiency gain, affected by the RPA deployment features which include the length of the sequence of actions that is automated via RPA and the occurrence frequency threshold used to qualify a sequence of actions for RPA implementation?

## 3 Literature Review

Pentland et al. (2020) developed a theory about how drift influences digitised processes due to the generative nature of digital technologies by analysing the impacts of incremental process changes on process complexity and drastic phase changes. He developed a simulation model to identify the emergence or dissolution of paths and observe the structural changes in the network. Based on simulation results, it is concluded that digital technology can afford variations and the creation of new paths, thus leading to bursts in process complexity and affecting the likelihood and magnitude of transformative endogenous phase changes to different extents at different levels of the lexicon of actions, reinforcement, and modularity. In their paper, the phenomena of the sudden bursts in complexity and dramatic structural change in the network were interpreted as either positive impacts: opportunities for adaptation and enhanced efficiency, or negative impacts: complexity disaster and dysfunctionality. Further, they highlighted that in real-world contexts, adaptive systems attempt and generate new paths while considering the historical processes and knowledge, finally leading to self-reinforcing complexity growth and even phase change. Their paper contributes to the baseline simulation model in my research and the phenomena observed in their paper lays the foundation for my study on the dynamic impacts of RPA, as a form of automation, on digitised processes. Different from the generative nature of digitalisation affording more variations in Pentland et al.'s paper, the deterministic nature of RPA in my study restricts variations to some extent but enables higher efficiency to be achieved. The dependent variables in his simulation model are also selected in my study to set a comparison benchmark between standard but evolutionary processes and RPA-enabled processes. The impacts of digitalisation on organisations may vary depending on the nature of the technology that is being introduced into processes, not always being generative due to high variation in nature. Thus, my study expands the scope of the study from process drift and transformation further to the influence of RPA, specifically a type of automation technology focusing on efficiency boost. I analyse how those RPA deployment features affect process complexity and routine evolution, thus giving greater efficiency in organisations.

Wellmann et al. (2020) developed a framework for categorising and evaluating the potential candidates for RPA in business processes. His paper provided several guidelines for RPA practitioners to evaluate whether a process is worth the RPA implementation effort. He identified thirteen important process activity characteristics of RPA which are further categorised into five main perspectives as evaluation criteria, namely task, time, data, system and human. Some important criteria include frequency, the number of execution steps, the number of variations to execution flow in the business, the number of unsuccessful terminations in tasks, etc. These characteristics highlight the distinguishing features of RPA as a type of digital and automation technology. Wellmann et al. also tested and verified their framework in a practical setting by analysing the event log of a publicly available dataset related to a P2P process of a multinational coating and paint enterprise. They concluded that the framework provides effective insights into identifying the proper candidates for RPA. Wellmann et al.'s framework identifies the special characteristics of RPA, which contributes to the blueprint of the RPA simulation model in my study. I included important RPA implementation features, such as length, frequency and variation, in the simulated network to mimic the actual RPA process. However, one limitation of their paper is that the framework was only evaluated based on one dataset, and not all the criteria in his proposed framework were tested for generalisability. Also, their paper only focuses on how to qualify RPA process candidates and the immediate impacts of RPA which is efficiency gain but neglects how implementing RPA might impact organisational adaptability in the long run. My study further analyses the impact of RPA on routine evolution in a holistic and dynamic manner with greater robustness.

# 4 Research Design

## 4.1 Overview of the Simulation Model

In this paper, an evolving network model is used to simulate business process change over time to develop a theory about the dynamic effects of RPA on processes and routine evolution in organisations. Simulation models create an environment where the effects of digital intervention can be clearly demonstrated via quantitative metrics. We are thus able to visualise the interaction of mechanisms under the influence of a range of variables of interest over many process interactions (Davis, Eisenhardt & Bingham, 2007; Simon 1996). Therefore, in this paper, simulations are used as an effective tool for theory construction.

Since processes naturally evolve over time, the effects of deploying RPA on business process evolution can only be identified by analysing how RPA intervenes and transforms this existing benchmark digitised process evolution found by Pentland et al. (2020). Thus, my simulation model includes important parameters that can be shaped by digitisation and mentioned in Pentland's paper, such as the probability of workarounds and other variations (*V)* (Alter 2014; Ciborra et al. 2000), and the span of the process' memory (*R*) (Argote 2013; Darr, Argote & Epple, 1995; Holan & Phillips 2004). Besides, two more variables representing the unique deployment characteristics of RPA, namely the length of RPA sequence of actions (*L)* and RPA occurrence frequency threshold (*F*), are also included in the simulation model to further study the impacts of RPA.

A weighted directed network graph is used to model the sequence of actions in a business process, which represents a series of actions of completing a task in an organisation. The network trajectory changes when new edges form or existing edges are removed over time. The network is observed over many iterations by varying the values of the parameters to identify the changes in average process complexity, the magnitude of change in the network structure, the probability of transformative phase change in the process and efficiency gain.

4.2 Model and Important Mechanisms

**Task Network**

The Task network represents the possible ways of getting a task or business process done in the organisation. Networks of sequentially related actions which are connected by edges are modelled in a weighted directed graph with a source and sink (Pentland et al. 2012; Pentland, Recker & Wyner, 2015). A node represents an action or event in the process (Anderson and Robey 2017; Strong et al. 2014), and an edge with a specified direction represents a sequential relationship or a path from one action to the next action (Pentland et al., 2020). The weight on an edge refers to the transition probability from one action to the next possible action. The source represents the first step that kicks off a business process, while the sink represents the final action that concludes the business process and generates the outcomes. The source and sink are fixed throughout process evolution because the start and end points of a business process in organisations usually remain the same even when digitalisation and automation are carried out by a blend of digital technologies and people. A path in the network is a list of connected edges and each possible path from source to sink represents one way to carry out a process.

As a process evolves, the network will experience incremental structural change over time as edge formation and dissolution take place, which potentially give rise to surges in network complexity. Edge formation and dissolution as described below are important mechanisms influencing process complexity over time.

**Edge Formation**

Edge formation is driven by variations (Pentland et al. 2020). In a process, the next action happens sequentially and spontaneously once its previous action has been completed, which is represented by the connection established by an existing edge between two nodes in the network. However, as the process evolves, there is some probability of encountering an exception or issue and there is a need for re-arranging, identifying a new next action and jumping from the current node to another node which is different from the one connected by the original edge. For example, in the bank account opening process, if a client would like to update and resubmit the complementary financial documents that he has sent to the banker, there is a need to re-engineer some intermediate steps to make sure that the new documents are stored in the database and all the relevant information is changed accordingly. Variation creates a new workaround (Alter 2014), which is represented by the edge formation mechanism in the network model. Edge formation stimulated the branching out of new edges which give rise to alternative paths in the network.

**Edge Dissolution**

Retention results in edge dissolution. In the long enactment of organisational processes, remembering and forgetting co-exist. There is a limit on how much an organisational process can remember and reinforce the future paths based on its historical information during its evolution over time. Those edges which are usually taken will stay in the process memory, which is also known as retention, and enhance the likelihood of taking the same path again in the future. However, those edges which are visited less frequently in the network will fade over time and finally be forgotten. This results in edge dissolution due to the property of constrained process memory.

**Matrix Representations**

To model the process evolution and development and removal of edges among nodes in the network, three $N$ x $N$ (where $N$ represents the number of nodes) matrices are used in the simulation model. First, a Markov *transition* matrix is used to record the transition probability from one node to another node, which determines how likely the path between two nodes will be taken. Second, an *adjacency* matrix is used to remember which pairs of nodes are connected, which records the network structure and can be used to trace the number of possible paths. Finally, a *history* matrix memorises the number of times each node has been visited which is still remembered by the process, or in other words, within the retention. The history matrix is updated every process iteration by including the new routine and forgetting the sequences of actions when the memory span runs out. The history matrix is also an important input used in updating the transition matrix with transition probabilities between nodes.

**Implementing RPA in the Simulation Model**

RPA is a digital technology which can automate some segments of an organisational process. To stimulate the effects of RPA in the simulation model, an algorithm of identifying the suitable candidates of sequences for RPA and implementing RPA on them is required.

As the process evolves, new edges are formed and historical edges outside of the memory span are dissolved (i.e., removed). Throughout the process iterations, the algorithm first identifies the sequences of actions that occur repetitively with occurrence frequency exceeding the threshold (*F*). As suggested by Wellmann et al. (2020), a process can be qualified for RPA if a series of actions happens frequently, and occurrence frequency is an important criterion of qualifying a process for RPA. Thus, in the simulation model, the sequence of actions appearing with a frequency higher than the occurrence frequency threshold (*F*) will then be automated. Once automated, the process memory and corresponding transition probability of relevant nodes will be updated accordingly. More specifically, when a sequence of actions (e.g., Node 3 → Node 5) has been automated by RPA, this sequence becomes deterministic and it will always be remembered. The transition probability from the first node in the RPA sequence to the next node (e.g., from Node 3 to Node 5) will be fixed at 1.00 for subsequent process iterations. Due to the deterministic nature of RPA, the sequences which have been automated by RPA will never be forgotten throughout the entire process of routine evolution and they are removed from the memory altogether such that those new and less frequent sequences of actions can be retained in the memory and accumulate, which allows RPA to consistently happen in the rest of process iterations.

### 4.3 Independent Variables

The model has multiple parameters as the independent variables in the simulation model, which are variables of research interest. The independent variables include parameters that could affect the behaviour of process evolution and other variables relevant to the characteristics of RPA deployment. We would like to find out how the task network will evolve over time at various levels of these parameters.

#### 4.3.1 Number of Nodes (*N*)

This parameter represents how many nodes there are in the network simulation model and how many actions are involved in a process. Under the context of RPA, it represents how many specific steps are required to complete a task or business process within an organisation. This parameter is fixed throughout the simulation (i.e. all process iterations).

#### 4.3.2 Retention (*R*)

Retention restricts how much the historical process iterations can affect the current process iteration, which defines the extent of the influence of past information on the present. Retention is measured by how many past process iterations are remembered by the existing process, which is implemented as a fixed length of memory window saving historical paths in the simulation model. When the most recent sequence of actions is added to the process memory which exceeds the maximum memory span or retention, the oldest sequence of actions in the retention window is forgotten and removed (Pentland et al., 2020).

In the case of complete automation, which represents the sequence of actions where RPA is deployed, a deterministic programme will fully take over and control the sequence of actions that will take place. In this case, under the effect of RPA, this particular sequence of actions will no longer be added to retention again and history becomes irrelevant to its further evolution. More specially, it means that the sequence automated by RPA will not be affected by the incoming new memory from the subsequent iterations. Thus, the effect of retention on those RPA sequences of actions will be zero. However, on other fragments of a process where RPA has not taken place, it is not a deterministic program at all. Retention plays a role in providing valuable historical information and choices about what has been done in the past, what is remembered from past experiences and what can be done next.

The process of storing important documents digitally in an organisation is used to be frequently performed by manually selecting the required documents, uploading the documents to the company database through an online portal, and finally sending an email to the department manager to inform him about successful upload and documentation. With the help of RPA, the entire process can be automated by following the same sequence of steps but executed with a software robot. This is an example of how the historical sequences of action provide guidance and information based on past experience in process evolution. After automating the process by RPA, as the process undergoes further evolution, those past manual actions will be forgotten over time as the deterministic programme of RPA is more frequently applied. Thus, when we would like to introduce RPA to an evolving business process, where a human agent interacts with a deterministic program, introducing retention in the simulation model is important. Also, if a sequence of actions happens frequently in the retention window, we will have some ideas about selecting potential candidates for RPA because the purpose of RPA is to automate those repetitive sequences of actions. This is why the length of retention is an important factor which may impact the process evolution under the influence of RPA implementation.

When we would like to enable a longer memory span and let the system remember more historical choices or paths, we could increase the value of $R$. However, more memory is not necessarily better because based on Pentland's study, low values of retention are associated with a lower inertia and greater flexibility (Pentland et al. 2012). Adaptive systems need to have not only learning but also forgetting mechanisms, which means that those very historical paths containing outdated information have to be forgotten at some point. Otherwise, the oldest memories may hinder the discovery of new and innovative paths in the process evolution and also affect the decisions on the suitable candidates of sequences for PRA.

4.3.3 Variation ($V$)

Variation represents the probability of the need for workarounds (Alter 2014) which means that some sequential rearrangement in the sequence of actions when the process encounters an issue or an exception (Pentland et al., 2020). When an issue occurs unexpectedly, the simulation model randomly picks the next action from the possible destination node based on the weights of existing edges. $V = 0$ implies that there are no exceptions, and the process stays on track without any variation, which happens at the segments of the process where RPA is implemented. When $V = 0.01$, there is a 1% chance that an issue will arise at each step or transition and the next action is randomly determined.

In the context of RPA, when a sequence of action is fully automated by RPA, the algorithm becomes deterministic and there are no workarounds or variations. Variation has no impact on those sequences of actions that are fully programmed. However, for those actions where RPA is not deployed, variation may take place due to human intervention in the process of innovating and re-engineering the existing business process. In reality, when deploying RPA on a business process, business analysts will intervene and examine the entire process flow to identify and remove unnecessary actions or re-design the sequence of actions to achieve higher efficiency and faster output delivery of the digital system. In this case, some variations or workarounds will be introduced to the business process. For example, in the case that a retail company needs to document the return request from its customer, business analysts find out that keying in the return request and details into an Excel file and then uploading the file across multiple systems and databases are not efficient. Thus, he decided to re-engineer the sequence of actions and enabled the direct change of details on the company portal without manually logging the information into Excel files. From this example, a new exception path is enacted and a new way of getting the same thing done is developed. This is how variation may take place in a business process before automation takes place.

