Mixture Models Applied to Failure Times Modeling: An Exploratory Study in Software Reliability

Evelyn Natalie Aguiar de Almeida

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2025

Evelyn Natalie Aguiar de Almeida

Mixture Models Applied to Failure Times Modeling: An Exploratory Study in Software Reliability

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Rivalino Matias Júnior

Uberlândia
2025

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

A447 Almeida, Evelyn Natalie Aguiar de, 1997-
2025 Mixture models applied to failure times modeling [recurso eletrônico] : an exploratory study in software reliability / Evelyn Natalie Aguiar de Almeida. - 2025.

Orientador: Rivalino Matias Júnior.
Dissertação (Mestrado) - Universidade Federal de Uberlândia, Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
DOI http://doi.org/10.14393/ufu.di.2025.714
Inclui bibliografia.
Inclui ilustrações.

1. Computação. I. Matias Júnior, Rivalino ,1971-, (Orient.). II. Universidade Federal de Uberlândia. Pós-graduação em Ciência da Computação. III. Título.

CDU: 681.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Computação
Av. João Naves de Ávila, 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpgfacom@ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Dissertação, 49/2025, PPGCO				
Data:	15 de Dezembro de 2025	Hora de início:	16:05	Hora de encerramento:	17:55
Matrícula do Discente:	12412CCP007				
Nome do Discente:	Evelyn Natalie Aguiar de Almeida				
Título do Trabalho:	Mixture Models Applied to Failure Times Modeling: An Exploratory Study in Software Reliability				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Engenharia de Software				
Projeto de Pesquisa de vinculação:	----------				

Reuniu-se por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Stéphane Julia - FACOM/UFU, Paulo Romero Martins Maciel - UFPE, Lance Fiondella - University of Massachusetts Dartmouth e Rivalino Matias Junior - FACOM/UFU, orientador do(a) candidato(a).

Os examinadores participaram desde as seguintes localidades: Paulo Romero Martins Maciel - Recife/PE, Lance Fiondella - Dartmouth Massachusetts/EUA. Os outros membros da banca e o aluno(a) participaram da cidade de Uberlândia.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Rivalino Matias Junior, apresentou a Comissão Examinadora e o(á) candidato(a), agradeceu a presença do público, e concedeu ao(á) Discente a palavra para a exposição do seu trabalho

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o(á) candidato(a). Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato(a):

Aprovado

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

Ressalta-se que o examinador Lance Fiondella, por ser estrangeiro, residente em outro país e não possuir CPF registrado no Brasil não assinará a ata de defesa.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Rivalino Matias Júnior**, **Professor(a) do Magistério Superior**, em 17/12/2025, às 14:08, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Stéphane Julia**, **Professor(a) do Magistério Superior**, em 18/12/2025, às 11:19, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Paulo Romero Martins Maciel**, **Usuário Externo**, em 19/12/2025, às 16:10, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **6923425** e o código CRC **F5EE8423**.

Referência: Processo nº 23117.087253/2025-34 SEI nº 6923425

Este trabalho é dedicado à minha família, aos meus pais Odenir e Rosilene.

Agradecimentos

Primeiramente, agradeço a Deus por me conceder saúde, força e perseverança ao longo de toda esta jornada, especialmente nos momentos de dificuldade, nos quais a fé foi essencial para seguir em frente.

Aos meus pais, expresso minha mais profunda gratidão pelo amor incondicional e pelo apoio constante que foram fundamentais para a minha formação pessoal e acadêmica. À minha família, agradeço pelo incentivo contínuo, pela compreensão e pelo suporte emocional ao longo de todo o período do mestrado.

Ao meu orientador, Prof. Rivalino Matias Júnior, manifesto minha gratidão pelo incentivo, dedicação e paciência demonstrados durante todas as etapas deste trabalho. Sua orientação e conhecimento técnico foram essenciais para o desenvolvimento desta pesquisa e para o meu crescimento acadêmico.

Agradeço também aos professores da Faculdade de Computação com os quais tive a oportunidade de ser discente pelo conhecimento compartilhado e pela formação sólida proporcionada ao longo do curso.

Por fim, agradeço a todos que, direta ou indiretamente, contribuíram para a realização deste trabalho.

"A grandeza de uma pessoa está em saber reconhecer sua própria pequenez."
Blaise Pascal

Resumo

Na literatura de confiabilidade de software, em especial na modelagem de tempos de falha, predominam modelos probabilísticos baseados em distribuição única. No entanto, tais modelos mostram-se limitados diante da complexidade de softwares modernos, caracterizados por múltiplas causas de falha e pela variabilidade em seus perfis operacionais. Embora amplamente utilizados nas fases de desenvolvimento e de teste, os modelos de distribuição única apresentam desempenho insatisfatório em contextos reais de operação, devido às interações com o ambiente operacional, que resultam em uma maior diversidade de padrões de falha. Este trabalho investiga a aplicação de modelos de mistura à análise de confiabilidade de software com base em dados reais de operação. Essa abordagem permite representar padrões de tempos de falha intra e intergrupos, considerando grupos como aproximações de diferentes perfis operacionais, bem como a relação entre distintas causas de falha. Os resultados obtidos indicam que os modelos de mistura superam os modelos probabilísticos de distribuição única ao modelar a variabilidade dos dados de falha em ambientes reais, além de capturar a influência dos diferentes perfis operacionais sobre os tempos de falha.

Palavras-chave: Confiabilidade de software; modelos de mistura; causas de falha; modelos probabilísticos; tempos de falha.

Mixture Models Applied to Failure Times Modeling: An Exploratory Study in Software Reliability

Evelyn Natalie Aguiar de Almeida

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2025

Abstract

In the software reliability literature, particularly in the modeling of failure times, probabilistic models based on a single distribution are predominant. However, such models are limited when dealing with the complexity of modern software systems, which are characterized by multiple failure causes and variability in operational profiles. Although widely used during development and testing phases, single-distribution models often perform poorly in real operational contexts due to interactions with the operational environment, which lead to a greater diversity of failure patterns. This study investigates the application of mixture models to software reliability analysis based on real operational data. This approach allows representing intra- and inter-group patterns of time between failures, considering groups as approximations of different operational profiles, as well as the relationship between distinct failure causes. The results indicate that mixture models outperform single-distribution probabilistic models by modeling the variability of failure data in real environments and by accounting for the influence of different operational profiles on failure times.

Keywords: Software reliability; mixture models; failure causes; probabilistic models; failure times.

List of Figures

Figure 1 – Goodness-of-Fit Analysis for Exponential Distribution: Histogram with Theoretical PDF Overlay, Empirical vs. Theoretical CDF Comparison, P-P Plot, and Q-Q Plot. 36
Figure 2 – Relationship Between Failure Causes, Operational Profiles, Execution Environments and Failure Modes. 45
Figure 3 – Ridgeline Plot Comparing the Distribution of Failures per Computer Across Four Groups (G1–G4). 47
Figure 4 – Ridgeline Plot Comparing the Distribution of Days Sampled per Computer Across Four Groups (G1–G4). 48
Figure 5 – Ridgeline Plot Comparing the Distribution of Application Failures per Computer Across Four Groups (G1–G4). 49
Figure 6 – Ridgeline Plot Comparing the Distribution of Failure Causes per Application per Computer Across Four Groups (G1–G4). 49
Figure 7 – Cluster Evaluation Plots for the Sample G1Rac_MC0-018_iexplore_c0000005. 68
Figure 8 – G1Rac_MC0-018_iexplore_c0000005 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 69
Figure 9 – G1Rac_MC0-018_iexplore_c0000005 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components. . . . 72
Figure 10 – G1Rac_MC0-018_iexplore_c0000005 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 74
Figure 11 – G1Rac_MC0-018_iexplore_c0000005 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 75
Figure 12 – G1Rac_MC0-018_iexplore_c0000005 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 76
Figure 13 – Cluster Evaluation Plots for the Sample G2Rac_MC1-074. 97

Figure 14 – G2Rac_MC1-074 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 98
Figure 15 – G2Rac_MC1-074 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components. 101
Figure 16 – G2Rac_MC1-074 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 103
Figure 17 – G2Rac_MC1-074 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 104
Figure 18 – G2Rac_MC1-074 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 105
Figure 19 – Cluster Evaluation Plots for the Sample G2Rac_MC1-074_iexplore_c0000005. 108
Figure 20 – G2Rac_MC1-074_iexplore_c0000005 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 109
Figure 21 – G2Rac_MC1-074_iexplore_c0000005 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components. . . . 111
Figure 22 – G2Rac_MC1-074_iexplore_c0000005 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 113
Figure 23 – G2Rac_MC1-074_iexplore_c0000005 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 114
Figure 24 – G2Rac_MC1-074_iexplore_c0000005 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 115
Figure 25 – Cluster Evaluation Plots for the Sample G2Rac_MC1-074_iexplore_c0000096. 118
Figure 26 – G2Rac_MC1-074_iexplore_c0000096 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 119
Figure 27 – G2Rac_MC1-074_iexplore_c0000096 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components. . . . 121
Figure 28 – G2Rac_MC1-074_iexplore_c0000096 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 123
Figure 29 – G2Rac_MC1-074_iexplore_c0000096 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 125
Figure 30 – G2Rac_MC1-074_iexplore_c0000096 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 126
Figure 31 – Cluster Evaluation Plots for the Sample G2Rac_MC-157400. 129

Figure 32 – G2Rac_MC-157400 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 130
Figure 33 – G2Rac_MC-157400 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components. 133
Figure 34 – G2Rac_MC-157400 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 135
Figure 35 – G2Rac_MC-157400 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 136
Figure 36 – G2Rac_MC-157400 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 137
Figure 37 – Cluster Evaluation Plots: G2Rac_MC-157400_iexplore_c0000374. 140
Figure 38 – G2Rac_MC-157400_iexplore_c0000374 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 141
Figure 39 – G2Rac_MC-157400_iexplore_c0000374 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components. . . . 143
Figure 40 – G2Rac_MC-157400_iexplore_c0000374 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 145
Figure 41 – G2Rac_MC-157400_iexplore_c0000374 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 146
Figure 42 – G2Rac_MC-157400_iexplore_c0000374 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 147
Figure 43 – Cluster Evaluation Plots for the Sample G3Rac_DSK023. 150
Figure 44 – G3Rac_DSK023 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 151
Figure 45 – G3Rac_DSK023 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components. 154
Figure 46 – G3Rac_DSK023 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 156
Figure 47 – G3Rac_DSK023 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 157
Figure 48 – G3Rac_DSK023 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 158
Figure 49 – Cluster Evaluation Plots for the Sample G3Rac_DSK023_iexplore_c0000005. 161

Figure 50 – G3Rac_DSK023_iexplore_c0000005 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 162
Figure 51 – G3Rac_DSK023_iexplore_c0000005 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components. 165
Figure 52 – G3Rac_DSK023_iexplore_c0000005 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 167
Figure 53 – G3Rac_DSK023_iexplore_c0000005 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 168
Figure 54 – G3Rac_DSK023_iexplore_c0000005 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 169
Figure 55 – Cluster Evaluation Plots: G413452266836854502e87bba75907480. 172
Figure 56 – G413452266836854502e87bba75907480 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 173
Figure 57 – G413452266836854502e87bba75907480 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components. . . . 176
Figure 58 – G413452266836854502e87bba75907480 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 178
Figure 59 – G413452266836854502e87bba75907480 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 179
Figure 60 – G413452266836854502e87bba75907480 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 180
Figure 61 – Cluster Plots: G41345226...bba75907480_iexplore_e06d7363. 183
Figure 62 – G41345226...bba75907480_iexplore_e06d7363 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 184
Figure 63 – G41345226...e87bba75907480_iexplore_e06d7363 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components. . 186
Figure 64 – G41345226...bba75907480_iexplore_e06d7363 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots. 188
Figure 65 – G41345226...bba75907480_iexplore_e06d7363 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 189
Figure 66 – G41345226...bba75907480_iexplore_e06d7363 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations. 190

List of Tables

Table 1 – Goodness-of-fit Test Results for Exponential Distribution. 37
Table 2 – Comparison of Related Studies on Software Reliability. 39
Table 3 – Estimated Computational Complexity for Each Distribution. 57
Table 4 – Number and Proportion of Zero-Valued TBFs in Each Group. 63
Table 5 – The Selected Computers for Approach 1 Analysis. 63
Table 6 – The Selected Samples for Approach 2 Analysis (Internet Explorer). 65
Table 7 – Internet Explorer Failure Causes and Their Classification. 66
Table 8 – Distribution of Internet Explorer Failure Causes. 66
Table 9 – Representative Samples Selected for Detailed Analysis. 66
Table 10 – Descriptive Statistics: G1Rac_MC0-018_iexplore_c0000005. 67
Table 11 – Cluster Results: G1Rac_MC0-018_iexplore_c0000005. 67
Table 12 – Goodness-of-fit Test Results: G1Rac_MC0-018_iexplore_c0000005. 70
Table 13 – Mixture Model Parameters: G1Rac_MC0-018_iexplore_c0000005. 71
Table 14 – Mixture vs. Single Distributions: G1Rac_MC0-018_iexplore_c0000005. . . . 73
Table 15 – Best Mixture Models and Goodness-of-Fit Tests by Dataset. 77
Table 16 – Statistical Characteristics by Computer Group. 78
Table 17 – Clustering Method Performance Comparison. 79
Table 18 – Distribution Preferences by Goodness-of-Fit Criteria. 79
Table 19 – Mixture Model Components by Failure Type. 80
Table 20 – Descriptive Statistics: G2Rac_MC1-074. 95
Table 21 – Cluster Results: G2Rac_MC1-074. 96
Table 22 – Goodness-of-fit Test Results: G2Rac_MC1-074. 99
Table 23 – Mixture Model Parameters: G2Rac_MC1-074. 99
Table 24 – Mixture vs. Single Distributions: G2Rac_MC1-074. 102
Table 25 – Descriptive Statistics: G2Rac_MC1-074_iexplore_c0000005. 106
Table 26 – Cluster Results: G2Rac_MC1-074_iexplore_c0000005. 107
Table 27 – Goodness-of-fit Test Results: G2Rac_MC1-074_iexplore_c0000005. 110
Table 28 – Mixture Model Parameters: G2Rac_MC1-074_iexplore_c0000005. 110

Table 29 – Mixture vs. Single Distributions: G2Rac_MC1-074_iexplore_c0000005. . . . 112
Table 30 – Descriptive Statistics: G2Rac_MC1-074_iexplore_c0000096. 117
Table 31 – Cluster Results: G2Rac_MC1-074_iexplore_c0000096. 117
Table 32 – Goodness-of-fit Test Results: G2Rac_MC1-074_iexplore_c0000096. 120
Table 33 – Mixture Model Parameters: G2Rac_MC1-074_iexplore_c0000096. 120
Table 34 – Mixture vs. Single Distributions: G2Rac_MC1-074_iexplore_c0000096. . . . 122
Table 35 – Descriptive Statistics: G2Rac_MC-157400. 127
Table 36 – Cluster Results: G2Rac_MC-157400. 128
Table 37 – Goodness-of-fit Test Results: G2Rac_MC-157400. 131
Table 38 – Mixture Model Parameters: G2Rac_MC-157400. 131
Table 39 – Mixture vs. Single Distributions: G2Rac_MC-157400. 134
Table 40 – Descriptive Statistics: G2Rac_MC-157400_iexplore_c0000374. 139
Table 41 – Cluster Results: G2Rac_MC-157400_iexplore_c0000374. 139
Table 42 – Goodness-of-fit Test Results: G2Rac_MC-157400_iexplore_c0000374. . . . 142
Table 43 – Mixture Model Parameters: G2Rac_MC-157400_iexplore_c0000374. 142
Table 44 – Mixture vs. Single Distributions: G2Rac_MC-157400_iexplore_c0000374. . . 144
Table 45 – Descriptive Statistics: G3Rac_DSK023. 148
Table 46 – Cluster Results: G3Rac_DSK023. 149
Table 47 – Goodness-of-fit Test Results: G3Rac_DSK023. 152
Table 48 – Mixture Model Parameters: G3Rac_DSK023. 152
Table 49 – Mixture vs. Single Distributions: G3Rac_DSK023. 154
Table 50 – Descriptive Statistics: G3Rac_DSK023_iexplore_c0000005. 160
Table 51 – Cluster Results: G3Rac_DSK023_iexplore_c0000005. 160
Table 52 – Goodness-of-fit Test Results: G3Rac_DSK023_iexplore_c0000005. 163
Table 53 – Mixture Model Parameters: G3Rac_DSK023_iexplore_c0000005. 163
Table 54 – Mixture vs. Single Distributions: G3Rac_DSK023_iexplore_c0000005. . . . 166
Table 55 – Descriptive Statistics: G413452266836854502e87bba75907480. 171
Table 56 – Cluster Results: G413452266836854502e87bba75907480. 171
Table 57 – Goodness-of-fit Test Results: G413452266836854502e87bba75907480. . . . 174
Table 58 – Mixture Model Parameters: G413452266836854502e87bba75907480. 174
Table 59 – Mixture vs. Single Distributions: G413452266836854502e87bba75907480. . . 177
Table 60 – Descriptive Statistics: G41345226...bba75907480_iexplore_e06d7363. 182
Table 61 – Cluster Results: G41345226...bba75907480_iexplore_e06d7363. 182
Table 62 – Goodness-of-fit Test Results: G41345226...bba75907480_iexplore_e06d7363. . 185
Table 63 – Mixture Model Parameters: G41345226...bba75907480_iexplore_e06d7363. . 185
Table 64 – Mixture vs. Single Distributions: G41...e87bba75907480_iexplore_e06d7363. 187

Listings

List of Algorithms

1 Multimodality and Cluster Analysis Algorithm 54
2 Expectation-Maximization Algorithm 56

Acronyms list

TBF Time Between Failures

RAC Reliability Analysis Component

AIC Akaike Information Criterion

BIC Bayesian Information Criterion

KS Kolmogorov-Smirnov Test

AD Anderson-Darling Test

Q-Q Plot Quantile-Quantile Plot

P-P Plot Probability-Probability Plot

PDF Probability Density Function

CDF Cumulative Distribution Function

GMM Gaussian Mixture Model

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise

Symbols List

α Shape parameter of gamma distribution

$\arg\max_{\theta \in \Omega}$ Argument that maximizes the function over parameter space Ω