### 4.3.4 Length of RPA Sequence of Actions ($L$)

The length of the RPA sequence of actions is a parameter specific to the characteristics of RPA deployment. In the business context of RPA, it refers to the number of steps or actions to be performed in order to complete the segment of an organisational process that is automated by RPA. In the simulation model, it refers to the number of nodes involved in the sequence of actions which is automated by RPA. For example, when $L = 3$, it implies that only those sequences of actions consisting of three nodes can be automated, such as Node 3 → Node 5 → Node 6 (where $L = 3$) but not Node 3 → Node 4 (where $L = 2$).

In the business context, the length of the RPA sequence of actions determines how complicated the RPA process can be. A longer RPA sequence of actions means that more steps are involved and the RPA robot has to execute more actions sequentially, which reflects the complexity of a process in the organisational setting. By common business practice, RPA is seldom implemented on very long sequences of actions because it is hard to trace back the source of error if the robot fails. Besides, those shorter RPA sequences of actions could be easily qualified for RPA due to the higher likelihood of having a short sequence of actions occur repetitively in an organisational process.

### 4.3.5 Occurrence Frequency Threshold for RPA ($F$)

The occurrence frequency threshold for RPA refers to the occurrence frequency benchmark used to qualify a sequence of actions for automation by RPA. Once the frequency of a sequence of actions occurring exceeds the threshold value $F$, the sequence will be automated by RPA. This is a parameter representing one of the important features of RPA implementation. In the simulation model, frequency threshold is computed as the proportion of the frequency of a sequence out of the entire retention length. It is a proportion value which cannot go beyond 1for RPA to occur. For example, occurrence frequency threshold $F = 0.5$ means that only those sequences that occur more than or equal to 50% times in the retention window will be qualified for automation by RPA. The higher the frequency threshold for RPA, the harder it is for a sequence of tasks to be qualified for RPA.

In the organisational context, only sequences or routines which occur frequently and repetitively will be automated using RPA to boost operational efficiency. Those less frequently occurring sequences of actions are unlikely to be automated because less efficiency gain can be obtained from automating and digitising them.

### 4.4 Dependent Variables

To study the dynamic effects of RPA on processes and routine evolution, we use four measures to quantify the impacts, namely process complexity, the magnitude of change, the probability of phase change and efficiency gain.

#### 4.4.1 Process Complexity

Process complexity represents how complicated a process or task is. A process with higher complexity means that there are more ways of getting the process completed. In the simulation model, complexity is measured by counting the possible simple paths through the network which represents the process (Hærem, Pentland & Miller, 2015), which is the total number of acyclic paths from source to sink in the network. Thus, process complexity is high if a process can be completed via a range of possible ways, while complexity is low if a process can successfully be done by only a few paths.

As we apply a large network model in the simulation, due to the constraint of time and computational power, we adopted the strategy of measuring the complexity of a program by computing the number of linearly independent paths $V(G)$ which is equal to $e - n + p$, where $e$ represents the total number of edges, $n$ represents the number of vertices and $p$ represents the number of connected components in the network (McCabe, 1976). An appropriate scaling factor is added to the equation to generate the complexity index which represents process complexity (Goh & Pentland, 2019).

#### 4.4.2 Probability of Phase Change

The probability of phase change is also known as drift probability. A phase change is identified when there is a sudden burst in process complexity immediately followed by a low complexity after the surge. As the process evolves, the complexity of the simulated process gradually increases and can spike up by several orders of magnitude. Finally, process complexity falls back and reaches a relatively stable and low level. Due to the complexity burst, the network experiences drastic structural change, which may give rise to the birth of new dominant paths and transition to a new state of the network (Pentland et al., 2020). The presence of complexity burst is strong evidence of incremental change resulting in potential phase change or known as process transformation.

When quantifying the probability of phase change, we only consider and count the cases where incremental change ultimately leads to much lower complexity and fewer possible paths after the complexity burst. The probability of phase change is simply the fraction of simulated trajectories that meet these criteria. We would like to observe how the probability of phase change is affected by RPA; in other words, whether RPA promotes or impedes opportunities to develop new dominant paths and undergo a structural transformation during process evolution.

#### 4.4.3 Magnitude of Change

The magnitude of change in a process is measured by comparing the evolving network across a series of process iterations with respect to the original network at the start of process evolution. It indicates how much the network has changed from the original network over time in terms of its structure, which further implies how much the process has undergone structural change over process iterations. In the simulation model, the magnitude of change is computed using the sum of bitwise XOR of two adjacency matrices by identifying the differences in edges and summing them up. The two adjacency matrices are the adjacency matrix of the network at the start of process evolution, which is the simple happy path, and the adjacency matrix of the network at some process iteration later during process evolution.

#### 4.4.4 Efficiency Gain

The concept of finding the shortest path in the network is used to model efficiency in the simulation model. Efficiency means how fast a task or process can be completed and the concept of the shortest path also implies the fastest way of passing through a network from source to sink. Efficiency gain is measured by the reduction in the length of the shortest path from the source to the sink of the final network at the end of process evolution from the original length of the simple happy path. To identify the shortest path in the network, we applied Dijkstra's algorithm which can find the shortest path from the start node to the end node in a directed acyclic graph with equal weights (Javaid, 2013).

Efficiency gain is a very important benefit of RPA, which is also the main reason for automating business processes in reality. Thus, we would like to model efficiency gain in the simulation model to find out what factors affect efficiency and help boost efficiency gain with the implementation of RPA.

## 4.5 Simulation Model Initialisation

The simulation always starts from a straightforward path which means connecting all the nodes from source to sink sequentially based on the index of a node in increasing order e.g. 1, 2, … 99, 100. This is known as a simple happy path. As the process goes on, edges are formed and removed over time and the process drifts.

Below is a table of the initialised values of important parameters in the simulation model. For parameters *R*, *V*, *F*, *L*, three values are carefully picked to represent low, medium and high levels of each parameter. These parameters are treated as categorical variables to draw qualitative findings.

**Table 1. Model Parameters**

| ***Parameters*** | ***Simulated Values*** | ***Explanation*** |
|---|---|---|
| *Iterations* | 5000 | The process will evolve for 5000 process iterations which can be interpreted as time in this model. Each iteration gives a new network at that time stamp, which is also known as one trajectory. |
| *N* | 100 | We use a proper and realistic number of actions (*N* = 100) to represent a business process in an organisation, which will form the fundamental network that we could draw conclusions and develop theory from. |
| *R* | Low: 50<br>Medium: 100<br>High: 150 | The simulated values of *R* are suggested as reasonable values by Pentland et al. (2020). The *R* values in ascending order represent low, medium and high retention, respectively. |

| | | |
|---|---|---|
| *V* | Low: 0.001<br>Medium: 0.005<br>High: 0.01 | Three values of variations *V* are recommended by Pentland et al. (2020). The *V* values in ascending order represent low, medium and high variation, respectively. |
| *L* | Low: 3<br>Medium: 5<br>High: 10 | Three values of *L* are carefully picked to be much lower than 100 because it is usually not realistic to automate a very long sequence of actions in an actual business context. The *L* values in ascending order represent low, medium and high lengths of RPA sequence of actions respectively. |
| *F* | Low: 0.2<br>Medium: 0.5<br>High: 0.9 | We simulate three levels of frequency threshold, which represent low, medium and high levels of *F*. Low frequency threshold means that it is quite easy for a sequence to be qualified for RPA while high frequency threshold means that it is hard for a sequence to be qualified for RPA unless it occupies the majority of the space in the retention window. |

The process evolution for each parameter setting mentioned above will be run 1000 times and averages are computed for analysis to obtain stable results and statistics and draw reliable conclusions.

# 5 Model Implementation and Results

## 5.1 Simulation Results

### 5.1.1 Process Complexity

To understand the effects of RPA on process complexity, we observe the time series of process complexity over 5000 process iterations using line charts and then we further look into how the end-of-evolution process complexity varies with different parameter settings of *R*, *V*, *F*, and *L* under the influence of RPA deployment using bar plots and regression results. Each evolution (i.e. simulation) is repeated 1000 times and averages are taken to obtain stable and reliable results for analysis.

In Figure 1, line charts are plotted to visualise the time series of process complexity over all the process iterations under different settings of parameters, which highlights the progressive change in process complexity at various stages of evolution due to RPA implementation. Horizontally, *R* increases with the same value of *V*. Vertically, *V* increases with the same value of *R*. The *x*-axis means process iterations, which can be interpreted as time. Line colours represent different levels of *L* and line styles vary for different levels of *F*. Since process complexity is very large in magnitude for some combinations of the parameters, for ease of presentation, base-10 logarithm values of process complexity are computed and displayed below. The scale of all nine line graphs is standardised for the convenience of comparison.

All the lines in Figure 1 demonstrate that process complexity increases at the start of process evolution and declines after the peak and peaks happen roughly at the same process iteration or timestamp across all the different parameter settings of *R*, *V*, *F*, *L*. Complexity rises and declines at a faster pace for greater values of *R* and *V*, which makes the shape of complexity peak more obvious. Also, near the end of process evolution, all the lines show an increasingly gentler gradient, which implies that process complexity tends to stabilise and converge to a value at the end of evolution which can sustain in the long run. Observing the line graphs horizontally and vertically, we can see that the magnitude of the process complexity peak increases with increasing levels of *R* and *V*. Observing different colours of lines which represent different levels of *L*, we can observe that low *L* tends to have the greatest process complexity throughout all process iterations, followed by medium *L* and high *L*. With greater values of *R* and *V*, the gaps between the lines with different settings of *F* and *L* in one graph become wider, which suggests consistent interaction among them over the entire evolution process. Further, by observing various line types, we can see that varying *F* does not have a significant impact on process complexity over time, no matter whether it is at the start, peak or end of evolution.

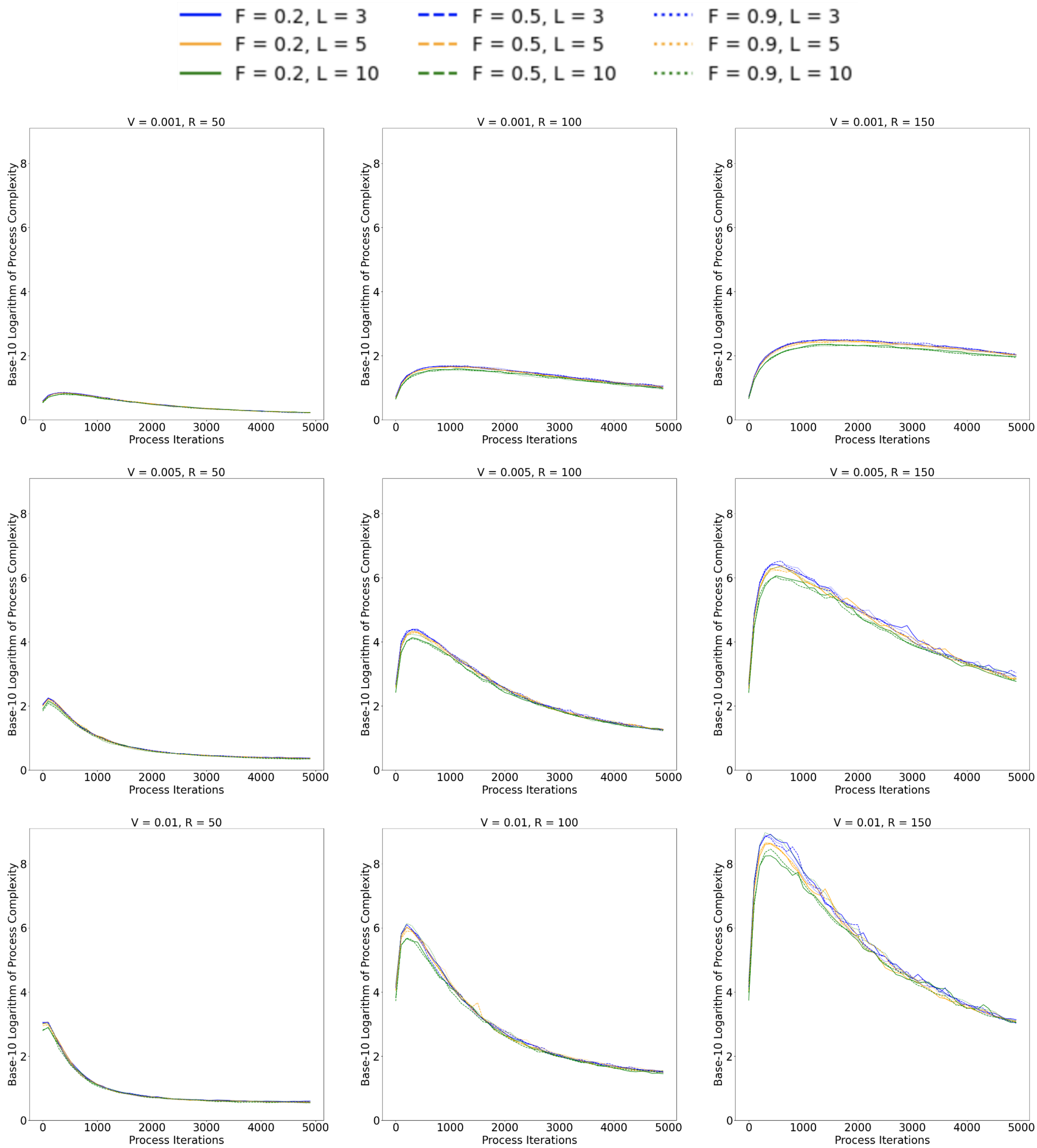


*Figure 1. Line Charts of Base-10 Logarithm of Process Complexity over 5000 Process Iterations for Low, Medium and High Levels of V, R, F, L*

To take a closer look at the final outcome of routine evolution with RPA implementation on process complexity, bar charts are plotted below in Figure 2, which show how the end-of-evolution process complexity differs for various combinations of $V$, $R$, $F$ and $L$. Horizontally, $R$ increases with the same value of $V$. Vertically, $V$ increases with the same value of $R$. The $x$-axis means frequency threshold $F$. Colours of the bars represent different values of $L$ and the value of $L$ increases with colour intensity. Similarly, base-10 logarithm values of process complexity are displayed in the bar charts and the scale of all nine bar charts is standardised for the convenience of comparison.

Observing the bar charts in Figure 2 horizontally, we could visualise the effects of retention *R* on end-of-evolution process complexity. The observation is that process complexity increases with increasing *R*. Higher value of *R* means a longer memory span, which suggests that more historical choices and prior sequences of actions can be remembered by the system. This will enable the system to draw more information based on past experiences and learn further, which stimulates edge formation. As a result, increasing the memory span will increase the end-of-evolution process complexity.

Observing the bar charts in Figure 2 vertically, we could conclude that increasing *V* will result in higher end-of-evolution process complexity. This is because higher variation means a higher probability of randomly jumping to the next node and thus a higher likelihood of having workarounds. There will be more new edges formed which enhances the process complexity at the end of all process iterations. In other words, with a higher value of variation *V*, we could obtain a much more complex network at the end of process iterations, which means higher end-of-evolution process complexity.

Observing every group of bars in Figure 2, increasing *L* has minimal effect on end-of-evolution process complexity. However, when considering the interaction between *F* and *L*, we could observe that at a low level of *F*, increasing *L* will decrease the end-of-evolution process complexity. When the frequency threshold *F* is low, it is more likely for a sequence to be qualified as a suitable candidate for automation by RPA. Thus, for a sequence of actions with a longer length, once the long sequence gets automated, the sequence from one node to the next node is determined by the RPA. This will prevent future formation of new edges stemming from the starting node of RPA sequence, which will lead to a lower end-of-evolution process complexity of the final network formed after 5000 iterations.

Observing each bar chart across its *x*-axis value *F* in Figure 2, we can see that end-of-evolution process complexity does not vary much as *F* changes. This implies that the frequency threshold has negligible effects on end-of-evolution process complexity.

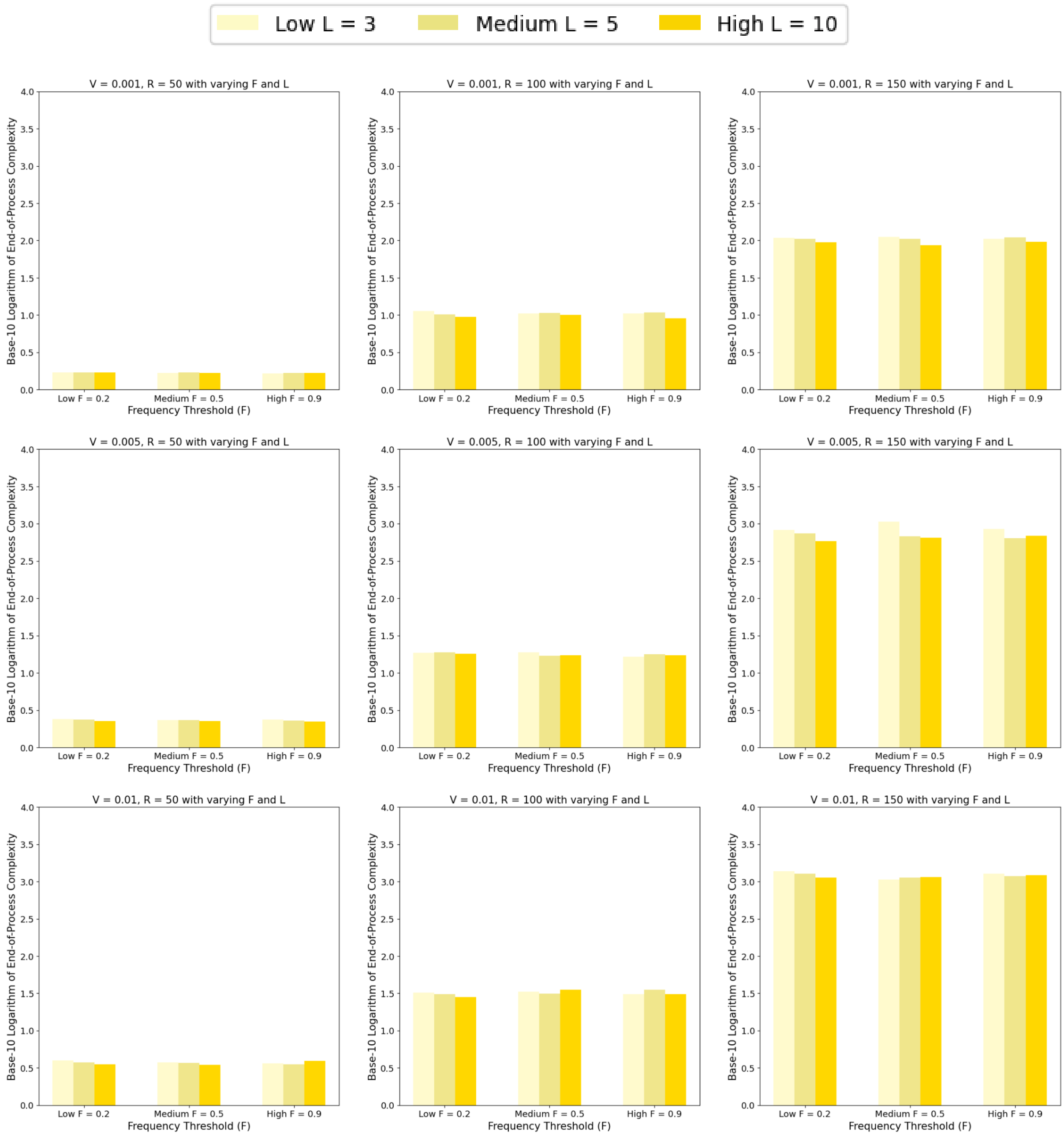


*Figure 2. Bar Charts of Average Base-10 Logarithm of End-of-Evolution Process Complexity at Low, Medium and High Levels of V, R, F, L*

To confirm that the findings from the visual inspection are statistically valid, ordinary least squares (OLS) regression with two-way interactions is run on 81000 observations of end-of-evolution process complexity to identify statistically significant parameters and interaction terms. As parameters are in three levels (i.e. low, medium, high), categorical variables are used in the regression model and thus dummy coding is applied to create binary indicators for two out of three levels of the categorical variable. The OLS regression results are specified in Table 2 below.

Based on the regression results in Table 2, we could notice that $V$ and $R$ at low and medium levels (highlighted in green) are all statistically significant with the $p$-value less than 0.01 and a greater magnitude of coefficient. This indicates that $R$ and $V$ are significant factors which may influence the end-of-evolution process complexity. Even though the $p$-value of $F$ and $L$ at low and medium levels is less than 0.05 (highlighted in orange) suggesting that they can be statistically significant, the much smaller magnitude of coefficient implies that the impacts of $F$ and $L$ without considering interactions are minimal. By observing the interaction terms between $F$ and $L$ at low and medium levels (highlighted in red), the interaction terms all have $p$-value less than 0.01, which suggests that interaction terms between $F$ and $L$ are statistically significant and $F$ and $L$ altogether have a significant impact on end-of-evolution process complexity. Since the coefficient of $Low\ F \times Medium\ L$ is 0.029 which is smaller than the coefficient of $Low\ F \times Low\ L$ being 0.048, our visual inspection on that at a low level of $F$, increasing $L$ will decrease the end-of-evolution process complexity is validated.

***Table 2. Regression Results of Base-10 Logarithm of End-of-Evolution Process Complexity***

| ***OLS Regression Results*** | | | |
|---|---|---|---|
| Dependent Variable | Base-10 Logarithm of End-of-Evolution Process Complexity | | |
| No. of Observations | 81000 | | |
| Adjusted R-squared | 0.768 | | |
| F-statistics | 8393 | | |
| Prob (F-statistics) | 0.000 | | |
| ***Variable*** | ***Coefficient*** | ***Standard Error*** | ***P-value*** |
| Low V | -0.863*** | 0.010 | 0.000 |
| Medium V | -0.262*** | 0.010 | 0.000 |
| Low R | -2.028*** | 0.010 | 0.000 |
| Medium R | -1.254*** | 0.010 | 0.000 |
| Low F | -0.066*** | 0.010 | 0.000 |
| Medium F | -0.089*** | 0.010 | 0.000 |
| Low L | 0.036*** | 0.010 | 0.000 |
| Medium L | 0.022** | 0.010 | 0.029 |
| Low V × Low R | 0.524*** | 0.009 | 0.000 |
| Low V × Medium R | 0.380*** | 0.009 | 0.000 |
| Low V × Low F | 0.043*** | 0.009 | 0.000 |
| Low V × Medium F | 0.052*** | 0.009 | 0.000 |
| Low V × Low L | 0.000 | 0.009 | 0.969 |
| Low V × Medium L | 0.008 | 0.009 | 0.406 |
| Medium V × Low R | 0.063*** | 0.009 | 0.000 |
| Medium V × Medium R | 0.008 | 0.009 | 0.412 |
| Medium V × Low F | 0.044*** | 0.009 | 0.000 |
| Medium V × Medium F | 0.032*** | 0.009 | 0.000 |
| Medium V × Low L | -0.004 | 0.009 | 0.695 |

| | | | |
|---|---|---|---|
| Medium V × Medium L | -0.017* | 0.009 | 0.064 |
| Low R × Low F | 0.019** | 0.009 | 0.035 |
| Low R × Medium F | 0.035*** | 0.009 | 0.000 |
| Low R × Low L | -0.052*** | 0.009 | 0.000 |
| Low R × Medium L | -0.032*** | 0.009 | 0.000 |
| Medium R × Low F | 0.011 | 0.009 | 0.221 |
| Medium R × Medium F | 0.038*** | 0.009 | 0.000 |
| Medium R × Low L | -0.031*** | 0.009 | 0.000 |
| Medium R × Medium L | -0.004 | 0.009 | 0.654 |
| Low F × Low L | 0.048*** | 0.009 | 0.000 |
| Low F × Medium L | 0.029*** | 0.009 | 0.002 |
| Medium F × Low L | 0.045*** | 0.009 | 0.000 |
| Medium F × Medium L | 0.033*** | 0.009 | 0.000 |
| Constant | 2.555*** | 0.009 | 0.000 |
| Remarks: *** p <0.01, ** p <0.05, * p<0.1 | | | |

5.1.2 Probability of Phase Change

As mentioned earlier, a phase change is identified when there is a sudden burst in process complexity immediately followed by a low complexity after the surge. Before we look into how the probability of phase change is impacted by parameter settings of *R*, *V*, *F*, *L* under the influence of RPA deployment, we need to first observe how and when complexity bursts take place over 5000 process iterations and how complexity bursts are affected over time with varying values of parameters.

Figure 3 is a trajectory of the time series of process complexity over 5000 process iterations. We can observe that there is a sudden burst of complexity at about *iteration* = 600, which is followed by later periods of much lower complexity when complexity declines over further iterations. After the complexity burst, the process can settle into a new dominant path that is characterised by much lower complexity. There is a clear cut-off between the stage of complexity surge and the stage of complexity recession until a stable state across the entire period of 5000 process iterations. The stable state achieved is known as a phase change.

A complexity burst happens when new edges form at a faster pace than edge dissolution. Under the influence of RPA implementation, we could observe that the initial burst of complexity is due to the consistent formation of new edges in the network. Once there are repeating sequences of actions in the process memory, those sequences will be automated by RPA. With RPA taking place consistently on some segments of the process, the network after automation will be somewhat fixed to achieve higher operating efficiency. At this moment, a new dominant path will be formed and reinforced over iterations. As a result, there will be less room for new edge formation and the process complexity after RPA is deployed extensively will decline and finally reach a stable level. By looking at Figure 3 below, we notice an interesting observation when we contrast the complexity burst in the RPA network (solid line) with Pentland's original model which is a network without RPA deployed (dashed line). The original model without RPA reaches a greater magnitude of complexity burst than the model with RPA implemented. This is because once RPA happens, the sequence of actions will have fixed edges and the edge formation from the nodes in the RPA sequence will become impossible, which results in a lower maximum magnitude of complexity achieved.

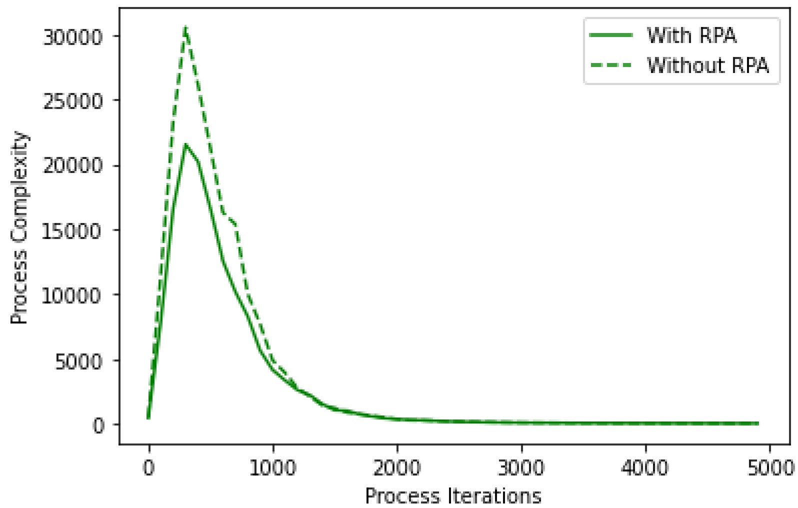


*Figure 3. Process Complexity Burst and the Following Decline with and without RPA Implementation (V = 0.005, R = 100, F = 0.2, L = 5)*

Thus, we can conclude that RPA as an automation technology can still experience complexity bursts and the generative nature of process evolution is preserved. However, RPA reduces the magnitude of the complexity burst in business processes. Also, RPA can better stabilise the network by significantly lowering the complexity and quickly settling into a rigid dominant path. More process iterations after a complexity burst will not provide more opportunities for sequences of actions that have been automated to undergo further phase change or digital innovation.