$\arg\max_{\theta}$ Argument that maximizes the function with respect to θ

β Rate parameter of gamma distribution

$\exp(\cdot)$ Exponential function

$\Gamma(\alpha)$ Gamma function

$\gamma(\alpha, \beta x)$ Lower incomplete gamma function

γ_i Responsibility weight for observation i

$\hat{\lambda}$ Maximum likelihood estimate of rate parameter

$\hat{\mu}$ Maximum likelihood estimate of mean parameter

$\hat{\sigma}^2$ Maximum likelihood estimate of variance parameter

λ Scale parameter

$\ln x$ Natural logarithm of x

$\log(\cdot)$ Natural logarithm function

$\log(\text{likelihood})$ Log-likelihood function

μ Mean parameter of normal distribution

Ω Parameter space

$\Phi(\cdot)$ Standard normal cumulative distribution function

π Mathematical constant (approximately 3.14159)

π_i	Weight of the i-th component in mixture model
σ	Standard deviation parameter
σ^2	Variance parameter
$\sqrt{\cdot}$	Square root function
$\sum_{i=1}^{m}$	Summation over i from 1 to m
$\sum_{i=1}^{N}$	Summation over i from 1 to N
$\sum_{i=1}^{n}$	Summation over i from 1 to n
$\sum_{j=1}^{K}$	Summation over j from 1 to K
$\text{erf}(\cdot)$	Error function
θ	Parameters to be estimated
$\theta^{(m)}$	Parameter estimates at iteration m
$\theta^{(m+1)}$	Updated parameter estimates at iteration $m+1$
θ_j	Parameters of component j
θ_k	Parameters of component k
$\theta_k^{(t+1)}$	Updated parameters of component k at iteration $t+1$
$f(t)$	General probability density function
$F(x)$	General cumulative distribution function
$f_1(t)$	Probability density function of the first component
$f_2(t)$	Probability density function of the second component
$F_j(t)$	Cumulative distribution function of the j-th subpopulation
$f_j(t)$	Probability density function of the j-th subpopulation
$G(t)$	Cumulative distribution function of the overall mixture model
$g(t)$	Probability density function of a mixture model
i	Index for components ($i = 1, 2, ..., m$)
j	Index for subpopulations ($j = 1, 2, ..., K$)
K	Number of model parameters or components

k Rate parameter

m Number of components in a mixture model

N Total number of data points

n Number of observations

$P(X \leq x)$ Probability that random variable X is less than or equal to x

p_j Mixing probability (weight) of the j-th subpopulation

$R(t)$ Reliability function

$R_i(t)$ Reliability function of the i-th component

r_{ik} Responsibility of component k for observation i

$r_{ik}^{(t)}$ Responsibility of component k for observation i at iteration t

w Weight parameter in two-component mixture model

w_j Weight of component j

w_k Weight (mixing proportion) of component k

$w_k^{(t+1)}$ Updated weight of component k at iteration $t+1$

X Complete data (observed plus missing data)

x General random variable value

x_i i-th observation in the dataset

Y Observed (incomplete) data

y Observed data value

Contents

1 INTRODUCTION . . . 17
1.1 Contextualization . . . 17
1.2 Research Relevance . . . 18
1.3 Research Goals . . . 19
1.3.1 General . . . 19
1.3.2 Specific . . . 19
1.3.3 Research Development . . . 20
1.3.4 Literature Review . . . 20
1.3.5 Material . . . 20
1.3.6 Methods . . . 20
1.4 Document Structure . . . 21

2 LITERATURE REVIEW . . . 23
2.1 Introduction . . . 23
2.2 Software Reliability . . . 23
2.3 Mixture Models . . . 25
2.4 Clustering Methods . . . 29
2.4.1 K-Means Clustering . . . 30
2.5 Expectation-Maximization (EM) Algorithm . . . 31
2.6 Goodness-of-Fit Tests and Graphical Analysis . . . 33

3 RELATED WORKS . . . 38

4 METHODOLOGY . . . 44
4.1 Data Description . . . 44
4.1.1 Parameters and Failure Characterization . . . 46
4.1.2 Dataset Characterization . . . 46
4.2 Modeling Approach . . . 50

4.3 Cluster and Statistical Analysis . . . 52
4.3.1 Cluster Evaluation Metrics . . . 53
4.3.2 Evaluation and Selection of Clustering Configurations . . . 53
4.4 Expectation-Maximization Algorithm . . . 54
4.4.1 Computational Complexity Analysis . . . 55
4.4.2 Initialization Strategy . . . 58
4.4.3 EM Algorithm Implementation . . . 58
4.4.4 Incremental Components for Mixture Model . . . 59
4.4.5 Convergence and Optimization Strategies . . . 60
4.4.6 Sensitivity Analysis and Computational Cost . . . 60

5 RESULTS . . . 62
5.0.1 Sample Selection and Data Preparation . . . 62
5.0.2 Occurrence of Zero-Valued TBFs . . . 62
5.0.3 Approach 1: Complete TBF Analysis . . . 63
5.0.4 Approach 2: Internet Explorer Failure Analysis . . . 64
5.0.5 Sample Analysis . . . 64
5.0.6 G1Rac_MC0-018_iexplore_c0000005 (Approach 2) . . . 65
5.1 Results Summary . . . 75
5.2 Results and Pattern Analysis . . . 78
5.2.1 Comparative Group Analysis . . . 78
5.2.2 Clustering Methodology Comparison . . . 79
5.2.3 Distributional Analysis and Goodness-of-Fit Patterns . . . 79
5.2.4 Failure Type Characterization . . . 80
5.2.5 Proposed Statistical Classification Algorithm . . . 80

6 CONCLUSION . . . 83
6.1 Key Research Findings . . . 83
6.1.1 Operational Environment Impact . . . 84
6.1.2 Failure Pattern Characterization . . . 85
6.1.3 Multi-Modal Failure Behavior . . . 85
6.1.4 Distributional Preferences and Model Complexity . . . 86
6.2 Research Limitations . . . 86
6.3 Future Research Directions . . . 87

BIBLIOGRAPHY . . . 88

APPENDIX 94

APPENDIX A – RESULTS G2RAC-MC1-074 (APPROACH 1) . . . 95

APPENDIX B – G2RAC_MC1-074_IEXPLORE_C0000005 (APPROACH 2) 106

APPENDIX C – RESULTS G2RAC-MC1-074-IEXPLORE-C0000096 (APPROACH 2) 116

APPENDIX D – RESULTS G2RAC-MC-157400 (APPROACH 1) 127

APPENDIX E – RESULTS G2RAC-MC-157400-IEXPLORE-C0000374 (APPROACH 2) 138

APPENDIX F – RESULTS G3RAC-DSK023 (APPROACH 1) 148

APPENDIX G – RESULTS G3RAC-DSK023-IEXPLORE-C0000005 (APPROACH 2) 159

APPENDIX H – RESULTS G413452266836854502E87BBA75907480 (APPROACH 1) 170

APPENDIX I – RESULTS G413452266836854502E87BBA75907480-IEXPLORE-E06D7363 (APPROACH 2) 181

Chapter 1

Introduction

1.1 Contextualization

Software has become an essential component of modern society, deeply integrated into various human activities, ranging from routine tasks to highly complex operations. It is difficult to separate software from everyday life, as the automation and optimization it provides are crucial for the efficient functioning of multiple sectors. This growing dependence highlights the importance of software reliability, especially considering the severe consequences that failures can have. In such contexts, software failures can lead not only to significant financial losses but also pose risks to human safety and lives (MATIAS; OLIVEIRA; ARAUJO, 2013).

Software reliability is defined as the probability of software operating without failure for a specific period in a given environment (IEEE, 1990). Its definition is conceptually similar to hardware reliability but presents key differences, while hardware reliability can be quantified based on physical failure rates and component degradation over time, software failure is primarily associated with logical and programming faults.

Unlike hardware, software does not suffer from physical wear, instead, failures typically arise from bugs, design flaws, or improper user interactions (PHAM, 2007). Consequently, software reliability is not only about preventing failures, but also about ensuring when failures occur their impact is minimized and the system can recover swiftly and effectively. To achieve this, software engineers implement testing, continuous validation, and real-time monitoring techniques, among other practices.

One approach to software reliability is probabilistic modeling of failure events, which applies mathematical models, such as probability distributions, to describe software failure times behavior and estimate the likelihood of failures. The premise of this approach is to use historical failure data to predict the probability of future failures (MUSA, 2004) and enabling the implementation of preventive measures and mitigating potential issues in critical scenarios (PHAM, 2007).

Numerous analytical models have been proposed to address the quantification of software

reliability, most of these models estimate software reliability based on failure history, treating software as a black box (GOKHALE; TRIVEDI, 1999) and assuming a parametric model for either the time between failures or the number of failures over a finite time interval. Conversely, white-box models (PAI, 2013) estimate reliability by representing the structure of the software, focusing on the operational phase of the software life cycle.

A common assumption in most software reliability models is that failures occur as independent events (GOSEVA-POPSTOJANOVA; TRIVEDI, 2000). This assumption is made to simplify mathematical modeling and parameter estimation. However, these assumptions may not accurately represent actual failure occurrences, potentially leading to incorrect reliability estimations. To avoid assumptions that may not reflect real-world failure behavior, it is essential to understand how failures occur under real operating conditions. Therefore, failure data collected during software testing phases may be limited and may not fully represent all scenarios encountered in a real execution environment.

Furthermore, synthetic data is frequently employed in software reliability studies. Unlike real-world failure data collected from production environments, synthetic failure event data is artificially generated based on assumptions about system behavior (BUČAR; NAGODE; FAJDIGA, 2004; KUMAR; JAIN; GANGOPADHAY, 2021; WU et al., 2024). While synthetic data facilitates experimentation in controlled settings, it also has limitations: as models trained on synthetic data may not accurately reflect the real-world complexity and unpredictability of operational environments (BIRD et al., 2014).

Many traditional reliability models rely on the assumption that failures arise from a single underlying cause, and these single-failure-cause models assume homogeneous failure behavior across the system. However, this assumption is often unrealistic for modern software systems, which tend to experience failures due to diverse and interacting causes. To address this limitation, mixture models (MCLACHLAN; LEE; RATHNAYAKE, 2019) can be utilized, as they incorporate multiple probability distributions, and each can represent groups of distinct failure cause.

Despite their potential, there is a significant gap in the literature regarding the application of mixture models using real-world operational data for software reliability analysis. This research aims to explore the applicability of mixture models in software reliability analysis based on real failure data. The focus is on modeling the reliability of systems with multiple failure causes, given the complexity and heterogeneity of modern systems.

1.2 Research Relevance

This study contributes to the field of software reliability by exploring how mixture models can be effectively applied to model real-world Time Between Failures (TBF) data. While previous researches have largely relied on synthetic datasets or single-distribution models, this work adopts an empirical approach using real-world failure logs from heterogeneous comput-

ing environments. The goal is to offer a robust statistical representation of failure behavior in modern software systems.

Beyond assessing model fit, the research aims to uncover statistical patterns within TBF data that correspond to failure causes and operational conditions. By analyzing data from computer groups operating under varying workloads and usage contexts, the study examines how operational profiles influence the distributional characteristics of failures. Specifically, this work investigates whether mixture models can reveal hidden structures in the data, such as clusters of failures with similar statistical signatures, and how these relate to known failure causes.

1.3 Research Goals

1.3.1 General

The general goal of this research is to investigate whether mixture models can provide a more robust and flexible representation of software failure behavior, especially in systems exhibiting multiple failure causes and variability in operational usage.

1.3.2 Specific

To accomplish the general goal, the study is guided by the following specific objectives:

- ❑ **Evaluate whether mixture models provide a better fit for TBF data compared to single-distribution models:** This involves testing a wide range of homogeneous and heterogeneous mixture configurations using statistical criteria: Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Kolmogorov-Smirnov Test (KS), Anderson-Darling Test (AD) and graphical validation methods: Quantile-Quantile Plot (Q-Q Plot), Probability-Probability Plot (P-P Plot), Probability Density Function (PDF) and Cumulative Distribution Function (CDF).

- ❑ **Analyze the influence of operational profiles on TBF patterns:** By comparing computer groups with different workloads and usage contexts, the study will assess how operational characteristics affect the shape, scale, and composition of failure distributions.

- ❑ **Identify statistical patterns associated with different failure causes:** This includes determining whether specific causes of failure (access violation, heap corruption, privileged instruction violation, unhandled C++ exception) are associated with recurring distributional behaviors within the mixture models.

- ❑ **Determine the ability of mixture models to detect and represent failure subgroups in the data:** This study applies clustering algorithms (Gaussian Mixture Model

(GMM), Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), K-Means, Fuzzy C-Means) alongside mixture model analysis to uncover internal structure and potential failure subgroups within the datasets.

- ❑ **Identify the most suitable distribution combinations for modeling software failure data:** The study will map which distributions (Exponential, Normal, Lognormal, Weibull and Gamma) are most commonly selected in the best-fitting models.

1.3.3 Research Development

This research aims to explore and understand patterns of software failures, particularly in systems affected by multiple underlying failure causes. Emphasis is placed on identifying and characterizing distinct failure causes through the analysis of real-world failure data collected from software applications.

By analyzing failures occurring in actual operational environments, this research provides empirical insights into how heterogeneous failure behavior emerges in practice. The findings can enhance the understanding of software reliability under real conditions and demonstrate the capability of mixture models to capture the behavior of software failure data.

1.3.4 Literature Review

A literature review on software reliability was carried out to establish the theoretical foundation for this research. The search process aimed to identify relevant studies on the topic without imposing restrictions on publication dates. Key sources included books, peer-reviewed journal articles, and conference proceedings published in reputable scientific venues. These materials were accessed primarily through digital libraries and academic search engines, such as Google Scholar.

1.3.5 Material

The dataset comprises failure logs from applications running on Microsoft Windows 7 (Win7), gathered from four distinct workplaces. In total, 40,095 failure records were collected from 660 computers operating in various contexts, including academic institutions, corporate settings, and personal/home use. This diversity supports a comprehensive assessment of software failure patterns under different operational conditions.

A detailed description of the dataset is provided in Chapter 3.

1.3.6 Methods

This study applied statistical modeling using continuous probability distributions to estimate mixture models for TBF in order to account for heterogeneous failure behaviors through the composition of distribution models.

The dataset comprised real-world software failure records, including computer identifiers and timestamps of each occurrence, which enabled the calculation of TBFs used for reliability modeling (SANTOS; MATIAS, 2018). As a first step, outlier detection and removal techniques were applied to minimize the impact of extreme and invalid values on the analysis.

Following data processing, the Expectation-Maximization (EM) algorithm, as described in (MCLACHLAN; LEE; RATHNAYAKE, 2019), was applied to estimate the parameters of various probability density functions. The EM algorithm is well-suited for handling incomplete data and fitting mixture models. The study considered widely used distributions in reliability engineering, such as the Exponential, Normal, Lognormal, Weibull, and Gamma distributions (KUMAR; JAIN; GANGOPADHAY, 2021).

Mixture models combines different probability density functions, varying both the types and number of components in each fitting. Model adequacy was assessed through both quantitative (goodness-of-fit tests) and qualitative (graphical analysis) techniques. These included histograms, comparisons between empirical and fitted TBF distributions, and probability plots.

For the quantitative evaluation, numerical methods were used to conduct goodness-of-fit tests, including the Anderson-Darling (AD) and Kolmogorov-Smirnov (KS) tests, to determine whether the data followed specific theoretical distributions. The Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) were also applied to compare models and identify the best-fitting configurations. Based on these results, the most appropriate distribution types and combinations were identified for modeling the failure data.

Additionally, a clustering analysis was conducted to explore the hypothesis that the dataset includes multiple failure causes, resulting in distinct subgroups (clusters). This step supported the identification of similar failure behaviors and facilitated the exploration of potential patterns in the data.

Finally, a sensitivity and computational cost analysis was conducted to evaluate how the number of components in the mixture models influences the results. The goal was to identify a suitable trade-off between model complexity, goodness-of-fit, and computational cost.

1.4 Document Structure

The chapters of this dissertation are organized as follows:

Chapter 2 presents the theoretical foundation of the study, covering key concepts such as software reliability, mixture models, and the Expectation–Maximization algorithm.

Chapter 3 details the methods and techniques employed in the research. It describes the data preparation process, the application of mixture models, and the validation of results, including goodness-of-fit tests and cluster selection methods. Furthermore, the dataset used and the computational tools applied are introduced.

Chapter 4 presents and discusses the results obtained throughout the study, including the fitted mixture models, graphical analyses, identified clusters, and the distributions that best de-

scribe the time between failures. A comparison of different goodness-of-fit tests is conducted, and the performance of the models is analyzed in terms of complexity and fit to real-world data. The relevance of the findings in the context of software reliability is discussed.

Finally, Chapter 5 presents the conclusions of the study, highlighting its contributions to the software reliability literature, summarizing the key findings, discussing the study limitations, and suggesting directions for future research.

Chapter 2

Literature Review: Background

2.1 Introduction

This chapter reviews the key concepts and studies relevant to the present research. It begins with an overview of software reliability, then examines the role of mixture models in reliability analysis, and subsequently discusses the clustering techniques and the Expectation–Maximization algorithm employed for parameter estimation in mixture models.

2.2 Software Reliability

The study of software reliability is grounded in a set of fundamental concepts that describe how systems operate, experience disruptions and recover. Central to reliability analysis are three related concepts: faults, errors, and failures (AVIZIENIS et al., 2004):

- **Fault**: A fault is the adjudged or hypothesized cause of an error. Faults may be dormant (inactive) or active, and their activation can propagate through the system, potentially resulting in failures.
- **Error**: An error is the system state that may lead to a subsequent failure. It represents the internal manifestation of a fault.
- **Failure**: A failure occurs when a delivered service deviates from its correct service. This deviation can stem from non-compliance with the functional specification or deficiencies within the specification itself.

Software reliability is defined as the probability that software will operate without failure for a specified period under given environmental conditions (IEEE, 1990). This probabilistic definition captures key aspects essential to build dependable software systems (MUSA, 2004).

The stochastic nature of software reliability reflects uncertainties in both software behavior and operational usage patterns (GOEL, 1985). A failure is said to occur when software

behavior deviates unacceptably from its specified requirements (MUSA, 1979), with faults representing the underlying defects that trigger these failures (OHBA, 1984).

The surrounding environment, such as: hardware platforms, operating systems, and user behavior, directly affects the reliability of a system (GOEL, 1985). In this context, the operational profile, which characterizes expected usage scenarios, becomes a critical factor for accurate reliability assessment (GOKHALE; TRIVEDI, 1999).

During the operational phase, a system interacts continuously with its environment, which includes physical conditions, users, administrators, service providers, infrastructure elements, and even potential intruders. Maintenance during this phase involves not only correcting faults but also adapting the system to meet changing requirements and environmental conditions. The integration of off-the-shelf (OTS) components introduces additional complexities and challenges (AVIZIENIS et al., 2004).