To further study the impact on complexity burst with respect to varying levels of parameters *R* and *V*, we have an aggregate plot in Figure 4. We fix the values of other parameters (i.e. *F* and *L*) to ensure that other variables do not have additional impacts on the complexity burst.

When we observe the graphs in Figure 4 horizontally, we could find that with increasing retention $R$, the complexity burst happens at a later time and the maximum magnitude of the complexity burst increases. When the retention length is longer, it takes more process iterations to reach full retention. As a result, the complexity burst takes place at a slightly later time. The increase in the maximum magnitude of complexity burst with increasing $R$ is because when the memory space or retention length $R$ increases, more historical choices or past sequences of actions are remembered. There will be more prior information retained, which leads to more new edges formed in the network and thus a greater surge in process complexity. Besides these, we could also observe when $R$ increases, the rate of the decline in process complexity after the burst is slower. RPA is the major reason for the fast complexity declination after the burst. When a process can remember more historical sequences of actions, it is harder to fulfil the RPA requirement on the frequency threshold. Thus, RPA deployment may happen at a slower rate, which leads to a slower pace of the recession in process complexity after the complexity burst.

When we observe the graphs in Figure 4 vertically, we could find that with increasing variation $V$, the complexity burst happens at an earlier time and the maximum magnitude of the complexity burst increases. When the probability of finding a workaround is higher, new edges are formed much faster than edge removal. A surge in the number of new edges will increase the process complexity rapidly, which leads to the observation that complexity burst happens sooner with a much larger magnitude of complexity burst with increasing value of variation $V$. Besides these observations, we can also see that when $V$ increases, the rate of decline in process complexity after the burst is faster. An increase in $V$ implies higher graph density as the network becomes more complex with more edges introduced. When graph density is higher, variations will be more likely to reinforce an existing edge (Pentland, 2020). When the existing edges are reinforced and they frequently appear in the retention, it is more likely to be chosen as suitable candidates for RPA. As a result, with increasing $V$, the process will have more processes undergoing automation which will not be further affected by any workarounds, thus the complexity will decline faster after the complexity burst.

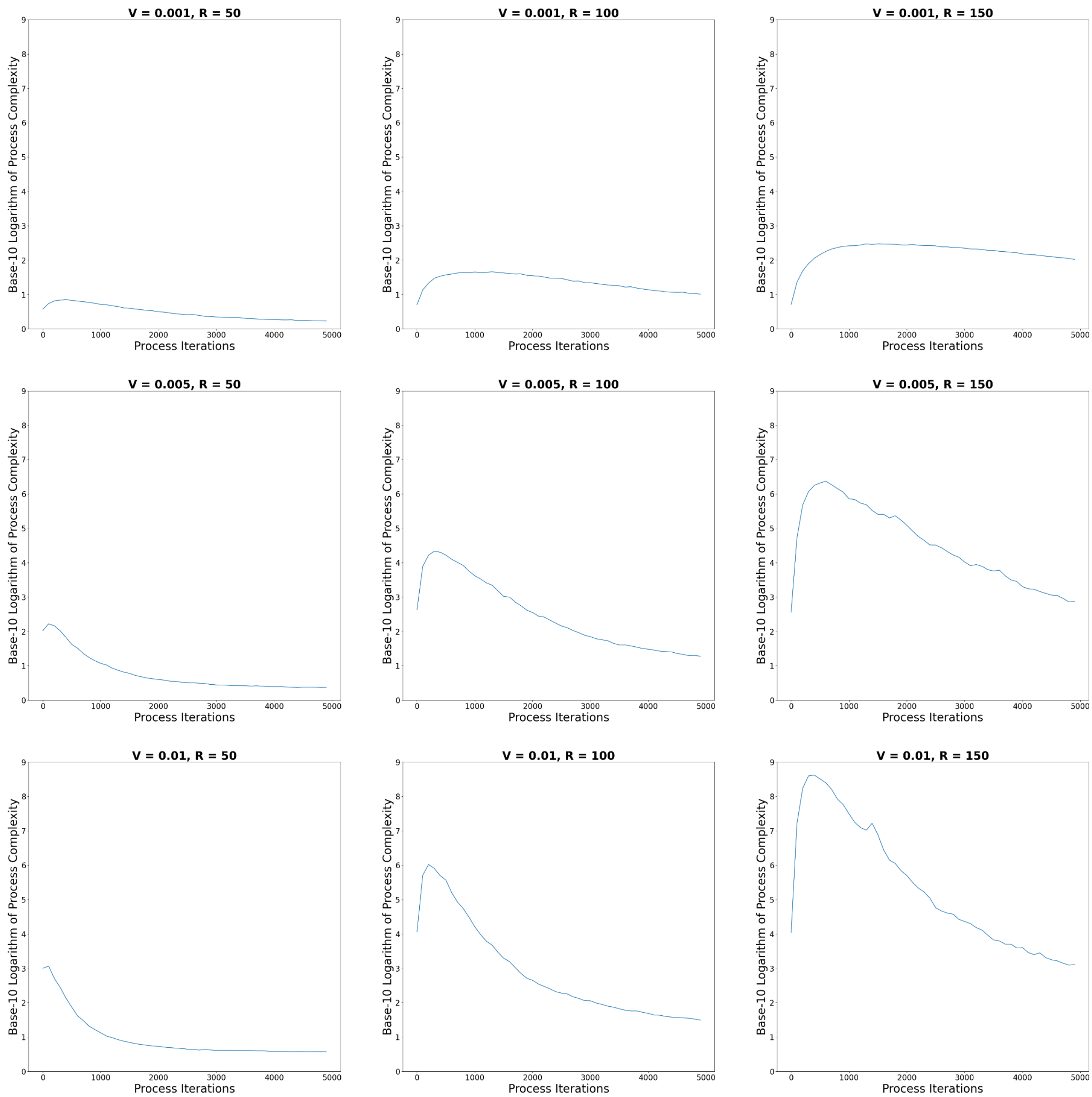


*Figure 4. Line Charts of Base-10 Logarithm of Process Complexity over 5000 Process Iterations at Low, Medium and High Levels of V, R with Fixed F = 0.2 and Fixed L = 5*

To understand the impact on complexity burst with respect to varying values of parameters $F$ and $L$, we have an aggregate plot in Figure 5. In the setting, we fix the values of other parameters (i.e. $R$ and $V$) to ensure that their effects on the complexity burst are controlled.

As observed in the graphs in Figure 5 horizontally, with an increasing value of $L$, the maximum magnitude of the complexity burst decreases. This is because when the length of sequences of actions for RPA is longer, more nodes in the network will have their next node determined and fixed in the sequence. This reduces the probability of edge formation as more actions in a sequence are programmed and automated, which leads to a lower maximum magnitude of complexity burst. However, the effect of frequency threshold $F$ on the complexity burst is random and hard to be patterned because we notice minimal changes when viewing the graphs in Figure 5 vertically.

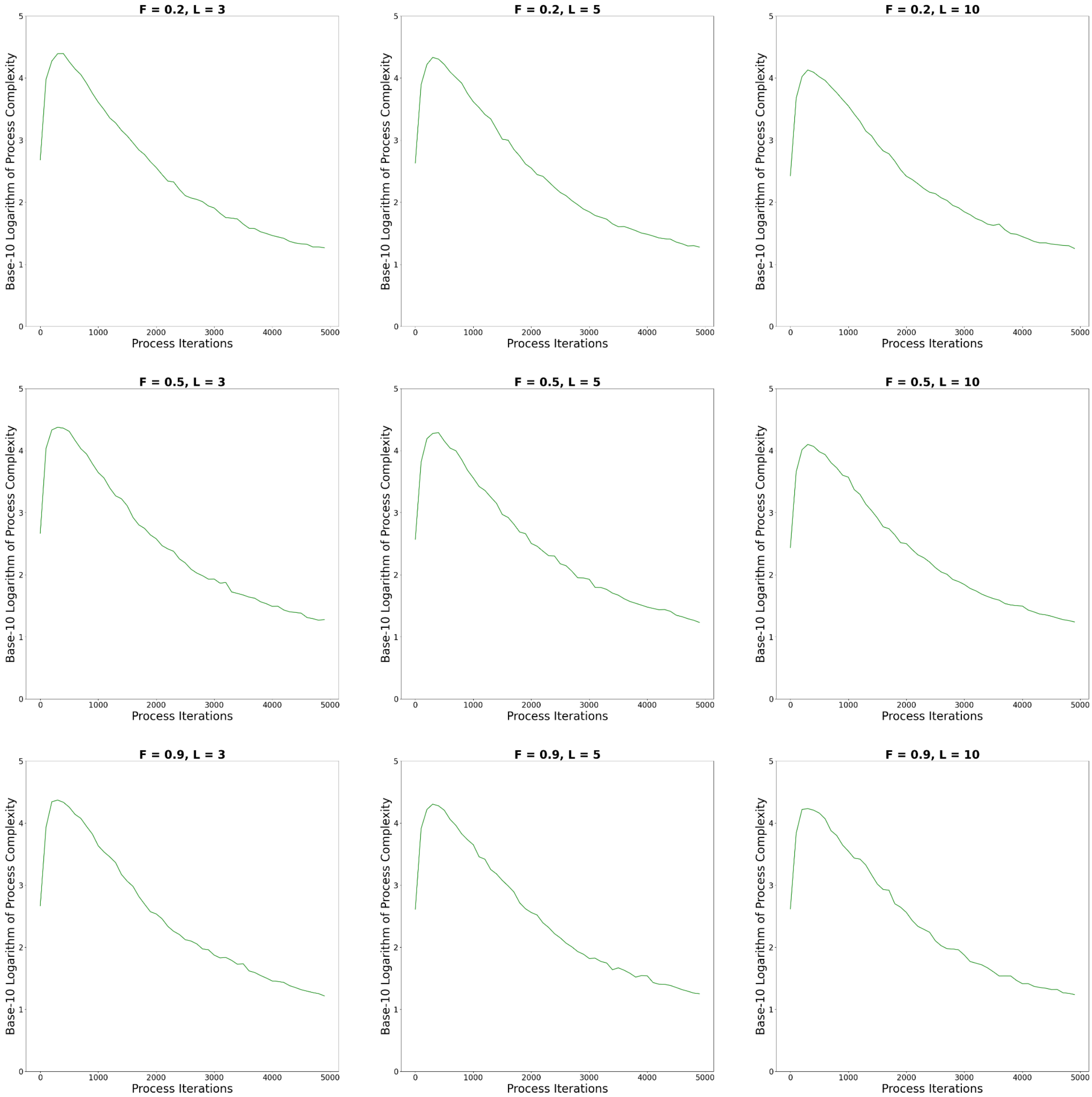


*Figure 5. Line Charts of Base-10 Logarithm of Process Complexity over 5000 Process Iterations at Low, Medium and High Levels of F, L with Fixed V = 0.005 and Fixed R = 100*

To understand the effects of RPA on the probability of phase change, we observe how the probability of phase change throughout the entire process evolution varies with different parameter settings of $R$, $V$, $F$, and $L$ using bar plots and regression results. Each evolution or so-called simulation is repeated 1000 times and averages are taken to obtain stable and reliable results for analysis.

Figure 6 contains bar charts which show the probability of phase change at various combinations of parameters *V*, *R*, *F* and *L*. Horizontally, *R* increases with the same value of *V*. Vertically, *V* increases with the same value of *R*. The *x*-axis means frequency threshold *F*. Colours of the bars represent different values of *L* and the value of *L* increases with colour intensity. The scale of all nine bar charts is standardised from 0 to 1 for the convenience of comparison.

Observing the bar charts in Figure 6 horizontally, we could visualise the effects of retention *R* on the probability of phase change. The observation is that with increasing levels of *R*, drift probability tends to increase. The increase in drift probability is more significant for a transition from a low retention *R* to a medium retention *R*. This is because a too-short memory span will not provide sufficient historical information for a process to settle down into a new path quickly and undergo transformation. A too-long memory span raises the difficulty of having some sequence meet the criteria of RPA and the process freezes more slowly and tends to preserve a more complex network structure after the complexity burst. This explains why a further increase in memory span from medium to high retention will not further boost the drift probability by a lot. This suggests the importance of setting an appropriate length of retention for processes in order to enable and encourage transformation.

Observing the bar charts in Figure 6 vertically, we could conclude that increasing *V* has minimal effects on the probability of phase change. This may be attributed to the effect of RPA: as more sequences in a process get automated by RPA, variation *V* will no longer affect the edge formation and edge dissolution on the nodes in the automated sequence. This means that the effects of varying *V* values will have marginal effects on drift probability for processes with RPA deployed.

Observing every group of bars in Figure 6, we could observe a surprising result which is that increasing *L* has minimal effects on the probability of phase change. This implies that the length of the sequence of actions for RPA has negligible effects on the drift probability, in other words, the probability of having the evolving process settle into a new dominant path.

Observing each graph across its *x*-axis value *F* in Figure 6, another surprising result can also be observed, which is that the probability of phase change does not vary much as *F* changes.

This implies that the frequency threshold has negligible effects on the drift probability and likelihood of whether a process undergoes a transformation.

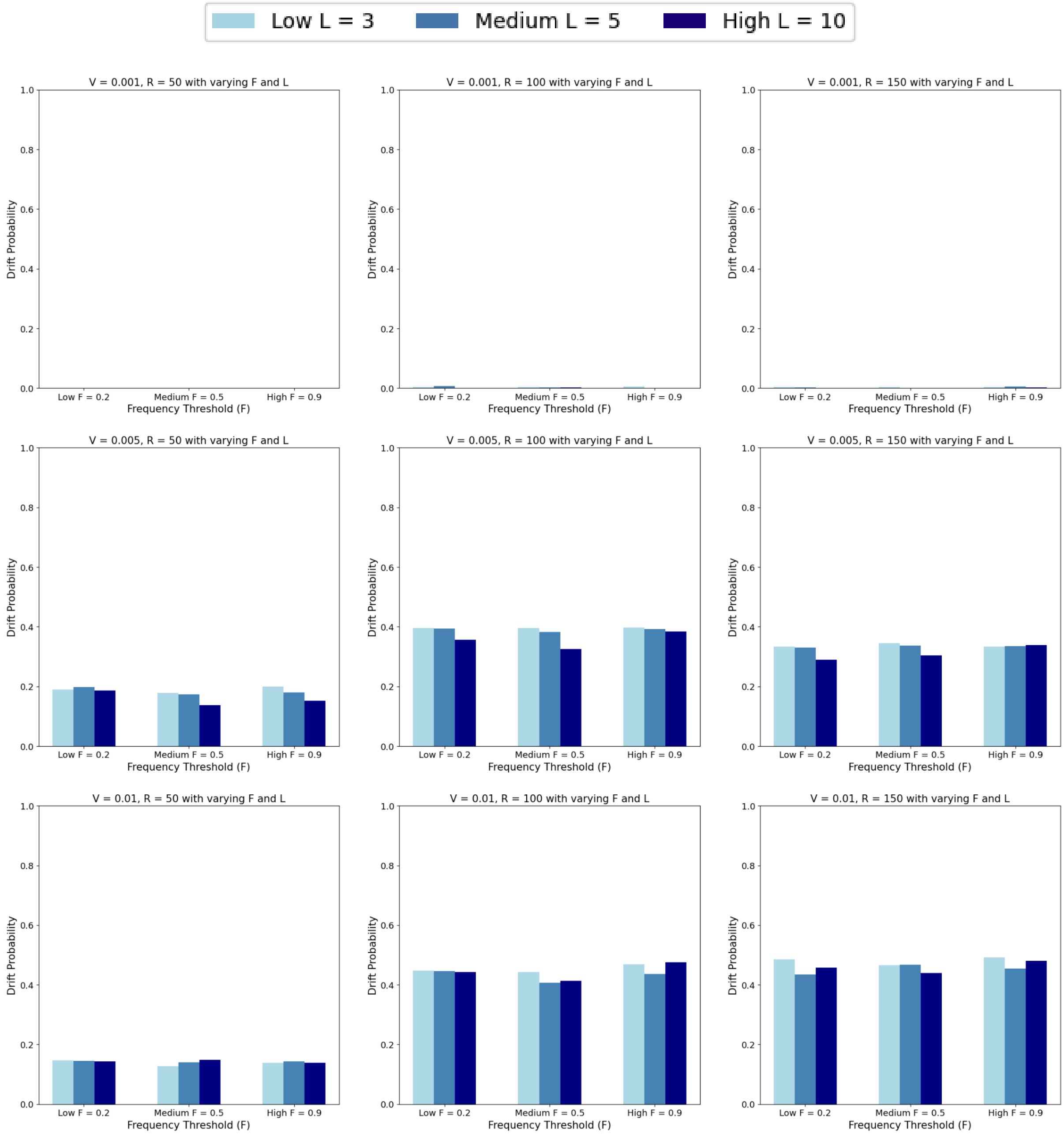


*Figure 6. Bar Charts of the Probability of Phase Change throughout 5000 Process Iterations at Low, Medium and High Levels of V, R, F, L*

To confirm that the findings from visual inspections are statistically valid, logistic regression with two-way interactions is run on 81000 observations of the probability of phase change to identify statistically significant parameters and interaction terms. As parameters are in three levels (i.e. low, medium, high), categorical variables are used in the regression model and thus dummy coding is applied to create binary indicators for two out of three levels of the categorical variable. The logistic regression results are specified in Table 3 below.

Based on the regression results in Table 3, we could observe that low *R* is statistically significant with *p*-value < 0.01 but high *R* is statistically insignificant with *p*-value being 0.576 > 0.1 (highlighted in green). The coefficient of low *R* is negative and its magnitude is relatively higher. Thus, we can conclude that increasing *R* from low to medium levels will increase the probability of phase change. Observing the coefficient of *V* and its interaction terms (highlighted in orange), the coefficients of medium and high *V* are significant with comparably large magnitudes (i.e. 6.048 and 6.402). Also, the coefficient of the interaction term *Medium V* ✕ *Low R* is statistically significant. This highlights that increasing *V* from low to medium level significantly increases the probability of phase change when it has low *R* but increasing *V* from medium to high level only brings a gentle further increase in drift probability. Most of the interaction terms of *V* are statistically insignificant because *p*-value is more than 0.1, which suggests that *V* has marginal effects on the probability of phase change under the influence of RPA implementation. Observing the *p*-values of *F* and its interaction terms with *V, R, L* (highlighted in red), we could find that almost all have *p*-values more than 0.1 and thus the coefficients are statistically insignificant. This confirms that *F* has marginal effects on the probability of phase change. Similarly, the interaction terms of *L* with *R* and *F* (highlighted in purple) are statistically insignificant because most have *p*-value more than 0.05, which confirms our previous finding that *L* has minimal effects on drift probability under the influence of RPA implementation.

| ***Table 3. Regression Results of the Probability of Phase Change*** | | | |
|---|---|---|---|
| ***Logistic Regression Results using Maximum Likelihood Estimation (MLE)*** | | | |
| Dependent Variable | Probability of Phase Change | | |
| No. of Observations | 81000 | | |
| Pseudo R-squared | 0.2358 | | |
| Log-Likelihood | -32244 | | |
| log-likelihood ratio p-value | 0.000 | | |
| ***Variable*** | ***Coefficient*** | ***Standard Error*** | ***P-value*** |
| Medium V | 6.048*** | 0.446 | 0.000 |
| High V | 6.402*** | 0.446 | 0.000 |
| Low R | -2.480*** | 0.737 | 0.001 |
| High R | -0.166 | 0.297 | 0.576 |
| Medium F | -0.416 | 0.368 | 0.258 |
| High F | 0.051 | 0.337 | 0.879 |

| | | | |
|---|---|---|---|
| Low L | 1.154*** | 0.436 | 0.008 |
| Medium L | 1.069** | 0.441 | 0.015 |
| Medium V × Low R | 1.483** | 0.736 | 0.044 |
| Medium V × High R | -0.110 | 0.295 | 0.709 |
| Medium V × Medium F | 0.303 | 0.366 | 0.408 |
| Medium V × High F | 0.039 | 0.336 | 0.908 |
| Medium V × Low L | -0.984** | 0.436 | 0.024 |
| Medium V × Medium L | -0.926** | 0.441 | 0.036 |
| High V × Low R | 0.960 | 0.736 | 0.192 |
| High V × High R | 0.212 | 0.295 | 0.472 |
| High V × Medium F | 0.295 | 0.366 | 0.421 |
| High V × High F | 0.050 | 0.336 | 0.881 |
| High V × Low L | -1.110** | 0.436 | 0.011 |
| High V × Medium L | -1.100** | 0.441 | 0.013 |
| Low R × Medium F | -0.050 | 0.062 | 0.421 |
| Low R × High F | -0.120* | 0.062 | 0.051 |
| Low R × Low L | -0.020 | 0.063 | 0.747 |
| Low R × Medium L | 0.040 | 0.063 | 0.525 |
| High R × Medium F | 0.102* | 0.053 | 0.054 |
| High R × High F | 0.028 | 0.053 | 0.600 |
| High R × Low L | 0.002 | 0.053 | 0.965 |
| High R × Medium L | -0.001 | 0.053 | 0.979 |
| Medium F × Low L | 0.052 | 0.058 | 0.369 |
| Medium F × Medium L | 0.054 | 0.058 | 0.348 |
| High F × Low L | -0.053 | 0.057 | 0.354 |
| High F × Medium L | -0.086 | 0.057 | 0.134 |
| Constant | -6.630*** | 0.446 | 0.000 |
| Remarks: *** p <0.01, ** p <0.05, * p<0.1 | | | |

### 5.1.3 Magnitude of Change

To understand the effects of RPA on the magnitude of change in process from the initial simple happy path, we observe the time series of the magnitude of change over 5000 process iterations using line charts and then we further look into how the end-of-evolution magnitude of change as compared to the initial simple happy path varies with different parameter settings of $R$, $V$, $F$, and $L$ under the influence of RPA deployment using bar plots and regression results. Each evolution or so-called simulation is repeated 1000 times and averages are taken to obtain stable and reliable results for analysis.

Line charts are plotted in Figure 7 to visualise the time series of the magnitude of change over all the 5000 process iterations under different settings of parameters, which highlights the progressive change in magnitude of change at various stages of evolution due to RPA implementation. Horizontally, $R$ increases with the same value of $V$. Vertically, $V$ increases with the same value of *R*. The *x*-axis means process iterations, which can be interpreted as time. Line colours represent different levels of *L* and line styles vary for different levels of *F*. The scale of all nine line graphs is standardised for the convenience of comparison.

All the lines in Figure 7 demonstrate that magnitude of change in process increases significantly at the start of process evolution across all the different parameter settings of *R*, *V*, *F*, *L*. Increasing *R* implies that the initial surge in the magnitude of change is at a faster speed. At a low level of *R*, the magnitude of change rises with an increasingly gentler gradient over time and tends to stabilise and converge to a fixed value sooner. However, at medium and high levels of *R*, the initial surge in the magnitude of change slows down at about the 1000th process iteration and then experiences another quick increase after a short period of stagnation. Besides, greater *V* also increases the likelihood of having the magnitude of change stabilise near the end of process evolution. Horizontally, we could see that the converging value of the magnitude of change at the end of evolution decreases at low *V*, stays constant at medium *V* and increases at high *V*. Observing different colours of lines which represent different levels of *L*, we can observe that low *L* tends to have the greatest magnitude of change throughout all process iterations, followed by medium *L* and high *L*. Further, by observing various line types, we can see that varying *F* does not have a significant impact on the magnitude of change over time as the lines are quite close to one another.

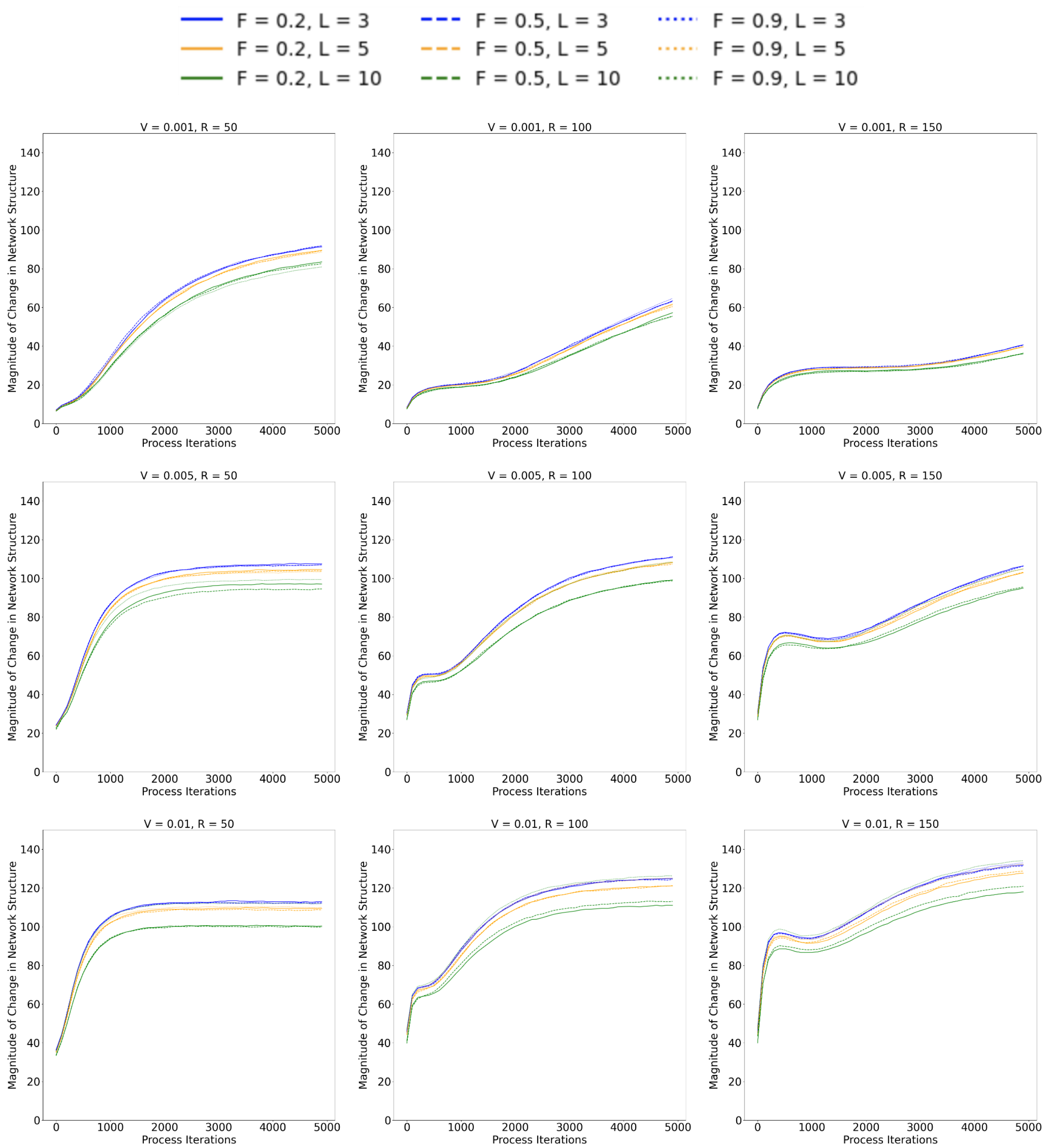


*Figure 7. Line Charts of Magnitude of Change in Process from the Original Network over 5000 Process Iterations at Low, Medium and High Levels of V, R, F, L*

To take a closer look at the magnitude of change in the final network with respect to the initial simple happy path under the influence of RPA implementation, bar charts are plotted in Figure 8, which show how the end-of-evolution magnitude of change from the initial path differs for various combinations of $V$, $R$, $F$ and $L$. Horizontally, $R$ increases with the same value of $V$. Vertically, $V$ increases with the same value of $R$. The $x$-axis means frequency threshold $F$. Colours of the bars represent different values of $L$ and the value of $L$ increases with colour intensity. The scale of all nine bar charts is standardised for the convenience of comparison.