Unlike physical systems, software does not deteriorate with use, faults, once corrected, generally remain fixed permanently (MUSA, 1979). However, software may become obsolete due to evolving user needs, hardware changes, or shifts in the technological landscape (LYU, 2007). Measuring and quantifying software reliability is essential throughout the software development lifecycle (MUSA, 1979).

As society becomes increasingly dependent on software systems, the importance of software reliability has grown substantially. From communication networks and transportation systems to medical devices and critical infrastructure, software failures can range from minor disruptions to catastrophic events including threats to human safety (MATIAS et al., 2014).

The economic impact of software usage is also significant, improving reliability can reduce development and maintenance costs, increase product value, and improve user satisfaction (LI et al., 2008). Conversely, poor reliability can lead to reputation damage, revenue loss, and possible legal consequences (LYU, 2007). As hardware reliability has advanced considerably, software is now often the primary limiting factor in overall system dependability (KEENE; LANE, 1992).

The increasing complexity and inter-connectivity of software systems raise the likelihood of failure and make fault isolation more difficult (MATIAS et al., 2014). For project managers, reliability metrics are crucial for decision-making, enabling better scheduling, progress tracking, and release planning (MUSA, 1979), while helping balance the costs of testing with the benefits of reliability improvement (GOEL, 1985).

Despite its critical role, software reliability remains difficult to measure and predict. Traditional reliability models often produce optimistic estimates, largely due to assumptions such as accurate operational profiles and exhaustive testing coverage (GOKHALE; TRIVEDI, 1999). Many residual faults remain latent, only surfacing under rare or unforeseen operating conditions, which makes them particularly challenging to detect and eliminate (AVIZIENIS et al., 2004).

To support reliability modeling and analysis, a set of standard metrics is widely used. These

metrics are used to analyze the system behavior across the software lifecycle:

- ❑ **Time Between Failures (TBF)**: The elapsed time between two consecutive failures, offering a basic measure of system reliability.
- ❑ **Mean Time To Failure (MTTF)**: The expected operational time before a failure occurs, typically derived from the reliability function $R(t)$.
- ❑ **Mean Time Between Failures (MTBF)**: The average time between failures in a repairable system, calculated as MTBF $=$ MTTF $+$ MTTR.
- ❑ **Mean Time To Repair (MTTR)**: The average time required to restore a failed system to operational condition.
- ❑ **Reliability** $R(t)$: The probability that the system will operate without failure over a time interval t.
- ❑ **Unreliability** $F(t) = 1 - R(t)$: The probability that the system will experience at least one failure within time t.
- ❑ **Failure Rate** $\lambda(t)$: The instantaneous rate of failure at time t, conditional on the system operating up to that point.
- ❑ **Bx Life**: The time by which $x\%$ of systems are expected to have failed, used to define reliability thresholds (e.g., B10 life).
- ❑ **Reliable Time** $t(R)$: The time interval during which the system maintains a specified level of reliability R.

While traditional models play an important role in software reliability engineering, they often rely on idealized assumptions, such as perfect fault removal, fault independence, and stable operational profiles, that rarely hold in real-world scenarios (GOEL, 1985). As a result, these models frequently face limitations in their predictive power across diverse software development contexts. Such limitations stem from their dependency on specific assumptions about the development environment and the nature of software failures (BLOSTEIN; MILJKOVIC, 2019). Consequently, a model that performs well for one dataset might be ineffective for another.

2.3 Mixture Models

Mixture models are a probabilistic tool designed to represent the presence of subpopulations within an overall population. They are useful when observations of a random variable arise from multiple underlying conditions or sources (ELMAHDY; ABOUTAHOUN, 2013). This section discusses the fundamental aspects, mathematical formulation, components, parameter estimation methods and applications of mixture models in the context of reliability.

Mixture models are effective for describing heterogeneous populations that can be decomposed into a finite number of more homogeneous subpopulations (ELMAHDY, 2017). In the context of this work, the assumption is that heterogeneity is linked to diverse failure causes and operational characteristics, such as workload intensity. By using mixture models, each latent subpopulation can be represented by a distinct probability distribution, capturing the diversity in failure behavior.

In reliability analysis, the failure of components or systems can occur through multiple distinct failure causes. A mixture of Weibull distributions, for example, is suitable for such scenarios, since failure causes may follow different statistical distributions (ELMAHDY, 2015).

Mixture models are a combination of several probability distributions. Therefore, understanding the components of a mixture model is essential for interpreting its structure and behavior:

- ❑ **Subpopulations or Components:** These are the individual distributions being combined, such as Weibull, Lognormal or Exponential distributions. Components can be of the same type or of different types. For example, in a sample of failure times, two or more subpopulations may be present due to different failure causes. Each failure cause can be responsible for generating a distinct subgroup of failure times. The same applies to failure type, operational profile, and other influential factors.

- ❑ **Mixing Weights (or Proportions):** Represented by p_j or w_i, these values indicate the proportion of each subpopulation within the mixture. They are positive and must sum to 1.

- ❑ **Parameters:** Each subpopulation has its specific parameters (e.g., shape and scale for Weibull, mean and variance for Normal distributions). The complete parameter vector for a mixture model includes both the parameters of each subpopulation and the mixing weights.

A general k-fold mixture model involves k subpopulations. The cumulative distribution function (CDF) of the overall mixture model is given by:

$$G(t) = \sum_{j=1}^{K} p_j F_j(t), \tag{1}$$

where $F_j(t)$ is the CDF of the j-th subpopulation, and p_j is the mixing probability (or weight) of the j-th subpopulation, with $p_j > 0$ and $\sum_{j=1}^{K} p_j = 1$.

The probability density function (PDF) of a mixture model is expressed as:

$$g(t) = \sum_{j=1}^{K} p_j f_j(t), \tag{2}$$

where $f_j(t)$ is the PDF associated with the j-th subpopulation. Specifically, for a two-component mixture model, the PDF becomes:

$$f(t) = wf_1(t) + (1 - w)f_2(t), \tag{3}$$

The reliability function $R(t)$ for an m-fold mixture model can be expressed as:

$$R(t) = \sum_{i=1}^{m} \pi_i R_i(t) \tag{4}$$

where π_i represents the mixing proportion of the i-th component, and $R_i(t)$ is the reliability function corresponding to the i-th distribution in the mixture, and m denotes the total number of component distributions included in the mixture model. The overall reliability function $R(t)$ is thus the weighted sum of the component reliability functions.

Estimating the parameters of a mixture model is a fundamental step that directly influences model performance, and several techniques are available for this purpose, the most commons are:

- ❑ **Maximum Likelihood Estimation (MLE):** MLE seeks to find the parameter values that maximize the likelihood of the observed data (ELMAHDY, 2015; RAZALI; SALIH, 2009; ZHAO; STEFFEY, 2013; ELMAHDY; ABOUTAHOUN, 2013; RUHI; SARKER; KARIM, 2015).

- ❑ **Expectation-Maximization (EM) Algorithm:** The EM algorithm is an iterative procedure commonly used for mixture models, in which the subpopulation identity of each observation is treated as missing data. It alternates between an expectation (E) step and a maximization (M) step to refine the parameter estimates. (BLOSTEIN; MILJKOVIC, 2019; ELMAHDY, 2015; RAZALI; SALIH, 2009; ZHAO; STEFFEY, 2013; ELMAHDY; ABOUTAHOUN, 2013; RUHI; SARKER; KARIM, 2015).

Mixture models are considered robust tools in data analysis due to their capacity to accommodate various data complexities. Unlike traditional single-distributions, mixture models can represent data arising from multiple sources, making them suitable for scenarios where the underlying structure is not homogeneous. Key reasons for employing mixture models in reliability studies include:

- ❑ **Heterogeneous Populations:** Many real-world datasets consist of observations from distinct subpopulations with different characteristics (ELMAHDY, 2017).

- ❑ **Multiple Failure Causes:** A single distribution is often inadequate to represent multiple failure mechanisms. Mixture models are appropriate for modeling situations involving early failures and wear-out failures (RAZALI; SALIH, 2009; ELMAHDY, 2015).

- ❑ **Improved Model Fit:** When a single distribution cannot satisfactorily model the data, as evidenced by goodness-of-fit tests or deviations on probability plots, a mixture model often provides a better fit (ZHAO; STEFFEY, 2013; ELMAHDY, 2015).

- ❑ **Modeling Complex Distribution Shapes:** Mixture models can represent complex PDFs and hazard functions (ELMAHDY, 2017; WU et al., 2024).
- ❑ **Physical Interpretability:** In some cases, each component of a mixture model corresponds to an identifiable physical process or group, offering meaningful insights into the system under study (ELMAHDY; ABOUTAHOUN, 2013).
- ❑ **Handling Complex Reliability Data:** When subpopulations cannot be easily separated due to undocumented changes or unknown factors, mixture models provide a robust approach to data characterization and prediction (ELMAHDY, 2017).

However, despite their flexibility and modeling power, their use introduces a number of practical and theoretical challenges that must be considered during implementation:

- ❑ **Increased Complexity and Number of Parameters:** Combining multiple models inherently increases the number of parameters, making estimation and interpretation more difficult (BLOSTEIN; MILJKOVIC, 2019).
- ❑ **Parameter Estimation Difficulties:** Traditional optimization methods, such as Newton's method, may struggle with multi-modal likelihoods, parameter constraints, and sensitivity to initial values, leading to convergence issues (OKAMURA; MURAYAMA; DOHI, 2004). Although the EM algorithm is less sensitive to initial values, it generally converges slowly and requires careful determination of a stopping criterion (OKAMURA; WATANABE; DOHI, 2002).
- ❑ **Selection of Component Models or Distributions:** Choosing suitable base distributions is critical. Inappropriate selection can degrade predictive performance, and often there is no clear theoretical guideline for selection, making it a largely empirical process (BLOSTEIN; MILJKOVIC, 2019).
- ❑ **Data Requirements:** Insufficient data can impair model performance and result in unreliable predictions (BLOSTEIN; MILJKOVIC, 2019).
- ❑ **Practical Implementation and Acceptance:** The increased sophistication of mixture models and associated estimation algorithms can hinder their adoption among practitioners accustomed to simpler models (OKAMURA; DOHI, 2021).
- ❑ **Interpretation of Results:** Interpreting the output of a mixture model can be more complex compared to a single model (BLOSTEIN; MILJKOVIC, 2019).

Mixture models represent an interesting approach for modeling failure data in reliability analysis. Their capacity to account for multiple failure mechanisms makes them suitable for complex datasets. However, the associated computational cost and model selection decisions

must be addressed carefully. The following section introduces clustering methods, which provide an alternative strategy for uncovering latent subpopulations in data without relying on predefined distributions assumptions.

2.4 Clustering Methods

Clustering is a process of organizing a collection of patterns into clusters based on their similarity. The main objective of clustering, also referred to as cluster analysis, is to discover the natural groupings of a set of patterns, points, or objects (JAIN; MURTY; FLYNN, 1999).

The primary goal of clustering is to uncover the inherent structure within data, making it inherently exploratory in nature (JAIN, 2010). Ideally, patterns within a valid cluster are more similar to each other than to patterns belonging to different clusters. An operational definition of clustering is to partition data into K groups based on a measure of similarity, such that similarities among objects within the same group are maximized, while similarities between objects in different groups are minimized (JAIN, 2010).

Clustering is a form of unsupervised learning, meaning that it operates on unlabeled data. Unlike supervised classification, clustering does not utilize prior category labels, rather, the groupings are derived directly from the data itself (JAIN; MURTY; FLYNN, 1999; JAIN, 2010).

Clustering algorithms can produce different partitions depending on the criterion used. Because the true structure is often unknown, evaluation techniques must be applied. They categorize cluster validity approaches into three main types (HALKIDI; BATISTAKIS; VAZIRGIANNIS, 2001):

- ❑ **External criteria:** Compare clustering outcomes to known ground truth or labeled data.
- ❑ **Internal criteria:** Assess the quality of clustering based solely on intrinsic properties of the data, such as cohesion and separation.
- ❑ **Relative criteria:** Evaluate and compare multiple clustering results using validity indices to determine the best configuration.

Mixture modeling plays a role in model-based clustering approaches (MEILĂ; HECKERMAN, 2001). In this framework, it is assumed that data points are generated from a mixture of several probability distributions, each representing a different cluster or subpopulation (MENARDI, 2011). The overall population density, $f(x)$, is modeled as the sum of the densities of these individual components, $f_m(x)$, weighted by their corresponding mixing proportions π_m (MENARDI, 2011). Key aspects of mixture modeling include:

- ❑ **Representation of Clusters:** Each cluster corresponds to a component in the mixture model (BAUDRY et al., 2010).

- ❑ **Parameter Estimation:** Parameters of the mixture components (e.g., means, variances) and their mixing proportions are estimated from the data. A widely used method for this estimation is the Expectation-Maximization algorithm (MENARDI, 2011).
- ❑ **Determining the Number of Clusters:** Model selection criteria such as the Bayesian Information Criterion (BAUDRY et al., 2010) and the Akaike Information Criterion (JAIN, 2010) are typically employed to select the optimal number of mixture components, thus determining the number of clusters.
- ❑ **Relationship to Density-Based Clustering:** Mixture modeling is a subclass of density-based clustering approaches (MENARDI, 2011). Clusters are identified as high-density regions in the data space, implicitly defined by the components of the mixture model (HARTIGAN, 1985).

2.4.1 K-Means Clustering

The k-means algorithm is one of the most widely used partitional clustering algorithm. It partitions a set of n data points into K clusters, where K is a user-specified parameter. The objective is to minimize the squared error between the empirical mean (centroid) of a cluster and the points assigned to that cluster (CELEBI; KINGRAVI; VELA, 2013). This minimization seeks to produce compact and well-separated clusters. The general steps of the k-means algorithm are as follows:

- ❑ **Initialization:** The algorithm begins by selecting K initial cluster centers. Several initialization methods exists, including:
  - Choosing the first K points from the dataset (sensitive to the order of the data).
  - Randomly selecting K points from the dataset (common practice is to run multiple times to mitigate randomness).
- ❑ **Assignment:** Each data point is assigned to the nearest cluster center, usually using the Euclidean distance as the distance metric.
- ❑ **Update:** The centroids of the clusters are updated by computing the mean of all points assigned to each cluster.
- ❑ **Iteration:** Steps of assignment and update are repeated until a stopping criterion is met. Typical stopping conditions include:
  - No significant change in cluster assignments between iterations.
  - The relative improvement in the Sum of Squared Errors (SSE) falls below a threshold.
  - A maximum number of iterations is reached.

A notable challenge with k-means is its sensitivity to the initial cluster center placements, different initializations can lead to distinct final clusterings and to suboptimal solutions. Therefore, it is common practice to perform multiple runs with different random initializations and select the result with the lowest SSE (CELEBI; KINGRAVI; VELA, 2013). In addition to algorithmic strategies, assessing the quality of clustering and selecting the appropriate number of clusters remain central concerns.

These considerations highlight the complexity nature of clustering analysis, the selection of an appropriate algorithm and validation method must be aligned with the data characteristics and the analytical objectives. Since clustering often serves as a step for subsequent modeling techniques, it is necessary to address its limitations. One such technique that builds upon clustering concepts while addressing its limitations is the Expectation-Maximization algorithm, which is explored in the following section.

2.5 Expectation-Maximization (EM) Algorithm

The Expectation-Maximization (EM) algorithm is an iterative technique for finding the maximum-likelihood estimate of the parameters of an underlying distribution from a given dataset, particularly when the data is incomplete or contains missing values (BILMES et al., 1998). Two primary applications of EM are distinguished in the literature:

- ❑ **Handling Missing Data:** EM is particularly effective when datasets suffer from missing values. Applications include missing output data in dynamic model identification, censored or truncated data, and multivariate datasets with partially missing observations (MCLACHLAN; KRISHNAN, 2008).
- ❑ **Dealing with Hidden Variables:** EM simplifies the likelihood function by assuming the existence of hidden (latent) variables (BILMES et al., 1998).

The EM algorithm operates through an iterative process composed of two fundamental steps, which alternate until the algorithm reaches convergence. Each iteration involves first estimating the hidden components of the data and then updating the model parameters accordingly. These steps are described as follows:

1. **Expectation Step (E-step):** Given the observed data and the current parameter estimates $\theta^{(m)}$, the E-step computes the conditional expectation of the complete-data log-likelihood:

$$Q(\theta|\theta^{(m)}) = \mathbb{E}_{X|Y=y,\theta^{(m)}}[\log p(X|\theta)], \tag{5}$$

where Y represents the observed (incomplete) data, X the complete data (observed plus missing or hidden data), and θ the parameters to be estimated. The E-step fills in the missing data by their conditional expectations given the observed data and current parameter estimates (MCLACHLAN; KRISHNAN, 2008).

2. **Maximization Step (M-step):** In this step, the parameters are updated by maximizing the Q-function:
$$\theta^{(m+1)} = \arg\max_{\theta \in \Omega} Q(\theta|\theta^{(m)}). \tag{6}$$
The M-step treats the conditional expectations from the E-step as known data and updates the model parameters accordingly (MCLACHLAN; KRISHNAN, 2008).

The E and M steps are repeated until convergence is achieved, typically when the changes in parameter estimates fall below a predefined threshold. The EM algorithm offers a range of features that contribute to its widespread use in statistical modeling, such as:

- ❑ **Monotonic Convergence:** The observed-data likelihood is guaranteed to increase or stay constant with each iteration, ensuring monotonic convergence (MCLACHLAN; KRISHNAN, 2008).
- ❑ **Numerical Stability:** EM is recognized for its stable numerical behavior (MCLACHLAN; KRISHNAN, 2008).
- ❑ **Handling Incomplete Data:** EM provides a principled and robust framework for parameter estimation in the presence of missing or hidden data (BILMES et al., 1998).
- ❑ **Robustness Against Outliers:** The E-step integrates missing data, leading to more robust parameter estimates even in the presence of outliers (SAMMAKNEJAD; ZHAO; HUANG, 2019).

Despite its advantages, the EM algorithm comes with important limitations that must be considered when applying it to solving problems. Among the most commonly cited drawbacks are:

- ❑ **Slow Convergence:** EM may converge slowly, especially near saddle points or flat regions of the likelihood surface (MCLACHLAN; KRISHNAN, 2008).
- ❑ **Sensitivity to Initialization:** The final solution heavily depends on the initial parameter values. Poor initialization can lead EM to converge to local rather than global optima (SAMMAKNEJAD; ZHAO; HUANG, 2019; MICHAEL; MELNYKOV, 2016). Strategies such as random starts, k-means clustering, and model averaging have been proposed to mitigate this issue (PANIĆ; KLEMENC; NAGODE, 2020).
- ❑ **Local Optima:** EM is only guaranteed to converge to a stationary point, which might be a local maximum or even a saddle point, not necessarily the global maximum (WU, 1983).
- ❑ **Intractable Steps:** For certain complex incomplete-data problems, either the E-step or the M-step can become analytically or computationally infeasible (MOON, 1996).