Observing the bar charts in Figure 8 horizontally, we could visualise the effects of retention on the magnitude of change. To analyse the effect of *R* on the magnitude of change, we have to consider the interaction between *V* and *R*. As observed, when *V* is at a low level (i.e. $V = 0.001$), increasing retention length *R* will lead to a lower magnitude of change across all combinations of *F* and *L*. When *V* is s at a medium level (i.e. $V = 0.005$), increasing retention length *R* will have negligible effects on the magnitude of change across all settings of *F* and *L*. When *V* is s at a high level (i.e. $V = 0.01$), increasing retention length *R* will lead to a higher magnitude of change across all settings of *F* and *L*. When variation is low, increasing retention means that it is harder for a sequence of action to qualify for RPA and the structural change is less vigorous, which leads to a lower magnitude of change. When variation is high, the graph will have a higher density. Now even with longer retention, the variation is very likely to reinforce an existing edge, which makes some sequence of action repetitively appear in the memory. This will help the sequence qualify for RPA. With automation, there will be more drastic changes in the structure of the network, which leads to a higher magnitude of change in the final network as compared to the initial simple happy path.

Observing the bar charts in Figure 8 vertically, we could conclude that increasing *V* will result in a higher magnitude of change. This is because higher variation means a higher probability of randomly jumping to the next node and thus a higher likelihood of having workarounds. The final network will undergo more drastic structural changes at the end of all the 5000 iterations, which leads to a higher magnitude of change with an increasing variation.

Observing every group of bars in Figure 8, increasing *L* generally decreases the magnitude of change. RPA is the major source of drastic structural change in routine evolution. When the sequence length for RPA is longer, it is harder to find qualified sequences to undergo RPA and the likelihood of a sequence of actions undergoing RPA is lower. With less automation taking place, the magnitude of change will be lower. However, there is some exception at a high frequency threshold *F*: increasing *L* leads to a higher magnitude of change when variation is at a high state. When the frequency threshold *F* is higher, which means the sequence has appeared very frequently in recent history. Given a high *V*, which suggests that there is a higher probability of workarounds, automating a long sequence of actions can bring more drastic structural change to the existing network than automating shorter slices of actions. As a result, the end-of-evolution magnitude of change will be larger in this case.

Observing each graph across its $x$-axis value $F$ in Figure 8, the magnitude of change does not vary much as $F$ changes. This implies that the frequency threshold has negligible effects on the magnitude of change. This can be due to the strong power of automation in fixing the structure of the network as once a sequence of actions (e.g. from node 4 to node 5) is automated, all the possible edges beginning from node 4 become redundant as the probability of entering node 5 after node 4 is 100%. Thus, regardless of the frequency threshold, throughout the 5000 iterations, once some sequences are automated by RPA, its impact on the entire structural change is huge. However, there is a special case where for a high $V$, increasing $F$ will lead to a more significant increase in the magnitude of change. At high $V$, there is a higher probability of workarounds and it is more likely to create innovative paths by jumping much forward towards the last node. Once a sequence of actions gets automated due to its high frequency, we could think that an excellent sequence which is very worth automation gets programmed, which leads to more significant structural change as compared to automating shorter slices of sequence.

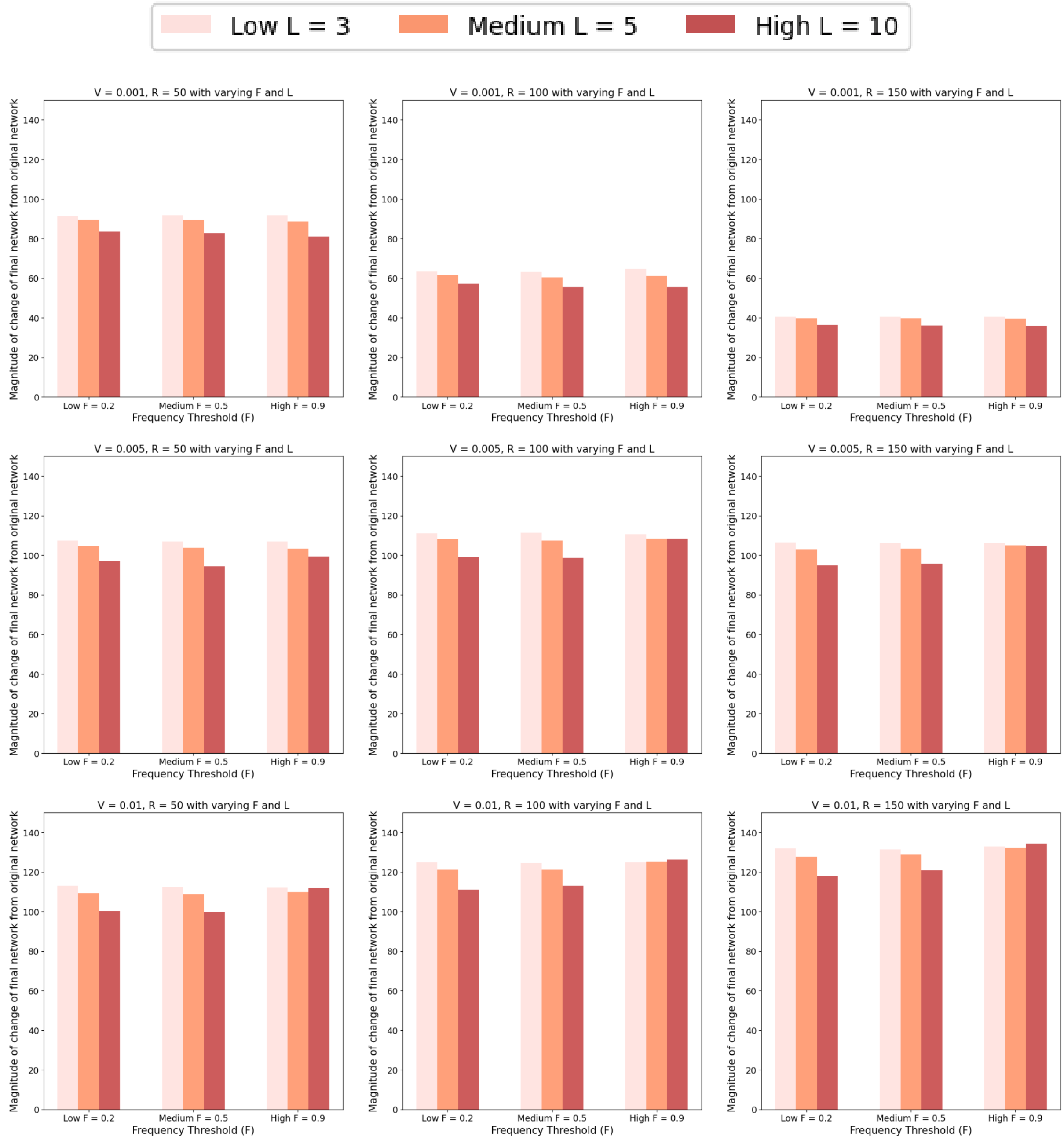


*Figure 8. Bar Charts of Average End-of-Evolution Magnitude of Change in Process from the Original Simple Happy Path at Low, Medium and High Levels of V, R, F, L*

To confirm that the findings from the visual inspection are statistically valid, OLS regression with two-way interactions is run on 81000 observations of the end-of-evolution magnitude of change in process to identify statistically significant parameters and interaction terms. As parameters are in three levels (i.e. low, medium, high), categorical variables are used in the regression model and thus dummy coding is applied to create binary indicators for two out of three levels of the categorical variable. The OLS regression results are specified in Table 4 below.

| *Table 4. Regression Results of the End-of-Evolution Magnitude of Change in Process* | | | |
|---|---|---|---|
| *OLS Regression Results* | | | |
| Dependent Variable | End-of-Evolution Magnitude of Change in Process | | |
| No. of Observations | 81000 | | |
| Adjusted R-squared | 0.854 | | |
| F-statistics | 1.476e+04 | | |
| Prob (F-statistics) | 0.000 | | |
| *Variable* | *Coefficient* | *Standard Error* | *P-value* |
| Medium V | 14.932*** | 0.260 | 0.000 |
| High V | 19.002*** | 0.260 | 0.000 |
| Medium R | -28.727*** | 0.260 | 0.000 |
| High R | -51.381*** | 0.260 | 0.000 |
| Medium F | -0.996*** | 0.260 | 0.000 |
| High F | -4.546*** | 0.260 | 0.000 |
| Medium L | -2.713*** | 0.260 | 0.000 |
| High L | -10.591*** | 0.260 | 0.000 |
| Medium V × Medium R | 31.745*** | 0.241 | 0.000 |
| Medium V × High R | 49.093*** | 0.241 | 0.000 |
| Medium V × Medium F | 0.020 | 0.241 | 0.934 |
| Medium V × High F | 2.907*** | 0.241 | 0.000 |
| Medium V × Medium L | -0.976*** | 0.241 | 0.000 |
| Medium V × High L | -1.864*** | 0.241 | 0.000 |
| High V × Medium R | 40.121*** | 0.241 | 0.000 |
| High V × High R | 68.977*** | 0.241 | 0.000 |
| High V × Medium F | 0.847*** | 0.241 | 0.000 |
| High V × High F | 6.291*** | 0.241 | 0.000 |
| High V × Medium L | -0.627*** | 0.241 | 0.009 |
| High V × High L | -0.911*** | 0.241 | 0.000 |
| Medium R × Medium F | 0.423* | 0.241 | 0.079 |
| Medium R × High F | 2.014*** | 0.241 | 0.000 |
| Medium R × Medium L | 0.340 | 0.241 | 0.158 |
| Medium R × High L | 1.130*** | 0.241 | 0.000 |
| High R × Medium F | 1.105*** | 0.241 | 0.000 |
| High R × High F | 2.661*** | 0.241 | 0.000 |
| High R × Medium L | 0.950*** | 0.241 | 0.000 |
| High R× High L | 2.614*** | 0.241 | 0.000 |
| Medium F × Medium L | -0.049 | 0.241 | 0.839 |
| Medium F × High L | 0.122 | 0.241 | 0.612 |
| High F × Medium L | 0.894*** | 0.241 | 0.000 |
| High F × High L | 6.520*** | 0.241 | 0.000 |
| Constant | 93.168*** | 0.230 | 0.000 |
| Remarks: *** p <0.01, ** p <0.05, * p<0.1 | | | |

Based on the regression results in Table 4, we could notice that *V* and *R* at medium and high levels as well as their interaction terms (highlighted in green) are all statistically significant with *p*-value less than 0.01 and a much greater magnitude of coefficient. This indicates that *R* and *V* as well as their interaction terms are significant factors which may influence the end-of-evolution magnitude of change in process. Besides, *L* is also a statistically significant parameter because the *p*-value of *L* at medium and high levels is less than 0.05 and coefficient has a greater magnitude (highlighted in orange). Since the coefficient of *High L* is -10.591 which is lower than the coefficient of *Medium L* being -2.713, it validates that decreasing *L* is correlated with a higher magnitude of change. By observing the interaction terms between *F* and *High V* (highlighted in red), the interaction terms all have *p*-value less than 0.01, which suggests that interaction terms between *F* and *High V* are statistically significant and *F* and *High V* altogether have a significant impact on the end-of-evolution magnitude of change. As the coefficient of *High V* × *Medium F* is 0.847 which is smaller than the coefficient of *High V* × *High F* being 6.291, it is testified that at a high level of *V*, increasing *F* will increase the end-of-evolution magnitude of change.

5.1.4 Efficiency

To understand the effects of RPA on efficiency gain from the initial simple happy path, we observe the time series of efficiency gain over 5000 process iterations using line charts and then we further look into how the ultimate efficiency gain at the end of evolution from the initial simple happy path varies with different parameter settings of *R*, *V*, *F*, and *L* under the influence of RPA deployment using bar plots and regression results. Each evolution or so-called simulation is repeated 1000 times and averages are taken to obtain stable and reliable results for analysis.

Line charts are plotted in Figure 9 to visualise the time series of efficiency gain over all 5000 process iterations under different settings of parameters, which highlights the progressive change in efficiency gain at various stages of evolution due to RPA implementation. Horizontally, *R* increases with the same value of *V*. Vertically, *V* increases with the same value of *R*. The *x*-axis means process iterations, which can be interpreted as time. Line colours represent different levels of *L* and line styles vary for different levels of *F*. The scale of all nine line graphs is standardised for the convenience of comparison.

All the lines in Figure 9 show that efficiency gain at the start of routine evolution increases sharply and then gradually converges to a stable stage over time across all the different parameter settings of *R*, *V*, *F*, *L*. Observing the line graphs vertically, low *V* gives rise to the greatest overall increase in efficiency gain from the start to the end of evolution, followed by medium and high *V*. Horizontally, we can notice that increasing *R* makes it earlier for evolution to reach a stable and fixed level of efficiency gain, as soon as the 1000th process iteration. Observing different colours of lines which represent different levels of *L*, we can observe that low *L* tends to have the greatest efficiency gain throughout all process iterations, followed by medium *L* and high *L*. Further, by observing various line types, we can see that at medium and high *V*, increasing *F* tends to give greater efficiency gain through the entire process evolution. This effect is more significant for longer sequences of actions with greater value of *L*.

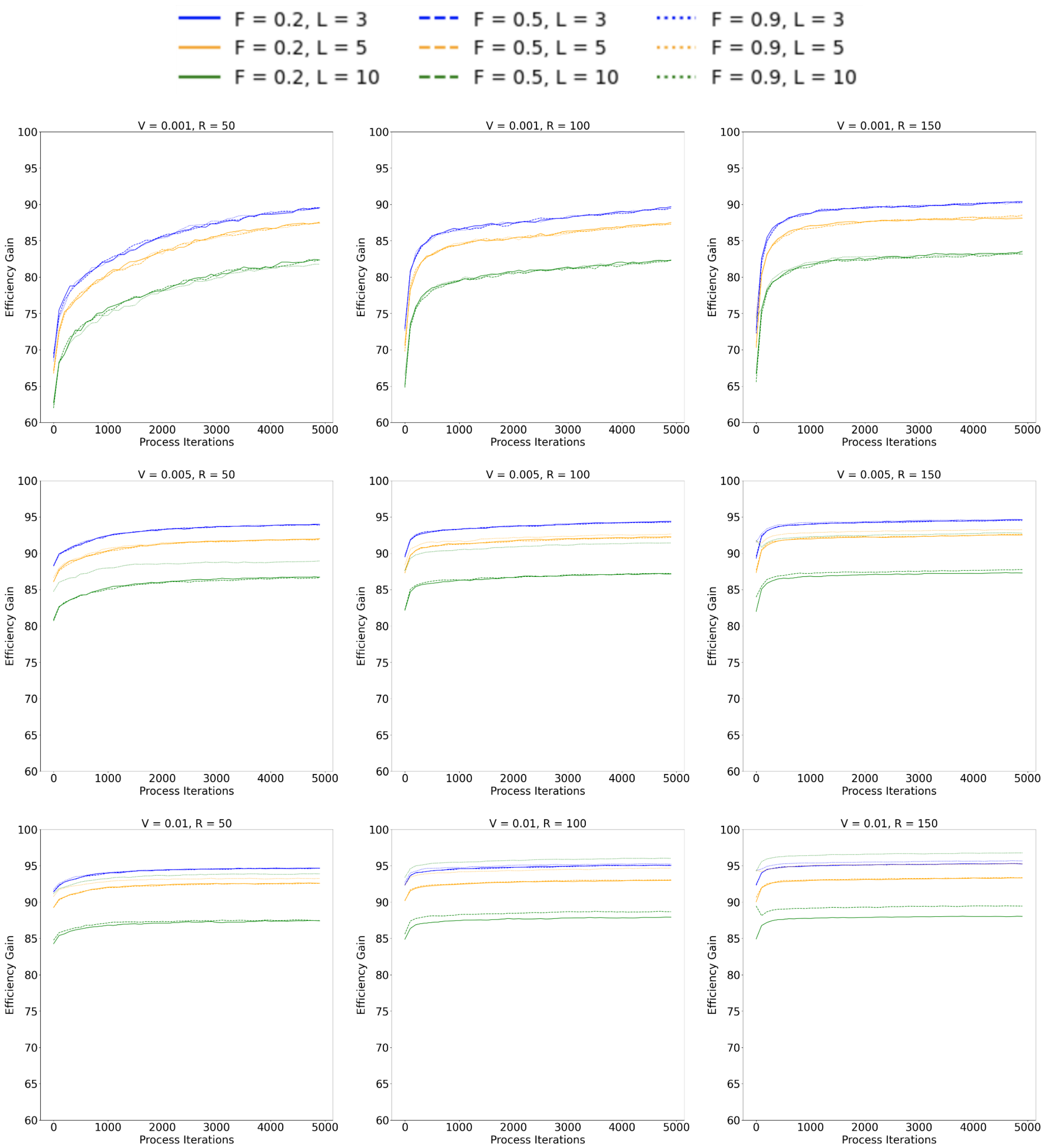


*Figure 9. Line Charts of Efficiency Gain from the Original Simple Happy Path over 5000 Process Iterations at Low, Medium and High Levels of V, R, F, L*

To take a closer look at the efficiency gain at the end of process evolution from the initial simple happy path under the influence of RPA implementation, bar charts are plotted in Figure 10, which show how the efficiency gain at the end of evolution from the initial path differs for various combinations of $V$, $R$, $F$ and $L$. Horizontally, $R$ increases with the same value of $V$. Vertically, $V$ increases with the same value of $R$. The $x$-axis means frequency threshold $F$. Colours of the bars represent different values of $L$ and the value of $L$ increases with colour intensity. The scale of all nine bar charts is standardised for the convenience of comparison.

Observing the bar charts in Figure 10 horizontally, increasing *R* generally leads to higher efficiency gain and the increase is more significant for higher frequency threshold. Increasing retention means that there will be more influence from the historical choices or memory on the current process evolution. Consistently repeating sequences are remembered by the memory and the memory will stimulate RPA to take place, which enhances efficiency gain.

Observing the bar charts in Figure 10 vertically, increasing *V* leads to higher efficiency gain, which contributes to the most significant effect on efficiency gain. Higher variation introduces a higher probability of creating a new path by jumping to a further node which substantially boosts the efficiency. Besides, under a network with high density, higher variation will be more likely to reinforce an existing edge. Thus, higher variation will lead to more RPA and enhance the efficiency gain through the process.

Observing every group of bars in Figure 10, increasing *L* generally decreases efficiency gain. RPA is the major driving force for efficiency gain in the process evolution. When the length of the sequence of actions for RPA is longer, it is harder to find qualified sequences to undergo RPA, which leads to lower efficiency gain. However, one exception we could observe here is that at a higher value of *F*, increasing *L* may lead to higher efficiency gain when the variation is at a high level. This is an interesting observation. When the frequency threshold is higher, which means the sequence has appeared very frequently in recent history. Given a high variation level, which suggests that there is a higher probability of workarounds, automating a long sequence of actions can be more effective and efficient than automating shorter slices of actions. This can be attributed to the fact that once a very long sequence of actions gets automated, it will significantly reduce the possibility of having shorter sequences of actions being automated. With a high variation, the process is more likely to reach a very efficient and shortest path quickly. This will further boost the efficacy gain from RPA.

Observing each graph across its *x*-axis value *F* in Figure 10, efficiency gain increases as *F* increases at high *R*. Even though increasing *F* means the criterion for RPA is stricter and there is less likelihood for a sequence to be automated especially when the memory span is longer, a higher standard means that the sequences which are automated are high-quality candidates and once they are automated, the efficiency gain can be huge. Besides, the increase in efficiency gain with increasing $F$ is more significant at a high level of $L$. A sequence of actions with a large length but still appearing very frequently is considered an excellent candidate qualified for RPA. Once this sequence gets automated, the efficiency gain can be further boosted.

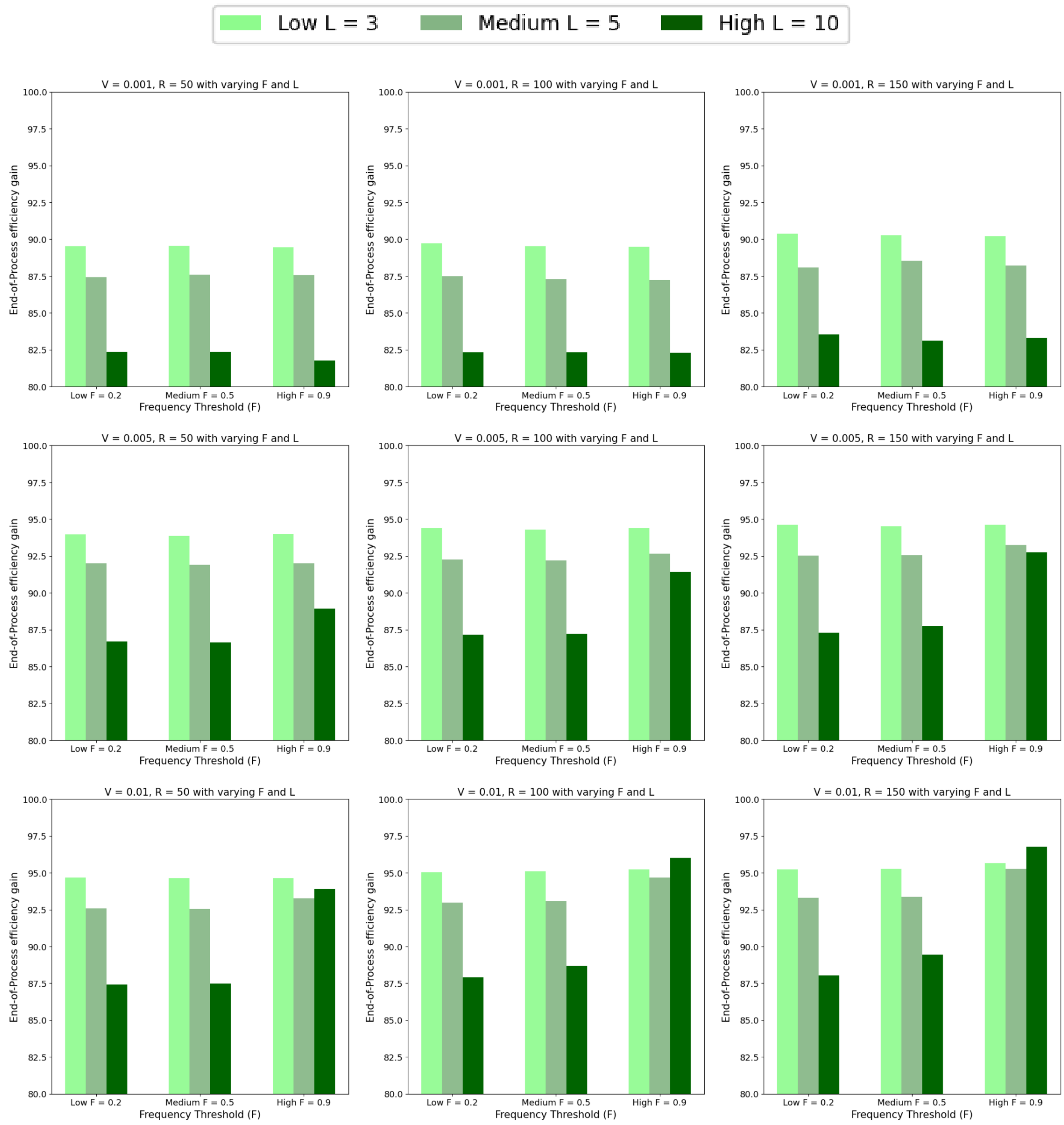


*Figure 10. Bar Charts of Average End-of-Evolution Efficiency Gain at Low, Medium and High Levels of V, R, F, L*

To confirm that the findings from the visual inspection are statistically valid, OLS regression with two-way interactions is run on 81000 observations of efficiency grain at the end of process evolution to identify statistically significant parameters and interaction terms. As parameters are in three levels (i.e. low, medium, high), categorical variables are used in the regression model and thus dummy coding is applied to create binary indicators for two out of three levels of the categorical variable. The OLS regression results are specified in Table 5.

***Table 5. Regression Results of the Efficiency Gain at the End of Process Evolution***

| *OLS Regression Results* | | | |
|---|---|---|---|
| Dependent Variable | Efficiency Gain at the End of Process Evolution | | |
| No. of Observations | 81000 | | |
| Adjusted R-squared | 0.638 | | |
| F-statistics | 4459 | | |
| Prob (F-statistics) | 0.000 | | |
| ***Variable*** | ***Coefficient*** | ***Standard Error*** | ***P-value*** |
| Medium V | 3.669*** | 0.067 | 0.000 |
| High V | 3.673*** | 0.067 | 0.000 |
| Medium R | -0.398*** | 0.067 | 0.000 |
| High R | 0.160** | 0.067 | 0.017 |
| Medium F | -0.261*** | 0.067 | 0.000 |
| High F | -2.109*** | 0.067 | 0.000 |
| Medium L | -2.380*** | 0.067 | 0.000 |
| High L | -9.046*** | 0.067 | 0.000 |
| Medium V × Medium R | 0.651*** | 0.062 | 0.000 |
| Medium V × High R | 0.202*** | 0.062 | 0.001 |
| Medium V × Medium F | 0.029 | 0.062 | 0.635 |
| Medium V × High F | 1.608*** | 0.062 | 0.000 |
| Medium V × Medium L | 0.157** | 0.062 | 0.011 |
| Medium V × High L | 1.334*** | 0.062 | 0.000 |
| High V × Medium R | 0.832*** | 0.062 | 0.000 |
| High V × High R | 0.355*** | 0.062 | 0.000 |
| High V × Medium F | 0.297*** | 0.062 | 0.000 |
| High V × High F | 3.285*** | 0.062 | 0.000 |
| High V × Medium L | 0.470*** | 0.062 | 0.000 |
| High V × High L | 2.765*** | 0.062 | 0.000 |
| Medium R × Medium F | 0.056 | 0.062 | 0.369 |
| Medium R × High F | 0.579*** | 0.062 | 0.000 |
| Medium R × Medium L | 0.026 | 0.062 | 0.675 |
| Medium R × High L | 0.555*** | 0.062 | 0.000 |
| High R × Medium F | 0.213*** | 0.062 | 0.001 |
| High R × High F | 0.910*** | 0.062 | 0.000 |
| High R × Medium L | 0.195*** | 0.062 | 0.002 |
| High R× High L | 0.887*** | 0.062 | 0.000 |
| Medium F × Medium L | 0.106* | 0.062 | 0.087 |
| Medium F × High L | 0.309*** | 0.062 | 0.000 |
| High F × Medium L | 0.584*** | 0.062 | 0.000 |
| High F × High L | 3.799*** | 0.062 | 0.000 |
| Constant | 90.469*** | 0.059 | 0.000 |
| Remarks: *** p <0.01, ** p <0.05, * p<0.1 | | | |

Based on the regression results in Table 5, we could notice that *V* at medium and high levels as well as their interaction terms with *R* at medium and high levels (highlighted in green) are all statistically significant with *p*-value less than 0.01 and a much greater magnitude of coefficient. This indicates that *R* and *V* as well as their interaction terms are significant factors which may influence the end-of-evolution magnitude of change in process. Since the coefficients of all these terms are positive and the magnitude increases from medium to high level of *V* or *R*, we can conclude that increasing *R* and increasing *V* can generate greater efficiency gain by the end of routine evolution under the influence of RPA implementation. Besides, *L* is also a statistically significant parameter because the *p*-value of *L* at medium and high levels is less than 0.01 (highlighted in orange). Since the coefficient of *High L* is -9.046 which is lower than the coefficient of *Medium L* being -2.380, it validates that decreasing *L* is correlated with higher efficiency gain. By observing the interaction terms between *F* and *High R* (highlighted in red), the interaction terms all have *p*-value less than 0.01, which suggests that interaction terms between *F* and *High R* are statistically significant and *F* and *High R* altogether have a significant impact on efficiency gain at the end of routine evolution. As the coefficient of *High R* × *Medium F* is 0.213 which is smaller than the coefficient of *High R* × *High F* being 0.910, it is testified that at a high level of *R*, increasing *F* will increase the end-of-evolution efficiency gain. Besides, looking into the interaction terms between *F* and *High L* (highlighted in purple), we can observe that the coefficient of *Medium F* × *High L* is 0.309 which is much smaller than the coefficient of *High F* × *High L* being 3.799, also that both coefficients have *p*-value less and 0.01 and thus being statistically significant. We can confirm that at *High L*, increasing *F* will lead to greater efficiency gain at the end of routine evolution.