- ❑ **No Direct Standard Errors:** The basic EM framework does not inherently provide standard errors for the parameter estimates. Additional methods such as numerical differentiation can be used to estimate them (MCLACHLAN; KRISHNAN, 2008).
- ❑ **Computational Cost:** As the size and dimensionality of datasets increase, the computational demands of EM can become significant (SAMMAKNEJAD; ZHAO; HUANG, 2019; ZHANG; ZHANG; YI, 2004).
- ❑ **Specifying Model Complexity:** In mixture modeling applications, the number of components must be specified beforehand. An incorrect choice can degrade model performance (PANIĆ; KLEMENC; NAGODE, 2020).

The EM algorithm guarantees improvement of the likelihood function under certain conditions, with each iteration ensuring that the likelihood of the model fitting the data increases or remains constant (GUPTA; CHEN et al., 2011). However, it does not ensure convergence to a global optimum. The likelihood surface contains multiple local maxima where EM can become trapped depending on the initial parameters.

The inherent sensitivity to initialization of the EM algorithm, selecting suitable starting values constitutes a critical step in ensuring robust and efficient performance (MICHAEL; MELNYKOV, 2016; PANIĆ; KLEMENC; NAGODE, 2020). Several strategies have been proposed to address this challenge, including random restarts and short preliminary runs of EM itself. Among these approaches, the latter has demonstrated particular promise due to its effectiveness in avoiding local maxima. Although no universally optimal method exists, studies suggest that designed initialization can improve convergence speed and likelihood maximization, especially in high-dimensional or overlapping cluster settings (BIERNACKI; CELEUX; GOVAERT, 2003; MELNYKOV; MELNYKOV, 2012).

With the key aspects of the EM algorithm established, the focus now shifts to assessing the adequacy of the fitted models. The next section introduces Goodness-of-Fit (GoF) tests and graphical analysis for distribution assessment, which serve as tools for evaluating how well statistical models capture the underlying structure of the observed data.

2.6 Goodness-of-Fit Tests and Graphical Analysis

Evaluating how well a statistical model represents observed data is a fundamental task in data analysis. In this context, Goodness-of-Fit (GoF) tests and graphical methods play a central role by providing formal and visual means to assess the adequacy of a hypothesized distribution.

At the core of distributional analysis are the Probability Density Function (PDF) and the Cumulative Distribution Function (CDF). The PDF describes the relative likelihood of a continuous random variable assuming a specific value, with the total area under the curve representing the probability over an interval (ANDERSON, 2011). The CDF is defined as:

$$F(x) = P(X \leq x), \tag{7}$$

expresses the probability that a random variable X takes on a value less than or equal to x. CDFs are particularly important in GoF testing, where comparisons are often made between empirical and theoretical CDFs.

Graphical techniques, such as quantile-quantile plots and probability-probability plots, offer visual diagnostics for assessing distributional assumptions. Q-Q plots are useful for detecting discrepancies in the tails of a distribution. In contrast, P-P plots emphasize differences in the central portion of the distribution, making them suitable for assessing overall fit (GNANADESIKAN; WILK, 1968).

Goodness-of-Fit tests are statistical procedures designed to determine whether a sample originates from a specified theoretical distribution (ANDERSON, 2011). These tests compare the empirical distribution function (EDF), a step function derived from the sample data, with the cumulative distribution function of the hypothesized distribution. The extent of deviation between the two functions forms the basis for computing test statistics.

By quantifying the distance between observed and expected values, GoF tests assist in identifying departures from the hypothesized model, guiding model refinement and selection (ANDERSON, 2011; GNANADESIKAN; WILK, 1968). Several GoF tests and model selection criteria are widely used in practice:

- ❑ **Anderson-Darling (AD) Test:** A non-parametric test that gives more weight to the tails of the distribution, making it particularly effective in detecting tail discrepancies. It is suitable for comparing a sample with a continuous distribution or testing homogeneity across multiple samples (SCHOLZ; STEPHENS, 1987; RAZALI; WAH et al., 2011).

- ❑ **Kolmogorov-Smirnov (KS) Test:** Another non-parametric test that measures the maximum distance between the empirical and theoretical CDFs (BERGER; ZHOU, 2014). The KS test is sensitive to deviations anywhere along the distribution.

- ❑ **Akaike Information Criterion (AIC):** A model selection criterion based on information theory. It penalizes model complexity to avoid overfitting and is defined as:

$$\text{AIC} = -2 \times \log(\text{likelihood}) + 2K, \tag{8}$$

  where K is the number of model parameters. Lower AIC values indicate better trade-offs between goodness-of-fit and simplicity (GERNAND; FENSKE, 2009).

- ❑ **Bayesian Information Criterion (BIC):** Similar to AIC but includes a stronger penalty for model complexity that increases with sample size:

$$\text{BIC} = -2 \times \log(\text{likelihood}) + K \times \log(n), \tag{9}$$

where n is the sample size. BIC tends to favor simpler models and is consistent under large samples when the true model is among the candidates (GERNAND; FENSKE, 2009).

It is important to clarify that the classical Anderson–Darling statistic (A^2) is, by definition, non-negative ($A^2 \geq 0$). However, in this study, the implementation provided by the `scipy.stats.anderson_ksamp` function was employed for multi-sample comparisons. This function does not return the traditional A^2 statistic. Instead, it outputs a standardized and transformed test statistic whose value may be either positive or negative, as explicitly documented in the SciPy library.

In this context, negative values do not indicate an error or invalid result. Rather, the sign of the statistic conveys information about the direction of the deviation between the compared distributions. Specifically, negative values may indicate a relatively better agreement among the samples in a given direction, whereas positive values suggest greater divergence. Therefore, the occurrence of negative Anderson–Darling statistics in the reported results is a consequence of the adopted implementation and does not contradict the theoretical properties of the classical Anderson–Darling test.

The assessment of distributional assumptions requires a combination of graphical diagnostic tools and formal statistical tests. Although graphical tools such as histograms, Q-Q plots, and P-P plots are inherently subjective, they provide valuable insight into the distributional characteristics of the data. Figure 1 shows a goodness-of-fit analysis of a synthetic data sample, which includes multiple diagnostic approaches to evaluate the adequacy of the fitted exponential distribution.

The histogram (upper left panel) provides a visual summary of the data's frequency distribution overlaid with the theoretical probability density function (PDF) of the fitted exponential distribution. The shape of the histogram allows identification of key features, such as skewness, the presence of multiple modes, or heavy tails. In this analysis, the histogram suggests a positively skewed distribution, consistent with exponential behavior. The close alignment between the empirical histogram and the theoretical PDF (orange dashed line) provides initial visual evidence supporting the exponential model assumption.

The empirical versus theoretical CDF comparison (upper right panel) displays both the empirical cumulative distribution function (blue line with circles) and the theoretical CDF (orange line with crosses). This plot allows for the assessment of distributional fit across the entire range of the data. Good model fit is indicated by close alignment between the two curves. Any systematic deviations suggest areas where the theoretical model fails to capture the data characteristics adequately.

The P-P plot (lower left panel) compares empirical probabilities against theoretical probabilities by plotting the empirical CDF values against the corresponding theoretical CDF values. Under perfect distributional fit, all points would lie exactly on the 45-degree reference line (orange dashed line). Deviations from this line indicate areas of poor fit.

The Q-Q plot (lower right panel) compares empirical quantiles against theoretical quantiles. This plot is particularly sensitive to departures in the tails of the distribution. Points lying on the 45-degree line indicate good distributional fit, while departures reveal specific areas of model inadequacy.

In contrast to graphical tools, formal GoF tests provide statistical criteria for decision-making, using significance thresholds and test statistics (RAZALI; WAH et al., 2011). These tests evaluate the null hypothesis that the data follows the specified theoretical distribution against the alternative hypothesis that it does not.

The null hypothesis (H_0) states that the observed data follows the fitted exponential distribution, while the alternative hypothesis (H_1) states that the data does not follow this distribution. The decision rule is based on comparing the calculated test statistic with critical values or p-values against a predetermined significance level, typically $\alpha = 0.05$ (5%).

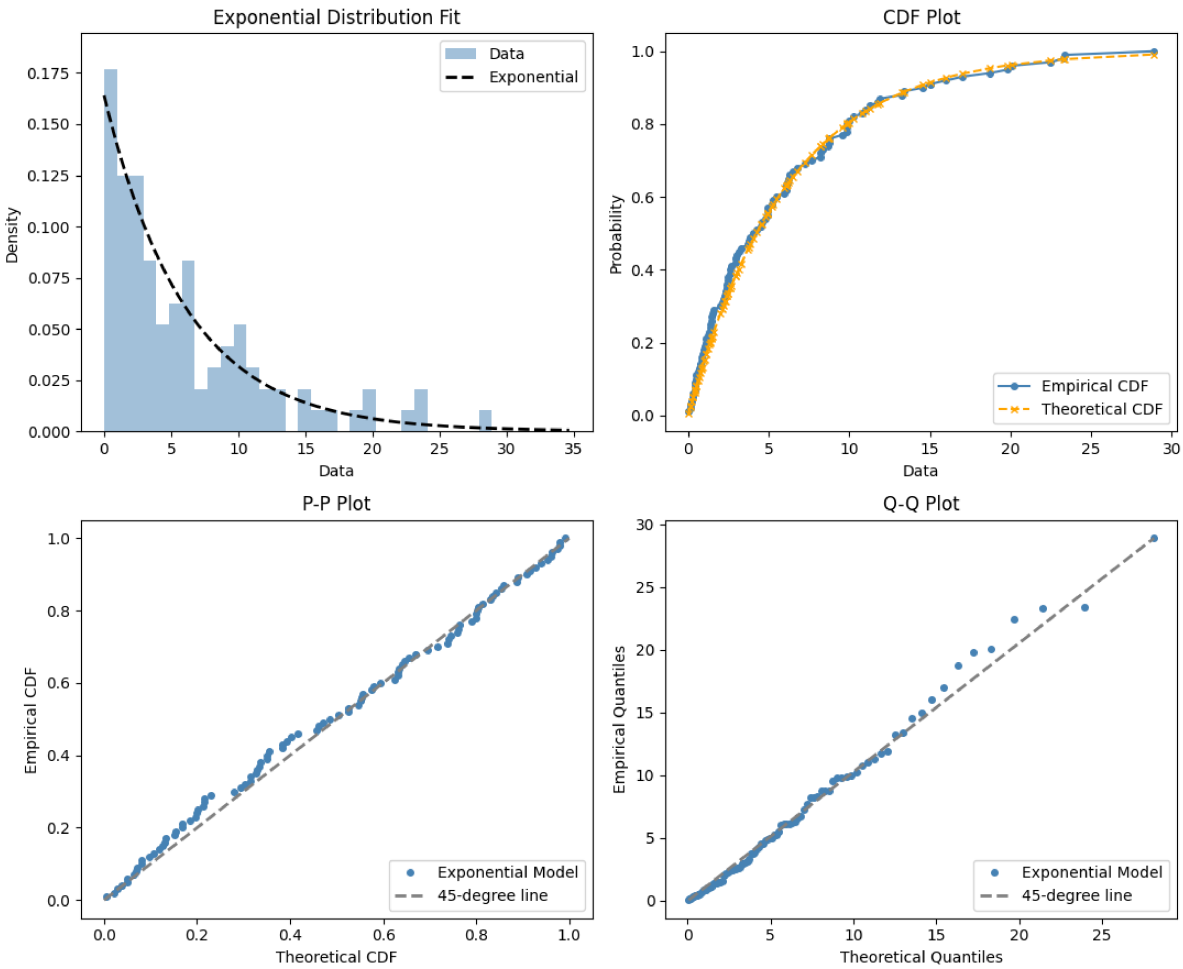

Figure 1 – Goodness-of-Fit Analysis for Exponential Distribution: Histogram with Theoretical PDF Overlay, Empirical vs. Theoretical CDF Comparison, P-P Plot, and Q-Q Plot.

- ❑ **P-value interpretation**: If $p > 0.05$, we fail to reject H_0, suggesting that the data is consistent with the theoretical distribution. If $p \leq 0.05$, we reject H_0, indicating significant departure from the assumed distribution.

Table 1 summarizes the goodness-of-fit test results for the exponential distribution fit. And to facilitate the interpretation of the results reported in Table 1, the main goodness-of-fit criteria employed in this analysis are briefly described below, along with their respective roles in assessing model adequacy.

- ❑ **Akaike Information Criterion (AIC)**: Balances model fit and complexity. Lower values indicate a preferable trade-off between goodness-of-fit and parsimony when comparing alternative models.
- ❑ **Bayesian Information Criterion (BIC)**: Similar to AIC but imposes a stronger penalty on model complexity. As with AIC, lower values are preferred.
- ❑ **Kolmogorov–Smirnov Test**: Assesses the maximum distance between the empirical and theoretical cumulative distribution functions. The test statistic of 0.0639 with $p = 0.7849 > 0.05$ indicates no statistically significant deviation from the exponential distribution.
- ❑ **Anderson–Darling Test**: Places greater emphasis on discrepancies in the distribution tails. The obtained statistic of 0.3192 with $p = 0.5344 > 0.05$ also supports the adequacy of the exponential model.

Together, the graphical tools complement formal goodness-of-fit tests by providing visual evidence of model adequacy. While histograms highlight the general shape of the data, Q-Q and P-P plots pinpoint specific areas of agreement or departure from the theoretical distribution. When used alongside statistical tests, these plots support a robust evaluation of the chosen model, ensuring that both visual and quantitative criteria are considered.

This background chapter has established the theoretical foundations necessary to support the methodology adopted in this dissertation. Building upon these concepts, the next chapter shifts the focus to the existing literature by reviewing previous studies on software reliability modeling, failure data analysis and related methodological approaches. This review enables the positioning of the present work within the broader literature and highlights its contributions in relation to prior research.

Table 1 – Goodness-of-fit Test Results for Exponential Distribution.

Criterion	**Value**	**Interpretation**	**Decision**	**Threshold**
AIC	563.60	Lower is better	–	–
BIC	566.21	Lower is better	–	–
Kolmogorov–Smirnov	0.0639	$p = 0.7849$	Fail to reject H_0	$p > 0.05$
Anderson–Darling	0.3192	$p = 0.5344$	Fail to reject H_0	$p > 0.05$

Chapter 3

Literature Review: Related Works

This chapter presents a review of related works relevant to software reliability analysis and failure modeling. It examines previous studies that employ statistical methods, clustering techniques and mixture models for the analysis of software failure data. The chapter aims to identify the main methodologies and limitations reported in the literature. By analyzing these works, this chapter establishes the research gap that motivates the methodology proposed in this dissertation and clarifies the originality and relevance of the present study.

The field of software reliability has been adopting statistical modeling techniques aimed at capturing the complex nature of failure behaviors in software systems. Among these techniques, mixture models have gained prominence due to their capacity to accommodate multimodal and non-homogeneous failure distributions. Several studies have proposed variations of mixture-based approaches, whether through the combination of different distributions and the use of clustering algorithms to enhance fault detection.

As shown in Table 2, prior research has explored modeling strategies, such as the use of Weibull and Lognormal distributions in manufacturing systems (VINEYARD; AMOAKO-GYAMPAH; MEREDITH, 1999), the application of lognormal reliability growth models (MULLEN, 1998), or the integration of probabilistic methods into classical software reliability growth models (SRGMs) (PHAM, 2003; OKAMURA; WATANABE; DOHI, 2002). More recent researches have also incorporated clustering techniques for failure pattern detection (ZHONG; KHOSHGOFTAAR; SELIYA, 2004; CAI; ZHAO; ZHU, 2020) and real-world failure pattern analysis in large-scale systems (SCHROEDER; GIBSON, 2009; SANTOS; MATIAS, 2018).

In contrast to prior studies that primarily addressed fault classification, model selection, or time-to-failure analysis in isolation, the present work proposes an explicit modeling of TBF distributions using both homogeneous and heterogeneous mixtures, combined with unsupervised clustering and comprehensive model validation metrics. Moreover, the integration of clustering algorithms with mixture model fitting enables the identification of latent structures in failure patterns, while sensitivity and computational cost analyses provide insight into model complexity trade-offs.

Table 2 – Comparison of Related Studies on Software Reliability.

Study	**Differences**	**Similarities**
(MULLEN, 1998)	Proposes a software reliability growth model based on the lognormal distribution (LNET), which offers an alternative to traditional exponential-based SRGMs and accounts for variability in fault detection times.	Emphasizes the use of statistical distributions to represent and analyze software failure behavior.
(VINEYARD; AMOAKO-GYAMPAH; MEREDITH, 1999)	Conducts an empirical investigation of Time Between Failures (TBF) and Time To Repair (TTR) in flexible manufacturing environments, focusing on both mechanical and human-induced failures.	Applies Weibull and Lognormal distributions to model failure and repair times; emphasizes data-driven distribution fitting based on observed operational failures.
(OKAMURA; WATANABE; DOHI, 2002)	Introduces a mixed software reliability model (SRM) that incorporates randomized fault detection rates, addressing challenges related to parameter estimation and the modeling of complex failure patterns.	Uses probabilistic modeling to capture failure dynamics and stresses the importance of model adaptability to real-world variability in failure rates.
(PHAM, 2003)	Provides an extensive review of NHPP-based software reliability growth models and associated cost models, integrating considerations of the software life cycle and release policies.	Focuses on statistical modeling of software failures over time.

Continued on next page

Table 2 (continued)

Study	Differences	Similarities
(ZHONG; KHOSHGOFTAAR; SELIYA, 2004)	Applies unsupervised clustering algorithms, such as K-means and Neural-Gas, to identify fault-prone modules in software systems without the need for prior labeling.	Leverages clustering techniques to explore the internal structure of failure data and supports fault localization by uncovering hidden patterns.
(OKAMURA; MURAYAMA; DOHI, 2004)	Develops a unified parameter estimation method based on the EM principle for Discrete SRMs.	Builds upon probabilistic modeling approaches and maintains emphasis on enhancing model generalization to accommodate different failure behaviors.
(HAMILL; GOSEVA-POPSTOJANOVA, 2009)	Performs large-scale fault analysis using datasets from GCC and NASA, focusing on fault localization frequency and categorization across diverse software systems.	Uses empirical methods to assess how faults are distributed and clustered, providing insights into common failure causes across systems.
(SCHROEDER; GIBSON, 2009)	Analyzes failure logs from high-performance computing (HPC) systems over a 9-year period, with an emphasis on hardware-related failures and system-level behavior.	Applies statistical distribution fitting (Weibull, Gamma) to model TBF and TTR, and critiques the inadequacy of simple exponential models in representing real-world reliability data.
(WANG; KHOSHGOFTAAR; NAPOLITANO, 2010)	Investigates ensemble-based feature selection techniques for software defect prediction, testing combinations of 17 ranking algorithms to improve classifier performance.	Applies statistical and ensemble learning techniques to enhance the accuracy of fault prediction models based on source code metrics.