## 5.2 Result Interpretation for RPA Application Strategy

Based on the analysis of simulation results as above, we could advise on how organisations could better implement RPA to achieve higher organisational efficiency gain and meanwhile lead the transformation of digitised processes in organisations.

The most important benefit and motivation of implementing RPA are to enhance the efficiency gain in organisational operations via letting robots substitute manual effort to perform repetitive processes. Based on the simulation results, we find that the length of RPA sequence and occurrence frequency threshold can indeed influence the extent of efficiency gain through RPA. Automating shorter sequences can bring higher efficiency gains. Also, automating those sequences which occur very frequently can maximise the efficiency gain reaped from RPA. These add more valued insights when people choose and determine suitable RPA candidates based on the criteria. Organisations could establish specific criteria to identify those simple and short processes which occur with a high enough frequency and only automate those routines using RPA. In addition, increasing retention and increasing variation both give rise to greater efficiency gain. Thus, to take the efficiency gain to the next level, organisations are recommended to provide sufficient and appropriate reference and information to historical ways of process execution, so that process evolves by learning from the past while generating innovative ways of getting the task done. To increase variation, organisations could perform analysis and further simplification of the manual processes before implementing RPA.

Implementing RPA on business processes may decrease some extent of variation present due to fewer new edges formed in the network, which reduces the process complexity and lowers the likelihood of process drift and phase change. However, a process with RPA deployed can still experience a complexity burst and settle into a new dominant path. From this, we can state that even though implementing RPA may turn some sequences in a process into deterministic programmes, the overall business process can still drift and end up with a stable new state. Thus, automation tools, such as RPA, still preserve some degree of process flexibility and adaptability and maintain some potential for transformation and innovation of digitised processes in organisations.

The magnitude of change in the network structure is the prior condition for a process to undergo a major transformation and form a new dominant path over process evolution. Based on the observations regarding the magnitude of change. Increasing variation can drive a higher magnitude of change in network structure and a shorter length of RPA sequences can lead to a greater magnitude of change in network structure. Thus, to stimulate more vigorous process transformation or innovation, organisations should provide more opportunities for new workarounds and process simplification by business analysts before deploying RPA. Also, choosing those short sequences of actions for RPA can better enhance the magnitude of change in the process structure, which creates more room for process transformation and innovation in organisations.

# 6 Conclusion

## 6.1 Summary

Through simulation results, we identify and highlight the dynamic effects of RPA on routine evolution over time in terms of process complexity, the probability of phase change, the magnitude of change in process and efficiency gain. Based on the observations and analysis, we could generalise some theories to help organisations better implement RPA and selecting high-quality sequences for automation to further boost organisational operation efficiency, which may potentially give rise to transformation and innovation of digitised processes in organisations. Summarised key findings about RPA deployment are stated in Table 6 below.

| **Table 6. Summary of Key Findings about RPA Implementation** | |
|---|---|
| Finding 1 | To enhance efficiency gain, we could select shorter sequences of actions *L* with a relatively higher occurrence frequency *F* as higher-quality candidates for automation using RPA. |
| Finding 2 | To further boost the efficiency increase, we may need to increase retention *R* and enhance variation *V* by carefully performing pre-automation analysis via further simplifying the manual processes before implementing RPA. |
| Finding 3 | Even though implementing RPA will reduce process flexibility and make organisations less adaptive, we could preserve the potential of process flexibility for transformation and digital innovation which are related to higher drift probability and greater magnitude of change by maintaining a higher retention *R* and implementing RPA on shorter sequences of actions *L*. |

## 6.2 Contributions

The model in this research paper is a good example of how to modify and extend Pentland's model which simulates the effects of digitalisation on processes and routines (Pentland et al., 2020). Our model extends the original model and adopts some innovation changes which introduce the effects of RPA and enable RPA to interact and influence the existing effects from the original generative model.

RPA is only one example of automation tools applied widely in organisations. The impacts on business processes from other automation technology or adaptive systems, such as artificial intelligence, can also be analysed using the model with some adjustments based on the nature and characteristics of the technology.

The conclusions obtained from the results of this study can offer some insights on how to implement RPA in a way of maximising operational efficiency while preserving the potential for future process transformation and innovation. The practical application of this framework and these insights can provide significant managerial benefits, particularly for automation technology managers within organisations.

## 6.3 Limitations and Future Work

Due to the limitation of time and computational power, the current simulation model only explores three values for each parameter, where each represents a categorical level, and qualitative conclusions are drawn. To further substantiate the conclusions, the model can be further applied to more values of parameters and more quantitative results can be generated and concluded.

Also, there are a few assumptions stated in the model where future work can be done to enhance the model robustness. First, in the simulation model, RPA is applied systematically, which means at every process iteration, there is only one sequence of actions automated by RPA. The observations may change if there are multiple candidates qualified for RPA criteria which will be automated by RPA altogether at a single process interaction. Second, in this study, we assume that once a sequence of actions e.g. Node 3 → Node 4 → Node 7 is automated by RPA and there is an incoming edge to Node 3, the path from Node 3 to Node 4 and finally to Node 7 is deterministic and fixed. Thus, there is no way for having any emerging edge from Node 4 to other nodes in the network in the future. In other words, there will be no potential RPA starting from Node 4 to other nodes. However, it is possible in reality to get the intermediate actions in an automated sequence involved in another RPA sequence. Some actions can be reused as part of a new RPA sequence to form a new and more efficient path. The current model can be adjusted by creating a copy of the intermediate node (e.g. Node 4) to get invoked independently later with further RPA implementation. Third, in the current simulation model, the original simple happy path is removed after RPA is implemented. If the simple happy path is maintained, the value of drift probability will differ, which may make the frequency threshold and length of RPA sequence significant factors affecting the probability of phase change. These are points worth further exploration by modifying the model.

References


Aguirre, S., and Rodriguez, A. 2017. "Automation of a Business Process Using Robotic Process Automation (RPA): A Case Study," Communications in Computer and Information Science, 65–71. https://doi.org/10.1007/978-3-319-66963-2_7

Alter, S. 2014. "Theory of Workarounds," Communications of the Association for Information Systems (34), pp. 1041-1066.

Anderson, C., and Robey, D. 2017. "Affordance Potency: Explaining the Actualization of Technology Affordances," Information and Organization (27:2), pp. 100-115.

Argote, L. 2013. Organizational Learning, Boston: Springer.

Bak, P. 1996. "How Nature Works: The Science of Self-Organized Criticality," Nature (383:6603), pp. 772-773.

Berente, N., Lyytinen, K., Yoo, Y., and King, J. L. 2016."Routines as Shock Absorbers During Organizational Transformation. Integration, Control, and NASA's Enterprise Information System," Organization Science (27:3), pp. 551-572.

Bickhard, M. H., and Campbell, D. T. 2003. "Variations in Variation and Selection: the Ubiquity of the Variation-and- Selective-Retention Ratchet in Emergent Organizational Complexity," Foundations of Science (8:3), pp. 215-82.

Bose, R. P. J. C., van der Aalst, W. M. P., Žliobaitë, I., and Pechenizkiy, M. 2011. "Handling Concept Drift in Process Mining," in Advanced Information Systems Engineering. CAiSE 2011. Lecture Notes in Computer Science (Vol. 6741), H. Mouratidis and C. Rolland (eds.), Berlin: Springer, pp. 391-405

Brown, J & Duguid, Paul. (2000). Balancing Act: How to Capture Knowledge Without Killing it. Harvard business review. 78. 73-80, 212.

Ciborra, C. U., Braa, K., Cordella, A., Dahlbom, B., Failla, A., Hanseth, O., Hepsø, V., Ljungberg, J., Monteiro, E., and Simon, K. A. 2000. From Control to Drift: The Dynamics of Corporate Information Infrastructures, Oxford, UK: Oxford University Press.

Darr, E. D., Argote, L., and Epple, D. 1995. "The Acquisition, Transfer, and Depreciation of Knowledge in Service Organi- zations: Productivity in Franchises," Management Science (41:11), pp. 1750-1762.

Davis, J. P., Eisenhardt, K. M., and Bingham, C. B. 2007. “Developing Theory through Simulation Methods,” Academy of Management Review (32:2), pp. 480-499.

Feldman, M. S., and Pentland, B. T. 2003. “Reconceptualizing Organizational Routines as a Source of Flexibility and Change,” Administrative Science Quarterly (48:1), pp. 94-118.

Foster, J. 1997. “The Analytical Foundations of Evolutionary Economics: From Biological Analogy to Economic Self- Organization,” Structural Change and Economic Dynamics (8:4), pp. 427-451.

Frigg, R. 2003. “Self-Organised Criticality—What it Is and What it Isn’t,” Studies in History and Philosophy of Science Part A (34:3), pp. 613-632.

Goh, Kenneth & Pentland, Brian. (2019). From Actions to Paths to Patterning: Toward a Dynamic Theory of Patterning in Routines. The Academy of Management Journal. 62. 1901-1929. 10.5465/amj.2018.0042.

Hærem, T., Pentland, B. T., and Miller, K. D. 2015. “Task Com- plexity: Extending a Core Concept,” Academy of Management Review (40:3), pp. 446-460.

Holan, P. M. D., and Phillips, N. 2004. “Remembrance of Things Past? The Dynamics of Organizational Forgetting,” Management Science (50:11), pp. 1603-1613.

Howard-Grenville, J., & Rerup, C. 2017. A Process Perspective on Organizational Routines. The SAGE Handbook of Process Organization Studies, 323–337. https://doi.org/10.4135/9781473957954.n20

Javaid, Adeel. (2013). Understanding Dijkstra Algorithm. SSRN Electronic Journal. 10.2139/ssrn.2340905.

McCabe, T.J. 1976. A Complexity Measure. IEEE Transactions on Software Engineering, SE-2, 308-320.

Pentland, B., Recker, J., and Wyner, G. 2015. “A Thermometer for Interdependence: Exploring Patterns of Interdependence Using Networks of Affordances,” in Proceedings of the 36th International Conference on Information Systems, Fort Worth, TX.

Pentland, B. T., Feldman, M. S., Becker, M. C., and Liu, P. 2012. “Dynamics of Organizational Routines: A Generative Model,” Journal of Management Studies (49:8), pp. 1484-1508.

Pentland, B. T., Liu, P., Kremser, W., & Haerem, T. 2020. “The Dynamics of Drift in Digitized Processes,” MIS Quarterly, 44(1), 19–47. https://doi.org/10.25300/misq/2020/14458

Pentland, B. T., Yoo, Y., Recker, J., & Kim, I. 2022. “From Lock-in to Transformation: A Path-centric Theory of Emerging Technology and Organizing,” Organization Science, 33(1), 194–211. https://doi.org/10.1287/orsc.2021.1543

Simon, H. A. 1996. The Sciences of the Artificial (3rd ed.), Cambridge, MA: MIT Press.

Strong, D. M., Volkoff, O., Johnson, S. A., Pelletier, L., R., Tulu, B., Isa, B.-O., Trudel, J., and Garber, L. 2014. “A Theory of Organization–EHR Affordance Actualization,” Journal of the Association for Information System (15:2), pp. 53-85.

Syed, R., Suriadi, S., Adams, M., Bandara, W., Leemans, S. J. J., Ouyang, C., ter Hofstede, A. H. M., van de Weerd, I., Wynn, M. T., & Reijers, H. A. (2020). Robotic Process Automation: Contemporary themes and challenges. Computers in Industry, 115, 103162. https://doi.org/10.1016/j.compind.2019.103162

Van der Aalst, W. M., Bichler, M., & Heinzl, A. 2018. “Robotic Process Automation,” Business & Information Systems Engineering, 60(4), 269–272. https://doi.org/10.1007/s12599-018-0542-4

Wellmann, C., Stierle, M., Dunzer, S., & Matzner, M. 2020. “A Framework to Evaluate the Viability of Robotic Process Automation for Business Process Activities,” Lecture Notes in Business Information Processing, 200–214. https://doi.org/10.1007/978-3-030-58779-6_14

Yoo, Y. 2010. “Computing in Everyday Life: A Call for Research on Experiential Computing,” MIS Quarterly (34:2), pp. 213-231.