Continued on next page

Table 2 (continued)

Study	Differences	Similarities
(WANG et al., 2011)	Explores the optimal number of software metrics required for effective defect prediction, using a threshold-based feature selection approach to balance model complexity and performance.	Emphasizes the role of statistical analysis in selecting relevant features for improving the classification of fault-prone modules.
(DIGIUSEPPE; JONES, 2012)	Investigates how multiple faults may interact and violate common assumptions in failure clustering approaches, challenging traditional views on failure independence.	Concentrates on the use of clustering to analyze and model failure behavior, highlighting the complexity of real-world failure patterns.
(SANTOS; MATIAS, 2018)	Focuses on identifying patterns of operating system-level failures that span user-space and kernel-space components, leveraging event correlation and pattern discovery techniques.	Utilizes real-world failure logs to uncover failure dependencies and recurrent behaviors, supporting pattern-based failure modeling.
(BLOSTEIN; MILJKOVIC, 2019)	Proposes a finite mixture model to fit left-truncated loss data, using combinations of the Gamma, Lognormal, and Weibull distributions.	Aims to improve model selection and prediction accuracy by using a flexible mixture framework that captures multiple failure behaviors.
(CAI; ZHAO; ZHU, 2020)	Develops a real-time reliability estimation method for mechanical systems using K-means clustering and spline interpolation, particularly suitable for limited data environments.	Applies clustering to interpret system performance data and emphasizes methods that remain effective even with data scarcity.

Continued on next page

Table 2 (continued)

Study	Differences	Similarities
(OKAMURA; DOHI, 2021)	Evaluates the performance of various NHPP-based SRMs with different underlying statistical distributions, aiming to enhance modeling precision and applicability.	Incorporates mixture modeling principles and addresses the importance of distributional diversity in capturing real-world failure behavior.
(KUMAR; JAIN, 2023)	Focuses on the development of heterogeneous mixture models combining Weibull, Lognormal, and Gompertz distributions, and employs an Artificial Neural Network to compare numerical results for software reliability indices.	Shares the use of Weibull and Lognormal distributions and both homogeneous and heterogeneous mixture models. Promotes the use of mixture models (homogeneous and heterogeneous) to improve fit and representation of failure data, acknowledging the limitations of models based on a single distribution.

While previous studies have made significant advancements in software reliability modeling using mixture approaches, this work introduces several improvements that distinguish it from the existing literature. The main contributions of this research are as follows:

- ❑ **TBF Centric Modeling Perspective:** Rather than relying on cumulative failure curves, this work centers on the statistical modeling of TBFs, providing a more flexible perspective on failure processes and facilitating the detection of behavioral patterns.
- ❑ **Comprehensive Exploration of Mixture Models:** Both homogeneous and heterogeneous mixtures were evaluated, combining up to 30 components drawn from Exponential, Normal, Lognormal, Gamma, and Weibull distributions. This modeling enables the representation of multimodal failure behaviors.
- ❑ **Robust Goodness-of-Fit and Visual Assessment:** The adequacy of the models was evaluated using a wide range of statistical metrics, including AIC, BIC, KS and AD tests. In addition, graphical methods, such as histograms with PDF overlays, CDFs, Q-Q plots, and P-P plots, were employed to provide visual support for model validation.
- ❑ **Integration of Clustering Techniques:** Clustering algorithms (K-Means, HDBSCAN, Fuzzy C-Means, and Gaussian Mixture Models) were used to identify latent structures in TBF data.

- ❏ **Pattern Identification Across Operational Environments:** The analysis considered TBFs from multiple operational groups running under the same OS and application context. This allowed for the investigation of environmental influences on failure behavior and the detection of statistical patterns across groups.

- ❏ **Exploratory Statistical Characterization:** For each sample, descriptive statistics (e.g., mean, median, skewness, kurtosis, percentiles, interquartile range) were computed to characterize the failure data and support subsequent modeling decisions.

- ❏ **Sensitivity Analysis for Model Complexity:** A sensitivity analysis was conducted to examine the relationship between the number of components and model performance. Metrics such as AIC, BIC, KS, and AD were tracked across model sizes. Computational cost (runtime, memory usage, CPU load) was also considered to evaluate trade-offs between complexity and efficiency.

- ❏ **Distribution Suitability Insights:** Across samples and models, trends were identified regarding the most frequently used distributions in mixture components.

The next section describes the methodology adopted in this study, including the data processing steps, modeling techniques, clustering strategies, and evaluation procedures employed.

Chapter 4

Methodology

The central motivation of this research stems from the hypothesis that time between failures in software systems are shaped by a multitude of factors, including operational profiles, failure modes, failure causes and types of failure. These influences result in varying failure behaviors across different computing environments, often producing complex distributions. Given this complexity, mixture modeling is proposed as a suitable technique to better capture the underlying structure of the lifetime data.

It is important to correctly distinguish between failure mode and failure cause in the context of this study. A failure mode refers to how the software fails: the externally observable manifestation of the failure and typical examples of software failure modes include crash, hang and error. A failure cause refers to why the software fails, the underlying defect, software fault or mechanism that triggered the failure. For example, the same memory allocation defect (failure cause) may lead to different failure modes depending on the circumstances: it may cause the software to crash, to hang, or to produce incorrect results.

Furthermore, the manifestation of a failure can depend on how the software is used: its operational profile and the execution environment where it runs. In practice, the combination of a given failure cause with a particular operational profile and execution environment can result in different failure modes.

The relationship between failure causes, operational profiles, execution environments, and failure modes is illustrated in Figure 2.

4.1 Data Description

An exploratory analysis was conducted on failure records collected from general purpose computers deployed in academic, administrative, and corporate settings. All computers are under the Windows 7 operating system, which includes a built-in infrastructure called the Reliability Analysis Component - RAC (MICROSOFT, 2020), responsible for automatically logging software failure events. Each RAC event log contains a set of attributes, including:

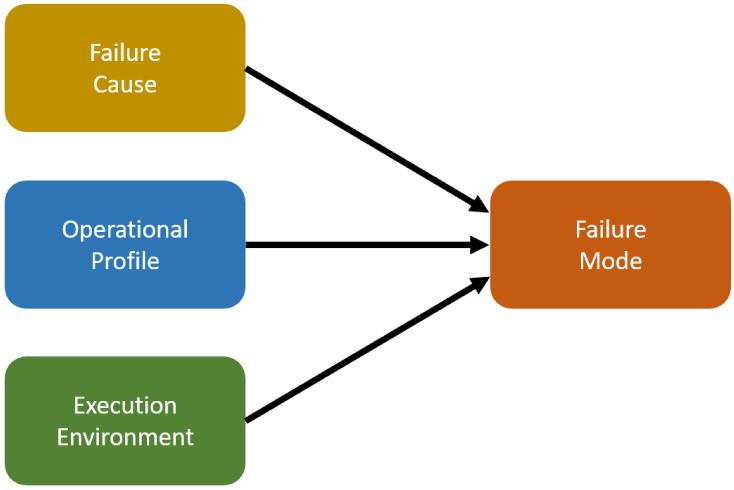

Figure 2 – Relationship Between Failure Causes, Operational Profiles, Execution Environments and Failure Modes.

- **ComputerName**: Identifier of the computer where the failure occurred.
- **EventIdentifier**: Numerical code associated with the event type.
- **InsertionStrings**: Technical details related to the event.
- **Logfile**: The source log file where the event was recorded.
- **Message**: Human-readable explanation of the event.
- **ProductName**: The application or system component involved in the failure.
- **RecordNumber**: Sequential identifier for the log entry.
- **SourceName**: Subsystem or component that triggered the event.
- **TimeGenerated**: Timestamp indicating when the event occurred.
- **User**: The logged-in user at the time of the failure.

To ensure data quality, a filtering process was applied to remove outliers. Specifically, records associated with user-created development programs or non-operational software were excluded. Moreover, while the original dataset includes TBF values equal to zero, these entries were disregarded in the clustering and mixture modeling analyses, as they indicate immediate reoccurrences. The number and proportion of zero-valued TBFs were computed and will be presented in Chapter 4. After this preprocessing, the dataset used for the analyses comprises a total of 40,095 failure records collected from 660 individual computers.

Computers were grouped according to their operational context, resulting in four distinct groups:

- **G1:** Undergraduate Laboratory Computers

- ❏ **G2:** University Administrative Department Computers
- ❏ **G3:** Corporate Environment Computers
- ❏ **G4:** Personal and HomeOffice Computers (Corporate, Academic, and Personal)

4.1.1 Parameters and Failure Characterization

To enhance the dataset for statistical analysis, new parameters were derived from the original RAC logs. These enrichments enabled a more detailed exploration of the failure behavior:

- ❏ **Application Failure:** The application responsible for the failure was inferred by analyzing the `ProductName` and `InsertionStrings` fields. A categorical column in the dataset, `Type`, was created to identify the main source of the failure (e.g., `iexplore` for Internet Explorer).
- ❏ **Failure Cause:** A semantic classification of the failure's cause was derived from diagnostic codes, keywords in `InsertionStrings`, and detailed messages in the `Message` field (e.g., Code: c0000005, Failure Cause: Memory addressing).

The failure records were collected from September 29, 2010 to August 7, 2014.

4.1.2 Dataset Characterization

To facilitate the analysis of group-level differences in key metrics, ridgeline plots were employed. This graphical technique is used for comparing multiple distributions simultaneously. Each plot consists of a series of smoothed density curves, one for each subgroup, stacked vertically with partial overlap. This layout highlights differences in the shape, spread, and central tendency of the distributions. In the context of this study, each curve represents the empirical distribution of a specific metric (e.g., number of failures, number of distinct causes) across computers belonging to the same group.

The x-axis of each ridgeline plot indicates the values of the analyzed metric. Although the y-axis is categorical and primarily used to visually separate the groups, shape and relative height of each curve carry statistical meaning, as they correspond to estimated probability density functions. Consequently, variations in peak height and curve width reflect differences in concentration, dispersion, and distributional structure among the groups. For each group, the following summary statistics are annotated directly on the subplot:

- ❏ **Mean:** The arithmetic average of the distribution.
- ❏ **Median:** The central value that divides the data into two equal halves.
- ❏ **Peak:** The location on the x-axis where the smoothed density function attains its maximum.

The computation of each curve involves multiple steps. Initially, for a given metric (e.g., number of failures), the metric is computed individually for each computer in the group. These individual observations are then aggregated to form an empirical distribution, which is subsequently smoothed to produce the final ridgeline curve. This approach ensures that the group-level curve encapsulates the variability observed at the individual machine level, enabling an understanding of intra-group heterogeneity.

Figure 3 presents the distribution of the number of failures recorded per computer across different groups. The distribution is highly right-skewed in all groups. G1 exhibits the lowest failure counts (mean = 10.3), with a sharp peak at 2.4, suggesting relatively stable systems. Conversely, G4 displays the highest failure incidence (mean = 135.1, median = 59.0), reflecting frequent failure events. Groups G2 and G3 exhibit intermediate behavior but still show considerable skewness and dispersion, indicating high heterogeneity in failure frequency.

As shown in Figure 4, the number of days over which failure events were recorded varies notably among the groups. G1 has a relatively narrow and symmetric distribution centered at 78 days, suggesting consistent, short-term monitoring periods. In contrast, G2 and G4 show right-skewed distributions with longer tails and higher medians (327 and 263 days, respectively), indicating prolonged observation periods and G3 and G4 exhibits a bimodal distribution.

Figure 5 displays the distribution of the number of application failures per computer. In this context, the failure is attributed to the individual application where it occurred, examples in-

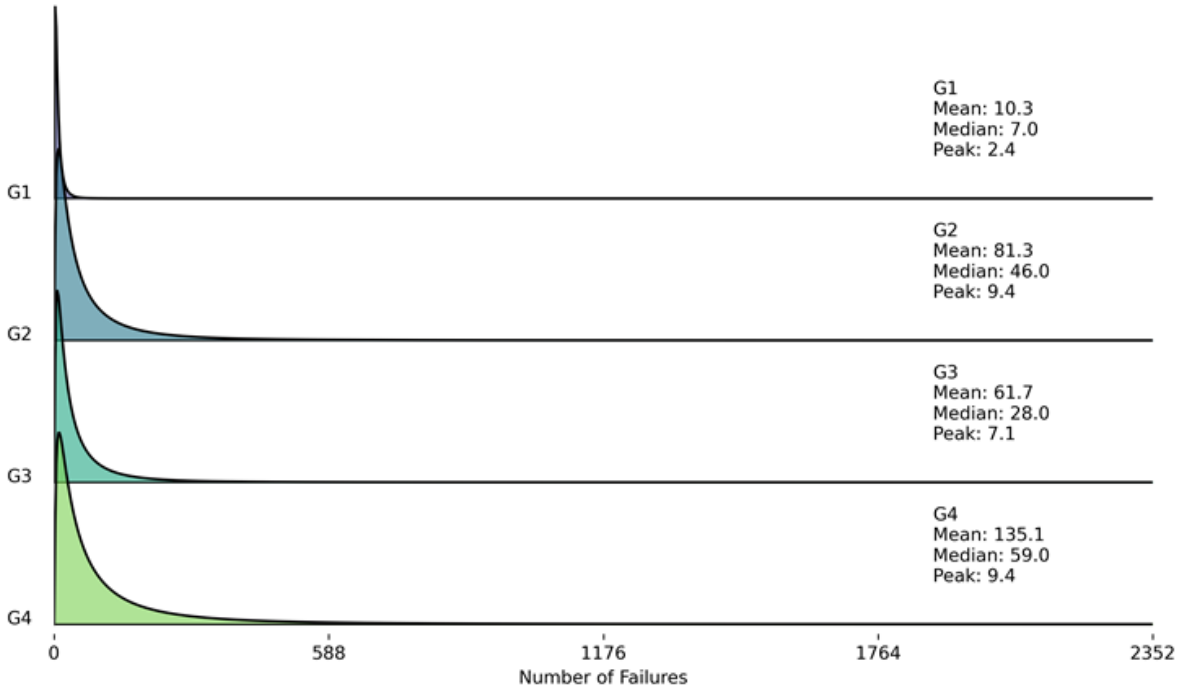

Figure 3 – Ridgeline Plot Comparing the Distribution of Failures per Computer Across Four Groups (G1–G4).

clude failures in user-facing applications such as `iexplore.exe`, `chrome.exe`, `excel.exe`, as well as failures in Operating System components such as `explorer.exe` and `U_Windows`.

In Figure 5, G1 again has the narrowest range (mean = 3.6), suggesting a simpler failure landscape or limited diversity of applications in use. G4 shows the most complex behavior (mean = 17.7, median = 16.0), likely due to varied usage profiles encompassing academic, administrative, and personal tasks. G2 and G3 fall in between, with means around 10.3 and 10.6, respectively.

Figure 6 illustrates the number of distinct causes associated with each application on a computer. G1 is concentrated near a mean = 1.2. On the other hand, G2, G3, and G4 exhibit higher means (2.7 to 2.8) and long right tails.

Together, these descriptive analyses highlight the diverse and heterogeneous nature of software failures across different environments. This variability justifies the need for more flexible modeling approaches, capable of capturing heterogeneous failure distributions effectively.

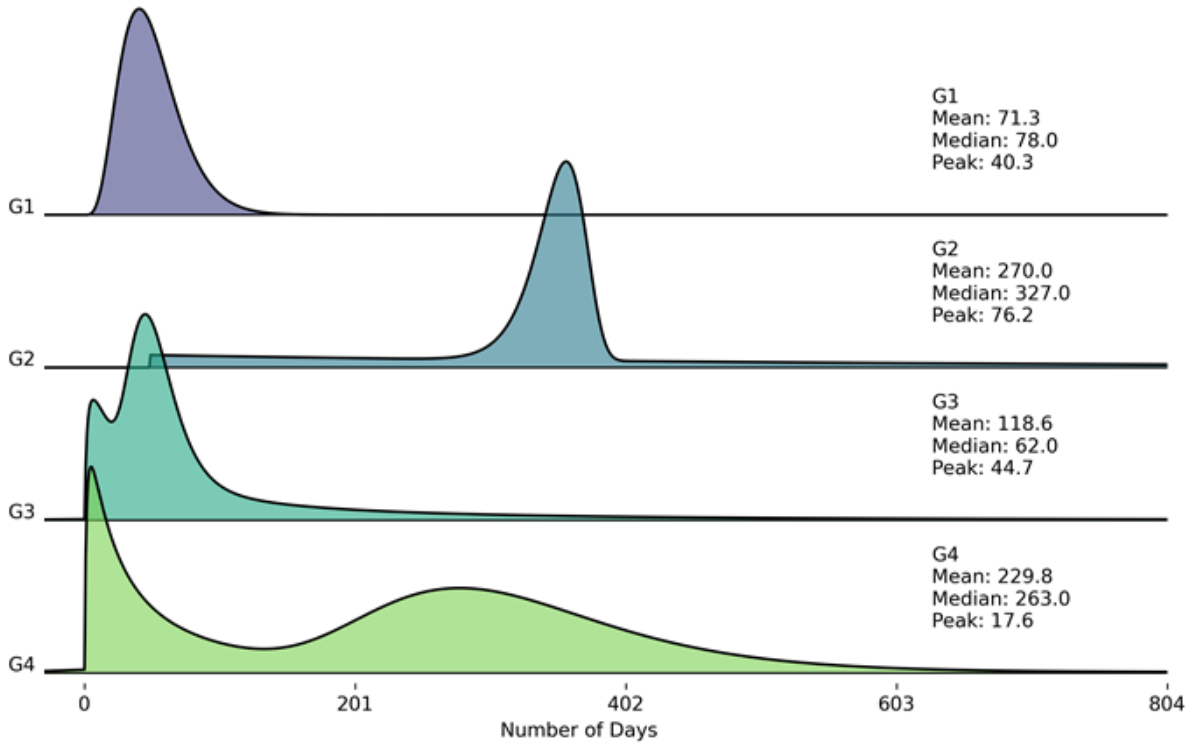

Figure 4 – Ridgeline Plot Comparing the Distribution of Days Sampled per Computer Across Four Groups (G1–G4).

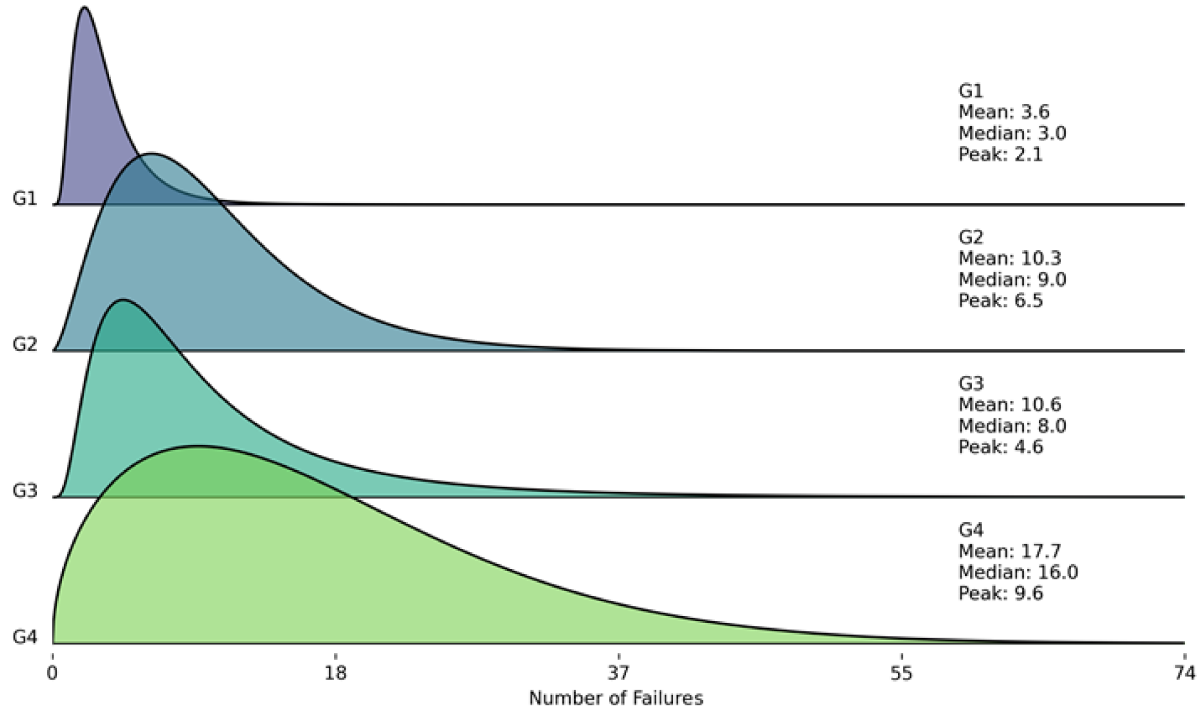

Figure 5 – Ridgeline Plot Comparing the Distribution of Application Failures per Computer Across Four Groups (G1–G4).

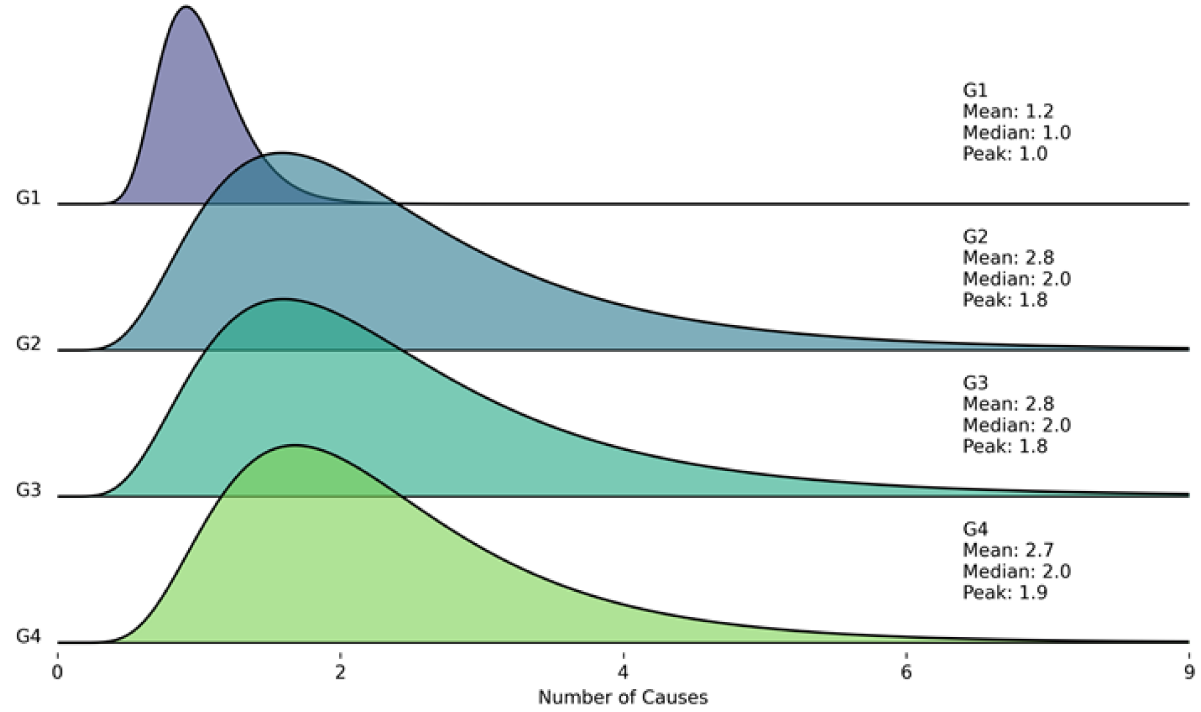

Figure 6 – Ridgeline Plot Comparing the Distribution of Failure Causes per Application per Computer Across Four Groups (G1–G4).

4.2 Modeling Approach

This study investigates the use of homogeneous and heterogeneous finite mixture models to represent the empirical distribution of time between failures. Homogeneous models are composed of multiple components of the same probability distribution (e.g., several Exponential distributions), whereas heterogeneous models allow combinations of different distribution types within a single mixture, such as a blend of Weibull, Gamma, and Lognormal equations. The modeling configuration considered up to 30 components, using five statistical distributions in reliability engineering: Exponential, Normal, Lognormal, Gamma and Weibull. These distributions were selected for their flexibility and widespread applicability in modeling failure data.

Alternative distribution families, such as phase-type distributions, were considered conceptually but not adopted in this study. Although phase-type models offer a high degree of flexibility and can approximate a broad class of distributions, their parameterization becomes increasingly complex as the number of phases grows, which can hinder interpretability and complicate estimation, particularly when combined with finite mixture structures. Given the objective of balancing modeling expressiveness with interpretability and computational tractability on large empirical datasets, the selected distribution families were deemed more appropriate for the scope of this work.

Failure records were analyzed to extract two features: application and failure cause. These classifications were constructed based on information available in the logs. Application refers to the specific executable program or system component where the failure occurred (e.g., `iexplore.exe`, `excel.exe`, or `U_Windows`). In contrast, the failure cause provides insight into the reason for the failure, such as memory errors or access violations.

The TBFs analyzed in this work were collected from several operational groups of computers running the same operating system. This common baseline allows the comparison of variability introduced by environmental or operational differences. These sources of heterogeneity in failure behavior were considered:

- ❑ **Operational Profile:** Differences in usage intensity and user interaction patterns can affect failure behavior across computers.
- ❑ **Failure Causes:** Software systems can experience failures due to multiple causes, such as software bugs, configuration errors, or interactions with external systems, and each cause may follow a distinct statistical distribution depending on the nature of the fault.

These factors can result in overlapping subpopulations within the data, which motivates the use of mixture models to uncover latent structures and to model complex failure distributions. To address these sources of heterogeneity and better understand the multimodal nature of failure patterns, two distinct modeling strategies were adopted:

Approach 1: Mixture models were applied to the complete set of TBFs for each computer, treating each machine as a single entity. This approach aims to capture patterns arising from the combination of different failure causes and usage behaviors present within a single system. To ensure meaningful statistical analysis, only computers with at least two sets of distinct applications failures, each comprising a minimum of 30 failure records, were included. This filtering resulted in 41 computers eligible for this analysis.

Approach 2: The dataset was segmented by application failure, and mixture models were applied separately to each one. This strategy focuses on understanding whether multimodal behavior is an inherent property of failures associated with specific applications, rather than the result of aggregated system behavior. To ensure meaningful statistical analysis of this modeling, only applications with at least 50 failure records per computer were considered. The only application in the dataset that satisfied this requirement across multiple computers was the Internet Explorer's process (`iexplore.exe`). Consequently, 36 computers with sufficient `iexplore` related failures were included in this second modeling approach.

The adoption of a minimum number of failure records per computer is motivated by both statistical and interpretative considerations. From a modeling perspective, mixture models require a sufficient number of observations to reliably estimate component parameters and mixing proportions. Applying the threshold at the computer level ensures that the inferred distributions reflect intrinsic failure behavior of individual systems rather than being dominated by sparse or incidental events. Moreover, this criterion promotes comparability across computers by reducing variability caused by unequal observation lengths or usage intensity.

In the context of application-level modeling, enforcing a per-computer threshold further guarantees that each application-specific failure process is supported by adequate empirical evidence, allowing multimodality to be interpreted as a structural characteristic of the failure mechanism rather than an artifact of data sparsity or random fluctuation.

Although the adoption of a minimum threshold of failure records per computer ensures statistical robustness for mixture model estimation, this criterion may introduce a selection bias at the computer level. Specifically, computers with low application usage or shorter observation periods are excluded from the analysis, even if they exhibit failure behaviors that are potentially relevant from a reliability perspective. As a result, the retained sample may be disproportionately composed of systems with higher usage intensity or greater instability, which tend to generate a larger number of recorded failures. This bias must be considered when interpreting the results, as the identified multimodal patterns may reflect the behavior of more active or failure-prone systems rather than being fully representative of the entire population of computers.

In total, 77 samples were selected for mixture modeling: 41 computers for Approach 1 and 36 for Approach 2. This strategy enables a comparative analysis of failure behavior from system-wide and cause-specific perspectives and allows the identification of statistical patterns across different abstraction levels.

It is also important to highlight that, unlike many studies that rely on simulated or constrained datasets, the data used in this study originate from a large empirical dataset collected from real-world systems, as reported by the research associated with this work (MATIAS et al., 2014).

4.3 Cluster and Statistical Analysis

To gain a deeper understanding of the failure patterns in each computer from the dataset, this study performed statistical and clustering analysis of the TBFs observed across all selected operational groups. The goal was to assess potential multimodal behavior in the data and to uncover complexity not captured by simple unimodal models.

For each computer selected for analysis, a set of statistical metrics was computed to characterize the TBF distribution. These include:

- ❑ **Failure Count:** Total number of recorded failures.
- ❑ **Central Tendency:** Mean, median, and mode.
- ❑ **Dispersion:** Standard deviation, interquartile range (IQR), and full range (minimum and maximum).
- ❑ **Distribution Shape:** Skewness and kurtosis.
- ❑ **Percentiles:** 25th and 75th percentiles to describe the data spread.

To investigate potential heterogeneity in failure behavior, clustering methods were applied considering only observations with $TBF > 0$. Multiple clustering algorithms were evaluated:

- ❑ **K-Means Clustering:** A widely used partitioning method that minimizes intra-cluster variance. It assumes spherical clusters and equal variance across groups. It is computationally efficient and interpretable, but sensitive to initialization and requires the number of clusters k to be defined a priori.
- ❑ **HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise):** A density-based method that automatically determines the number of clusters and identifies noise points. It handles clusters of varying shapes and densities but may be sensitive to its hyperparameters and struggles with low-density datasets.
- ❑ **Fuzzy C-Means (FCM):** Allows data points to belong to multiple clusters with varying degrees of membership. This is particularly suitable when failure patterns overlap or transition smoothly. However, FCM can be computationally intensive and sensitive to outliers.

- ❏ **Gaussian Mixture Models (GMMs):** Probabilistic models that assume the data is generated from a mixture of several Gaussian distributions. GMMs are flexible and can model elliptical clusters, but they are also sensitive to initialization and may overfit in high dimensions.

Each algorithm offers a unique perspective on the data. K-Means and FCM require specifying the number of clusters, while HDBSCAN and GMMs can estimate it from the data. The limitations of one method are often compensated by the strengths of another.

4.3.1 Cluster Evaluation Metrics

To evaluate and compare the clustering results, internal validation metrics were employed:

- ❏ **Silhouette Score:** Measures how similar a point is to its own cluster versus other clusters. Values range from -1 (incorrect clustering) to 1 (well-clustered).
- ❏ **Calinski-Harabasz Index:** Evaluates the ratio of between-cluster dispersion to within-cluster dispersion. Higher values indicate better-defined clusters.
- ❏ **Davies-Bouldin Index:** Measures average similarity between clusters, where lower values indicate better clustering.
- ❏ **Fuzzy Partition Coefficient (FPC):** Used for FCM, quantifies the degree of overlapping in fuzzy clusters; higher values indicate better partitions.
- ❏ **Bayesian Information Criterion (BIC)** and **Akaike Information Criterion (AIC):** Used with GMMs to evaluate model complexity versus fit. Lower values indicate more parsimonious models.

4.3.2 Evaluation and Selection of Clustering Configurations

To determine the most appropriate number of clusters (k), the `optimal_cluster_analysis` function was implemented. This function:

- ❏ Normalizes the input TBF data using `StandardScaler`.
- ❏ Iteratively applies each clustering method for different values of k (when required).
- ❏ Computes all internal evaluation metrics.
- ❏ Aggregates the results and proposes a consensus value for the optimal number of clusters based on the most frequently recommended k across all methods.

The complete clustering framework is summarized in Algorithm 1, which outlines the key preprocessing steps, statistical summarization, clustering execution, and evaluation procedures.

This clustering algorithm enables the identification of multimodal structure in the TBF data, serving as a basis for subsequent modeling. The following section presents a description of the Expectation-Maximization (EM) algorithm

4.4 Expectation-Maximization Algorithm

The Expectation-Maximization (EM) algorithm was implemented to fit finite mixture models to the observed data. The framework supports multiple parametric distributions and incorporates robust initialization procedures, convergence monitoring, and evaluation metrics to ensure efficient and reliable parameter estimation.

The general procedure of the Expectation-Maximization framework used in this study is summarized in Algorithm 2. This implementation iteratively alternates between the Expectation (E) step, which computes the responsibilities based on current parameter estimates, and the Maximization (M) step, which updates the mixture weights and distribution parameters using weighted maximum likelihood estimation. Convergence is monitored via the log-

Algorithm 1 Multimodality and Cluster Analysis Algorithm

Input: TBF data $X = \{x_1, x_2, \ldots, x_n\}$; max clusters $k_{\max}$
Output: Statistical summary, optimal k, clustering labels, evaluation metrics
Remove non-positive values: $X \leftarrow \{x \in X \mid x > 0\}$
Normalize: $X_{\text{scaled}} \leftarrow \text{StandardScaler}(X)$
Compute descriptive statistics: mean, std, skewness, etc.
K-Means:
for k in $[2, k_{\max}]$ **do**
    Apply K-Means and compute silhouette, CH, DB indices
end for
Use Elbow Method and Silhouette Score to estimate optimal k
HDBSCAN: Run once and extract clusters (excluding noise)
Fuzzy C-Means:
for k in $[2, k_{\max}]$ **do**
    Apply FCM, extract hard labels and FPC
    Evaluate with silhouette, CH, DB
end for
GMMs:
for k in $[1, k_{\max}]$ **do**
    Fit GMM, compute BIC, AIC, silhouette
end for
Aggregate results and determine consensus k
Generate visualizations: histograms, density plots, cluster assignments
return Statistical summaries, clustering results, evaluation scores

likelihood improvement, and the process continues until a predefined tolerance (`tol` = 10^{-4}) or iteration limit is reached (`max_iter` = 1000).

Five probability distributions were integrated into the Expectation-Maximization algorithm to accommodate the diverse characteristics of TBF data: Exponential, Normal, Log-normal, Weibull and Gamma. Each distribution was selected based on its suitability for modeling different failure behaviors observed in software reliability.

During the M-step of the EM algorithm, distribution-specific parameters are estimated using weighted statistics derived from the responsibilities computed in the E-step. These weights reflect the degree to which each data point is associated with each component. The estimation procedure for each distribution is described below, highlighting the underlying assumptions and the corresponding parameter updates within the EM framework.

Exponential distribution: Characterized by a single scale parameter λ, it assumes a constant failure rate and is commonly used for memoryless processes. The parameter is estimated via weighted maximum likelihood, using the expression $\hat{\lambda} = \max(10^{-10}, \bar{x}_w)$, where $\bar{x}_w$ is the weighted sample mean. A lower bound is enforced to prevent degenerate solutions.

Normal distribution: Parameterized by mean μ and standard deviation σ, it is suitable for modeling symmetric TBF patterns. The estimates are computed from the weighted sample statistics as $\hat{\mu} = \bar{x}_w$ and $\hat{\sigma} = \sqrt{\text{Var}_w(x)}$, where $\text{Var}_w(x)$ denotes the weighted variance.

Log-normal distribution: Well-suited for right-skewed data arising from multiplicative processes, it is parameterized by shape, location, and scale. The estimation is performed using the lognorm.fit routine from SciPy, incorporating the component responsibilities as frequency weights.

Weibull distribution: Known for its flexibility in representing increasing, decreasing, or constant failure rates depending on its shape parameter. Parameters are estimated via numerical maximum likelihood methods, without closed-form expressions, using numerical solvers initialized from method-of-moments estimates.

Gamma distribution: Useful for modeling skewed TBF data with variable failure intensities. Parameter estimation also relies on numerical maximum likelihood, and due to the involvement of special functions, this distribution often requires greater computational effort during the fitting process.

4.4.1 Computational Complexity Analysis

Each supported distribution was analyzed in terms of its computational demands. These demands were quantified by counting the number of floating-point operations required for evaluating the probability density function, cumulative distribution function, and parameter updates during the M-step.

The Exponential distribution has the lowest computational cost, due to its closed-form expressions. In contrast, the Gamma and Weibull distributions are more computationally intensive, as they rely on iterative or numerical procedures.

Algorithm 2 Expectation-Maximization Algorithm

Require: Data $X = \{x_1, ..., x_N\}$, List of distributions, Maximum iterations T_{max}, Tolerance ϵ

Ensure: Optimized parameters θ, Log-likelihood history $\mathcal{L}$

Initialization:
$\theta^{(0)} \leftarrow$ initialize_parameters$(X,$ distributions$)$ ▷ Uses K-means
$\mathcal{L} \leftarrow []$
Initial evaluation:
$\ell^{(0)} \leftarrow$ calculate_log_likelihood$(X, \theta^{(0)})$
$\mathcal{L}$.append$(\ell^{(0)})$
for $t \leftarrow 1$ to T_{max} **do**
    E-step: ▷ Compute responsibilities
    $\gamma \leftarrow$ zeros(N, K)
    log_pdfs $\leftarrow$ calculate_log_pdfs$(X, \theta^{(t-1)})$
    for $i \leftarrow 1$ to N **do**
        log_weighted $\leftarrow$ log_pdfs$[i] + \log($weights$)$
        max_val $\leftarrow \max($log_weighted$)$
        exp_vals $\leftarrow \exp($log_weighted $-$ max_val$)$
        $\gamma[i] \leftarrow$ exp_vals$/\sum($exp_vals$)$
    end for
    M-step: ▷ Update parameters
    for $k \leftarrow 1$ to K **do**
        effective_samples $\leftarrow \sum_{i=1}^{N} \gamma[i, k]$
        weights$[k] \leftarrow$ effective_samples$/N$
        if distribution $==$ "exp" **then**
            scale $\leftarrow \sum \gamma[:, k] \cdot X /$effective_samples
            $\theta_k \leftarrow (\max(1e-10,$ scale$),)$
        else if distribution $==$ "norm" **then**
            loc $\leftarrow \sum \gamma[:, k] \cdot X /$effective_samples
            scale $\leftarrow \sqrt{\sum \gamma[:, k] \cdot (X - \text{loc})^2 /\text{effective\_samples}}$
            $\theta_k \leftarrow$ (loc, scale)
        else if distribution $==$ "logn" **then**
            $\theta_k \leftarrow$ lognorm.fit$(X,$ fscale $= \gamma[:, k])$
        end if
    end for
    Convergence check:
    $\ell^{(t)} \leftarrow$ calculate_log_likelihood$(X, \theta^{(t)})$
    $\mathcal{L}$.append$(\ell^{(t)})$
    if $|\ell^{(t)} - \ell^{(t-1)}| < \epsilon$ **then**
        break
    end if
    Monitoring:
    check_memory_usage()
    snapshot_performance()
end for
Return $\theta^{(t)}, \mathcal{L}$

Table 3 summarizes the estimated number of operations required for each distribution during the EM algorithm. These estimates were obtained through a combination of analysis of the analytical expressions involved, such as the probability density functions, cumulative distribution functions, and parameter update formulas and empirical profiling of the implementation.

The analysis considered the functional forms of the PDFs and CDFs for each distribution, as well as representative expressions used in the parameter updates during the M-step. The main formulas are listed below:

❑ **Exponential:**

$$f(x;\lambda) = \lambda e^{-\lambda x}, \quad F(x;\lambda) = 1 - e^{-\lambda x}, \quad x \geq 0$$

M-step (MLE for rate parameter):

$$\hat{\lambda} = \frac{\sum_{i=1}^{n} \gamma_i}{\sum_{i=1}^{n} \gamma_i x_i}$$

❑ **Normal:**

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad F(x;\mu,\sigma) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right]$$

M-step (weighted sample mean and variance):

$$\hat{\mu} = \frac{\sum_{i=1}^{n} \gamma_i x_i}{\sum_{i=1}^{n} \gamma_i}, \quad \hat{\sigma}^2 = \frac{\sum_{i=1}^{n} \gamma_i (x_i - \hat{\mu})^2}{\sum_{i=1}^{n} \gamma_i}$$

❑ **Log-normal:**

$$f(x;\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right), \quad F(x;\mu,\sigma) = \Phi\left(\frac{\ln x - \mu}{\sigma}\right), \quad x > 0$$

M-step (weighted log-transform):

$$\hat{\mu} = \frac{\sum_{i=1}^{n} \gamma_i \ln x_i}{\sum_{i=1}^{n} \gamma_i}, \quad \hat{\sigma}^2 = \frac{\sum_{i=1}^{n} \gamma_i (\ln x_i - \hat{\mu})^2}{\sum_{i=1}^{n} \gamma_i}$$

❑ **Weibull:**

$$f(x;k,\lambda) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} \exp\left(-\left(\frac{x}{\lambda}\right)^{k}\right), \quad F(x;k,\lambda) = 1 - \exp\left(-\left(\frac{x}{\lambda}\right)^{k}\right), \quad x \geq 0$$

M-step: Closed-form solutions are not available; parameters are updated via numerical methods using weighted log-likelihood maximization.

Table 3 – Estimated Computational Complexity for Each Distribution.

Distribution	**PDF Operations**	**CDF Operations**	**M-step Operations**
Exponential	5	7	10
Normal	15	20	25
Lognormal	18	22	30
Weibull	20	25	35
Gamma	22	28	40

❑ **Gamma:**

$$f(x;\alpha,\beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, \quad F(x;\alpha,\beta) = \frac{1}{\Gamma(\alpha)} \gamma(\alpha,\beta x), \quad x > 0$$

M-step: As with the Weibull distribution, closed-form updates are not available. The shape and rate parameters are estimated numerically, often using weighted maximum likelihood with iterative solvers.

4.4.2 Initialization Strategy

Effective parameter initialization is critical for the convergence and performance of EM algorithms, therefore this implementation employs a strategy based on K-means clustering. The dataset is first partitioned into clusters corresponding to the desired number of mixture components. The centroids and weights of these clusters are then used to generate initial guesses for the component parameters.

Initialization is tailored to each distribution. For instance, in the case of the Exponential distribution, the scale parameter is initialized based on the median of the assigned data. For the Normal distribution, the initial mean and variance are set using the cluster's sample mean and standard deviation. More complex distributions such as the Log-normal, Weibull, and Gamma use moments derived from the cluster data. Finally, component weights are normalized to ensure they sum to one.

4.4.3 EM Algorithm Implementation

The EM loop consists of two main steps: the Expectation (E-step) and Maximization (M-step).

In the E-step, posterior probabilities (or responsibilities) are calculated for each data point with respect to each component. This is done using the numerically stable log-sum-exp formula:

$$r_{ik} = \frac{w_k \cdot p(x_i|\theta_k)}{\sum_{j=1}^{K} w_j \cdot p(x_i|\theta_j)} \tag{10}$$

These responsibilities quantify the degree to which each observation is associated with each component of the mixture model.

In the M-step, the model parameters are updated using the responsibilities from the E-step. Mixture weights are updated as:

$$w_k^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} r_{ik}^{(t)} \tag{11}$$

The parameters of each component distribution are re-estimated by maximizing the expected log-likelihood:

$$\theta_k^{(t+1)} = \arg\max_{\theta} \sum_{i=1}^{N} r_{ik}^{(t)} \log p(x_i|\theta) \tag{12}$$

All updates include bound checks and exception handling to avoid numerical instabilities or invalid parameter values.

4.4.4 Incremental Components for Mixture Model

To determine the optimal number of mixture components, an automated model selection routine was implemented through the `incremental_sensitivity_analysis` function. This function is designed to balance model accuracy and complexity, guiding the Expectation-Maximization algorithm toward a parsimonious yet effective configuration. Its operation is divided into two main phases:

Initial Exploration Phase

The function begins with an exhaustive evaluation of all possible combinations of the selected distributions, varying the number of mixture components from 1 up to a predefined base limit (default is 5). For each component count, it computes goodness-of-fit (GoF) scores using four metrics: Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Kolmogorov-Smirnov (KS), and Anderson-Darling (AD). At each step, the three best-performing models per metric are retained and stored for subsequent use.

Incremental Expansion Phase

From the base limit up to a maximum number of components (default is 30), the function incrementally generates more complex models by expanding the top-performing configurations. One distribution is added at each iteration, and new models are evaluated using the same set of GoF metrics. The expansion process is guided by a stopping criterion defined by one or more of the following conditions:

- ❏ The improvement in a GoF score falls below a predefined threshold (default is 0.0001);
- ❏ A GoF score deteriorates consistently across three consecutive steps;
- ❏ The maximum number of components is reached.

When a stopping criterion is met for two or more metrics, the expansion process is halted.

Final Selection and Output

At the conclusion of the analysis, the function determines the final optimal number of components by averaging the component counts associated with the best scores across all four metrics. A summary is printed detailing the reasons for stopping in each case. The function returns four key outputs: (1) a consolidated dictionary of the best model configurations per component count, (2) the GoF scores across the entire process, (3) the top models per metric at each component level, and (4) the final recommended number of components. This procedure provides a structured approach to mixture model selection, balancing statistical fit and computational feasibility.

4.4.5 Convergence and Optimization Strategies

Several optimization strategies are employed to ensure convergence and improve robustness. The algorithm executes multiple runs (default is ten) with different initialization to avoid local optima. Convergence is assessed based on changes in log-likelihood, and an early stopping criterion is applied if improvements fall below a predefined threshold. Memory usage is also monitored, with process termination if consumption exceeds 2GB. Additionally, a maximum number of 1000 iterations is enforced to prevent infinite loops.

4.4.6 Sensitivity Analysis and Computational Cost

A sensitivity study was conducted to investigate the impact of the number of mixture components on model performance. In this analysis, plots were constructed with the number of mixture components on the x-axis and the values of each applied Goodness-of-Fit (GoF) test on the y-axis.

By observing the behavior of these GoF statistics as the number of components increases, it is possible to identify the point at which adding more components no longer yields significant improvements in model fit. Typically, a sharp decrease followed by stabilization or even deterioration in the criteria suggests an optimal number of components.

To complement the sensitivity analysis, a computational cost study was conducted to evaluate the resource implications of increasing model complexity. A monitoring class tracks system performance throughout model fitting, it measures execution time, peak memory consumption, average CPU utilization, and estimates operation counts based on the complexity of the chosen distributions.

The cost of model estimation was assessed across different numbers of mixture components and types of distributions, both homogeneous mixtures (same distribution type across all components) and heterogeneous mixtures (combining different distribution families within a single model). The evaluation considered four key computational metrics:

- ❑ **Execution time:** Total runtime of the EM algorithm until convergence.

- ❏ **Memory usage:** Peak memory consumption observed during model estimation.
- ❏ **CPU usage:** Average percentage of CPU utilization throughout the EM iterations.
- ❏ **Number of operations:** A theoretical estimate of computational effort, derived from symbolic expressions of the model's structure.

Each distribution included in the mixture model imposes a distinct computational cost, which was estimated using a predefined dictionary of complexity factors:

- ❏ **PDF Operations:** Number of operations required to evaluate the probability density function.
- ❏ **CDF Operations:** Number of operations needed to compute the cumulative distribution function.
- ❏ **M-Step Operations:** Estimated number of operations required to update the parameters during the M-step of the EM algorithm.

For each mixture configuration, two types of plots were generated:

- ❏ **Homogeneous Mixtures:** For mixtures composed exclusively of a single distribution type, the computational cost was evaluated by incrementally increasing the number of components.
- ❏ **Heterogeneous Mixtures:** In the case of mixtures combining different distribution types, the overall computational cost was estimated by averaging the predefined complexity factors of the selected distributions

This analysis provides insights into the trade-offs between model complexity and computational efficiency. While models with more components (or with diverse distribution families) tend to offer better fit, they also demand more computational resources.

Chapter 5

Results

This chapter presents the results of the analysis conducted on failure data. Two distinct modeling approaches were employed to explore different perspectives on failure behavior and to assess the presence of patterns in time between failures.

5.0.1 Sample Selection and Data Preparation

Following the filtering approaches defined for each modeling strategy, a total of 77 samples were selected for mixture model analysis.

Approach 1: In this approach, mixture models were applied to the entire set of TBFs recorded for each computer, treating each machine as a single analytical unit. To ensure statistical significance, only computers with at least two distinct failure causes, each comprising a minimum of 30 failure records, were included. This criterion yielded a total of 41 eligible computers for analysis.

Approach 2: The dataset was segmented by failure type and failure cause. To ensure statistical significance, only failure causes with at least 50 recorded failures per computer were considered. As a result, 36 computers with a sufficient number of failures were selected for this second modeling approach.

5.0.2 Occurrence of Zero-Valued TBFs

During the preprocessing stage, the number and proportion of TBF values equal to zero were quantified for each operational group. These values were not removed from the dataset but were disregarded in the clustering and mixture modeling analyses. Table 4 summarizes the total number of TBF entries and the corresponding amount and percentage of zero-valued TBFs observed in each group.

5.0.3 Approach 1: Complete TBF Analysis

Table 5 lists the 41 computers selected for Approach 1, where mixture models were applied to the complete set of time-between-failures for each system.

Table 4 – Number and Proportion of Zero-Valued TBFs in Each Group.

Group	Total TBFs	Zero-valued TBFs	Percentage of Zeros (%)
G1	63	0	0.00
G2	8360	39	0.47
G3	1100	6	0.55
G4	3971	98	2.47

Table 5 – The Selected Computers for Approach 1 Analysis.

Computer IDs	
G2Rac_MC-151157	G2Rac_MC-157457
G2Rac_MC-156359	G2Rac_MC1-035
G2Rac_MC-157380	G2Rac_MC1-051
G2Rac_MC-157383	G2Rac_MC1-074
G2Rac_MC-157396	G2Rac_MC1-097
G2Rac_MC-157398	G2Rac_MC1-112
G2Rac_MC-157400	G2Rac_MC2-070
G2Rac_MC-157406	G2Rac_MC2-075
G2Rac_MC-157413	G2Rac_MC2-100
G2Rac_MC-157416	G2Rac_MC3-196
G2Rac_MC-157430	G3Rac_DSK021
G2Rac_MC-157432	G3Rac_DSK023
G2Rac_MC-157435	G3Rac_MB021
G2Rac_MC-157445	G41174137528899245fbeab8db8aa9320
G2Rac_MC-157447	G4117549040377494610 8f63bd34b5937
	G411774790268767462ee772d61026018
	G411814383442168466b518834f513970
	G4133483408722174f8ff3a7362535189
	G4133510115532394f9406e34f1c62093
	G4133535507490744f97e6c2dd8ed3561
	G413431410151584500eb49726b174735
	G41343155940620 6500eeee4978883010
	G413452266836854502e87bba75907480
	G4134852921215845060ec3c26b1e1089
	G413485294771326 5060ed45206293835
	G41353517451878650ad098bd68643260

5.0.4 Approach 2: Internet Explorer Failure Analysis

Notably, Internet Explorer was the only recurrent failure type found across all computer groups analyzed, which justified its selection for a more detailed investigation. For Approach 2, computers in the original dataset were examined to identify those containing Internet Explorer with multiple failure causes occurring in samples larger than 50 records. It was possible to identify distinct failure causes. For each computer meeting the criteria, failure records were filtered to isolate only a specific failure, excluding all other failure types from that computer's dataset. Time-between-failures values were then recalculated based solely on these filtered records.

Table 6 presents the 36 samples selected for Approach 2, focusing specifically on Internet Explorer (`iexplore.exe`) related failures with their corresponding error codes. This filtering process ensures that the mixture modeling analysis for Approach 2 reflects the pure temporal characteristics of individual failure causes, eliminating the effects of other failure types that might distort the pattern of the target failure mechanism.

The analysis of Internet Explorer failures revealed four distinct failure causes. Table 7 presents the identified failure causes along with their technical descriptions and classifications (SANTOS; MATIAS; TRIVEDI, 2021).

The distribution of these failure causes across the 36 selected samples is presented in Table 8. This table reveals the presence of each failure cause within the Internet Explorer failures.

The predominance of access violation errors (c0000005) accounts for approximately 80.6% of all Internet Explorer failure samples, indicating that memory addressing issues represent the most common cause of Internet Explorer failures in the analyzed dataset. Memory management issues (heap corruption) and security violations (privileged instruction) each contribute 8.3% of the samples, while unhandled C++ exceptions represent the least frequent cause at 2.8% of the samples.

5.0.5 Sample Analysis

For detailed presentation and analysis in this work, a subset of 10 representative samples was selected to illustrate the key findings from both approaches. The selection strategy prioritized computers that were included in both Approach 1 and Approach 2 analyses, enabling direct comparison between system-wide and cause-specific failure behaviors for the same systems. Table 9 shows the selected samples, highlighting the paired analysis structure.

Each analysis includes the following components: (1) descriptive statistics summarizing the temporal characteristics of the failure data, (2) cluster identification results from multiple unsupervised learning algorithms, (3) mixture model selection, validation through goodness-of-fit tests and sensitivity analysis, (4) parameter estimation for the selected mixture components, (5) comparative analysis against simple distribution models, and (6) computational cost evaluation for the modeling process.

5.0.6 G1Rac_MC0-018_iexplore_c0000005 (Approach 2)

This case refers to failures related to access violations (c0000005), associated with invalid memory adressing, occurring in the `iexplore.exe`'s process on computer MC0-018, part of the Group 1 (Undergraduate Laboratory Computers).

Table 6 – The Selected Samples for Approach 2 Analysis (Internet Explorer).

Computer ID and Failure Cause
G1Rac_MC0-018_iexplore_c0000005
G2Rac_MC-153821_iexplore_c0000005
G2Rac_MC-153873_iexplore_c0000005
G2Rac_MC-157334_iexplore_c0000005
G2Rac_MC-157380_iexplore_c0000005
G2Rac_MC-157396_iexplore_c0000005
G2Rac_MC-157399_iexplore_c0000005
G2Rac_MC-157400_iexplore_c0000005
G2Rac_MC-157400_iexplore_c0000374
G2Rac_MC-157431_iexplore_c0000005
G2Rac_MC-157431_iexplore_c0000096
G2Rac_MC-157435_iexplore_c0000005
G2Rac_MC1-033_iexplore_c0000005
G2Rac_MC1-035_iexplore_c0000005
G2Rac_MC1-041_iexplore_c0000005
G2Rac_MC1-074_iexplore_c0000005
G2Rac_MC1-074_iexplore_c0000096
G2Rac_MC1-076_iexplore_c0000005
G2Rac_MC1-083_iexplore_c0000005
G2Rac_MC1-085_iexplore_c0000005
G2Rac_MC1-085_iexplore_c0000374
G2Rac_MC1-086_iexplore_c0000005
G2Rac_MC1-089_iexplore_c0000096
G2Rac_MC1-094_iexplore_c0000005
G2Rac_MC1-096_iexplore_c0000096
G2Rac_MC1-102_iexplore_c0000005
G2Rac_MC2-046_iexplore_c0000005
G2Rac_TANNER_iexplore_c0000005
G2Rac_TANNER_iexplore_c0000374
G3Rac_DSK023_iexplore_c0000005
G3Rac_WKS138_iexplore_c0000005
G4133535507490744f97e6c2dd8ed3561_iexplore_c0000005
G4134340997945745012cf3b6fb106680_iexplore_c0000005
G413452266836854502e87bba75907480_iexplore_e06d7363
G4134852947713265060ed45206293835_iexplore_c0000005
G4134852968526425060ee15408575705_iexplore_c0000005

5.0.6.1 Statistical Characterization

Table 10 presents the statistical characteristics of the TBF data for this sample. The descriptive statistics reveal a heterogeneous failure pattern characterized by significant asymmetry and heavy-tailed behavior. The difference between mean (107.13 hours) and median (23.83 hours) indicates a right-skewed distribution, which is further confirmed by the positive skewness value of 2.22.

The extremely low mode value (0.0039 hours) combined with the minimum observation (0.0011 hours) indicates that some failures occur in rapid succession, possibly representing cascading failures or immediate re-failures after system recovery attempts.

The high kurtosis value (5.77) indicates a distribution with heavy tails and a sharp peak, suggesting the coexistence of multiple modes. The large interquartile range (143.54 hours)

Table 7 – Internet Explorer Failure Causes and Their Classification.

Error Code	**Failure Cause**	**Failure Context**
c0000005	Access violation	Memory addressing
c0000374	Heap corruption	Memory management
c0000096	Privileged instruction	Processor (rules) violation
e06d7363	Unhandled C++ exception	Exception handling

Table 8 – Distribution of Internet Explorer Failure Causes.

Error Code	**Failure Cause**	**Number of Samples**
c0000005	Access violation	29
c0000374	Heap corruption	3
c0000096	Privileged instruction	3
e06d7363	Unhandled C++ exception	1
Total		**36**

Table 9 – Representative Samples Selected for Detailed Analysis.

Group	**Approach**	**Sample ID**
G1	2	G1Rac_MC0-018_iexplore_c0000005
G2	1	G2Rac_MC-157400
	2	G2Rac_MC-157400_iexplore_c0000374
G2	1	G2Rac_MC1-074
	2	G2Rac_MC1-074_iexplore_c0000005
	2	G2Rac_MC1-074_iexplore_c0000096
G3	1	G3Rac_DSK023
	2	G3Rac_DSK023_iexplore_c0000005
G4	1	G413452266836854502e87bba75907480
	2	G413452266836854502e87bba75907480_iexplore_e06d7363

relative to the median further emphasizes the high variability in failure times.

5.0.6.2 Clustering Analysis

To support the selection of the number of mixture components, clustering algorithms were applied to the dataset. Table 11 summarizes the number of clusters recommended by each approach.

HDBSCAN's recommendation of 4 clusters suggests the presence of four major density-based groupings. In contrast, K-Means and Fuzzy C-means algorithms suggest a higher number of clusters (12 and 11, respectively), indicating finer granularity in failure pattern recognition. The Gaussian Mixture Model (GMM) approaches, using both BIC and AIC, recommend intermediate values of 9 and 10 components, respectively.

Figure 7 provides the visualization of the clustering analysis results across five different algorithms. The scatter plots reveal distinct clustering behaviors: K-Means (k=12) identifies numerous small clusters with many observations concentrated in the low TBF region (0-50 hours) and several isolated high-TBF outliers. HDBSCAN (k=4) shows a more conservative approach, identifying four main clusters with a significant portion of data classified as noise (gray crosses), particularly in the intermediate TBF range.

Table 10 – Descriptive Statistics: G1Rac_MC0-018_iexplore_c0000005.

Statistic	**Value**
Count	63
Mean (hours)	107.13
Median (hours)	23.83
Mode (hours)	0.0039
Standard Deviation	157.32
Minimum	0.0011
Maximum	813.91
First Quartile (Q1)	2.42
Third Quartile (Q3)	145.96
Interquartile Range (IQR)	143.54
Skewness	2.22
Kurtosis	5.77
Main Data Range	0.0011 – 387.63

Table 11 – Cluster Results: G1Rac_MC0-018_iexplore_c0000005.

Clustering Approach	**Recommended Clusters**
K-Means	12
HDBSCAN	4
Fuzzy C-means	11
GMM (BIC)	9
GMM (AIC)	10

Fuzzy C-Means (k=11) produces a clustering pattern similar to K-Means but with slightly different cluster boundaries, while the GMM approaches (BIC with k=9 and AIC with k=10) show intermediate clustering between HDBSCAN's conservative grouping and K-Means' fine partitioning. The Elbow and Silhouette analysis in the bottom-right panel demonstrates the trade-off between cluster quality metrics, with the Elbow method showing a sharp decline up to 4 components followed by gradual improvement, while the Silhouette score peaks around 2-3 clusters.

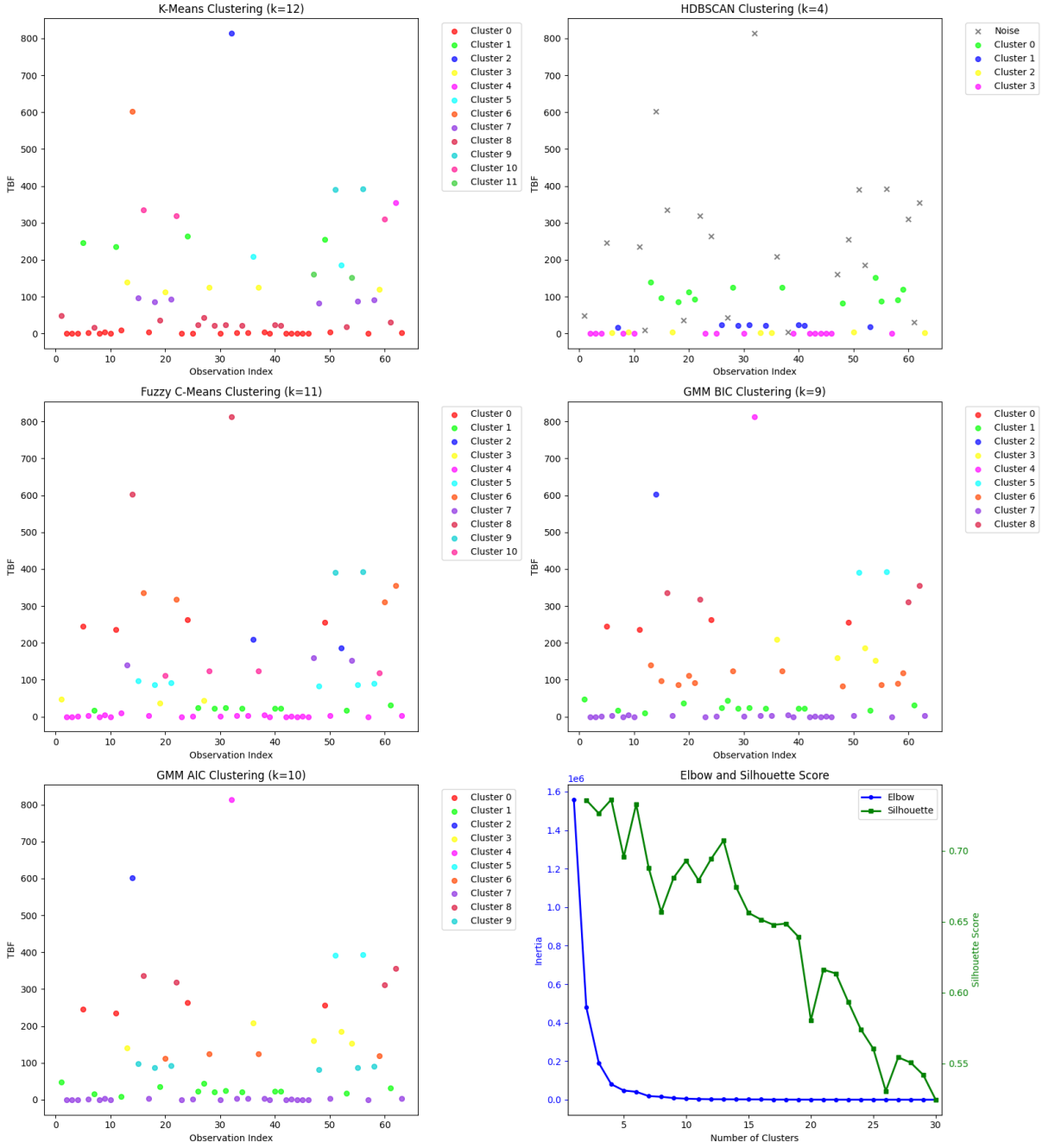

Figure 7 – Cluster Evaluation Plots for the Sample G1Rac_MC0-018_iexplore_c0000005.

5.0.6.3 Mixture Model Selection and Validation

The selection of the best-fitting mixture model was based on the top five models identified by each GoF test, combined with a visual comparison of the graphical results for each candidate model. This procedure was applied across all samples. The final model selected via the Expectation-Maximization (EM) algorithm consisted of 12 components with the following distributional structure: logn-logn-logn-logn-logn-gamma-logn-norm-logn-norm-logn-norm. This configuration was selected based on the outcomes of the KS goodness-of-fit test.

Figure 8 demonstrates the fit achieved by the 12-component mixture model. The probability density function plot shows a highly multimodal distribution with a dominant peak near zero and several smaller peaks at intermediate values (50-400 hours) and longer intervals (600-800 hours).

The cumulative distribution function comparison reveals agreement between the empirical and theoretical distributions, with the theoretical CDF (orange crosses) closely overlaying

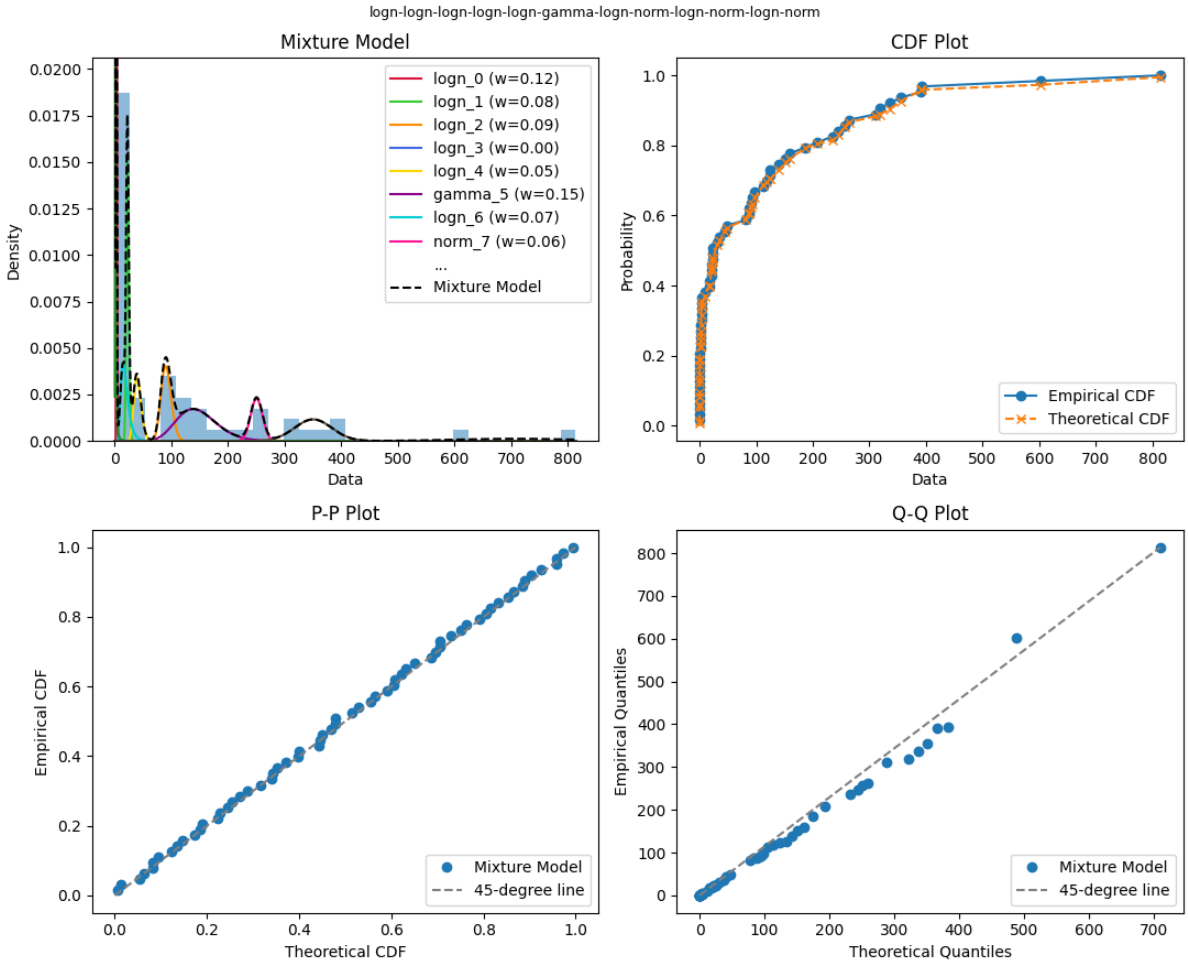

Figure 8 – G1Rac_MC0-018_iexplore_c0000005 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

the empirical CDF (blue circles) across the entire range. The probability-probability (P-P) plot shows linearity along the 45-degree line, indicating that the mixture model accurately represents the cumulative probability structure at all quantile levels.

The quantile-quantile plot further confirms model adequacy, with linear correspondence between theoretical and empirical quantiles. Minor deviations are observed only at the middle upper tail (around 400 hours). The overall linear pattern validates the mixture model's ability to capture both the central tendency and the extreme behavior of the failure process.

Table 12 summarizes the goodness-of-fit results for the selected model. The GoF test results provide statistical evidence for the adequacy of the 12-component mixture model. The KS test yields a low test statistic (0.0290) with an extremely high p-value (0.999999983), indicating that the null hypothesis of distributional equivalence cannot be rejected with confidence. The AD test, which is more sensitive to deviations in the distribution tails, produces a test statistic of 0.279 with a p-value of 0.4875. This result further confirms the model adequacy.

The log-likelihood value of -261.50 reflects the model's overall fit and provides a basis for comparison with alternative models. In addition, the AIC and the BIC values are 610.99 and 705.29, respectively, these criteria penalize model complexity.

Having established the adequacy of the selected mixture model in terms of global fit, the next step is to examine the estimated parameters associated with each mixture component. Table 13 presents the estimated parameters of the fitted mixture model.

Each distribution follows its standard statistical parameterization, respectively:

- ❏ Normal distribution uses the location (μ) and scale (σ) parameters.
- ❏ Lognormal distribution is parameterized by the shape (σ), location (μ), and scale (θ) parameters.
- ❏ Gamma distribution is characterized by the shape (α), location (μ), and scale (θ) parameters.

The mixture weights indicate the relative importance of each failure subpopulation, with several dominant components: logn_10 (14.25%), gamma_5 (15.00%), and logn_0 (12.09%). These high-weight components correspond to different failure regimes, from rapid succession

Table 12 – Goodness-of-fit Test Results: G1Rac_MC0-018_iexplore_c0000005.

Model	**Test**	**Statistic / p-value**
12-Component Mixture	KS	0.0290 / 0.999999983
	AD	0.279 / 0.4875
	Log-Likelihood	-261.50
	AIC	610.99
	BIC	705.29

failures (logn_10 with scale parameter 0.0034) to intermediate stability periods (gamma_5 with shape parameter 17.0351).

The lognormal components dominate the mixture (8 out of 12 components), reflecting the multiplicative nature of the failure process. The presence of three normal components (norm_7, norm_9, norm_11) with high location parameters (250.37, 350.19, and 708.12 hours, respectively) captures the extended TBFs.

5.0.6.4 Sensitivity Analysis

To assess the robustness of the selected 12-component model and evaluate how model performance responds to changes in complexity, a sensitivity analysis was performed. This analysis investigated the behavior of GoF metrics as the number of mixture components increased. The sensitivity analysis was conducted by plotting the best GoF statistics obtained for each number of mixture components.

Figure 9 presents the sensitivity analysis results, examining the behavior of key goodness-of-fit statistics across component numbers. The KS test sensitivity analysis reveals a performance trajectory that supports the 12-component selection. Starting from an single-component performance, the statistic shows improvement through the initial component additions, dropping sharply for two components. The most significant improvement occurs in the 2-8 component range, with continued refinement leading to optimal performance in the 8-13 component region.

The sensitivity of the AD test provides complementary evidence by emphasizing distribution's tail behavior. In the single-component model, the AD statistic highlights the inadequacy of simple models. As the number of components increases, the AD statistic transitions to negative values, indicating an improvement in the model's ability to represent distribution's tail behavior more accurately in multi-component configurations.

Table 13 – Mixture Model Parameters: G1Rac_MC0-018_iexplore_c0000005.

Component	Distribution	Weight	Param 1	Param 2	Param 3
logn_0	Lognormal	0.1209	0.1720	1E-10	3.07
logn_1	Lognormal	0.0845	0.1000	1E-10	22.47
logn_2	Lognormal	0.0910	0.1000	1E-10	90.49
logn_3	Lognormal	0.0009	1.0070	1E-10	56.39
logn_4	Lognormal	0.0537	0.1593	1E-10	40.13
gamma_5	Gamma	0.1500	17.0351	1E-10	8.63
logn_6	Lognormal	0.0666	0.3973	1E-10	17.34
norm_7	Normal	0.0595	250.37	10.38	—
logn_8	Lognormal	0.1027	1.3149	1E-10	0.33
norm_9	Normal	0.0958	350.19	32.87	—
logn_10	Lognormal	0.1425	0.6548	1E-10	0.0034
norm_11	Normal	0.0318	708.12	105.81	—

Information criteria sensitivity analysis reveals the trade-off inherent in model selection. The AIC trajectory shows improvement through 3-7 components, followed by a stagnation due to complexity penalties. The minimum AIC value was observed with 14 components. In contrast, the BIC analysis, which imposes a stronger penalty for model complexity, follows a similar initial trend but reaches its minimum between 6 and 8 components.

Additionally, graphical analyses of mixture model fits were conducted by inspecting the best-fitting result according to each individual GoF test. A notable trend emerged. The distributions selected as optimal by each test: AD, KS, AIC, and BIC rarely coincided. In particular, the models selected by AIC and BIC consistently exhibited poorer visual and statistical fit compared to those chosen by the AD and KS tests. This performance gap persisted even in the cases where AIC and BIC selected models with a higher number of components.

5.0.6.5 Comparison with Single-Component Models

For comparison, the best-fitting single distribution was the gamma model, as shown in Table 14. Although it achieved the best performance among the single-component models

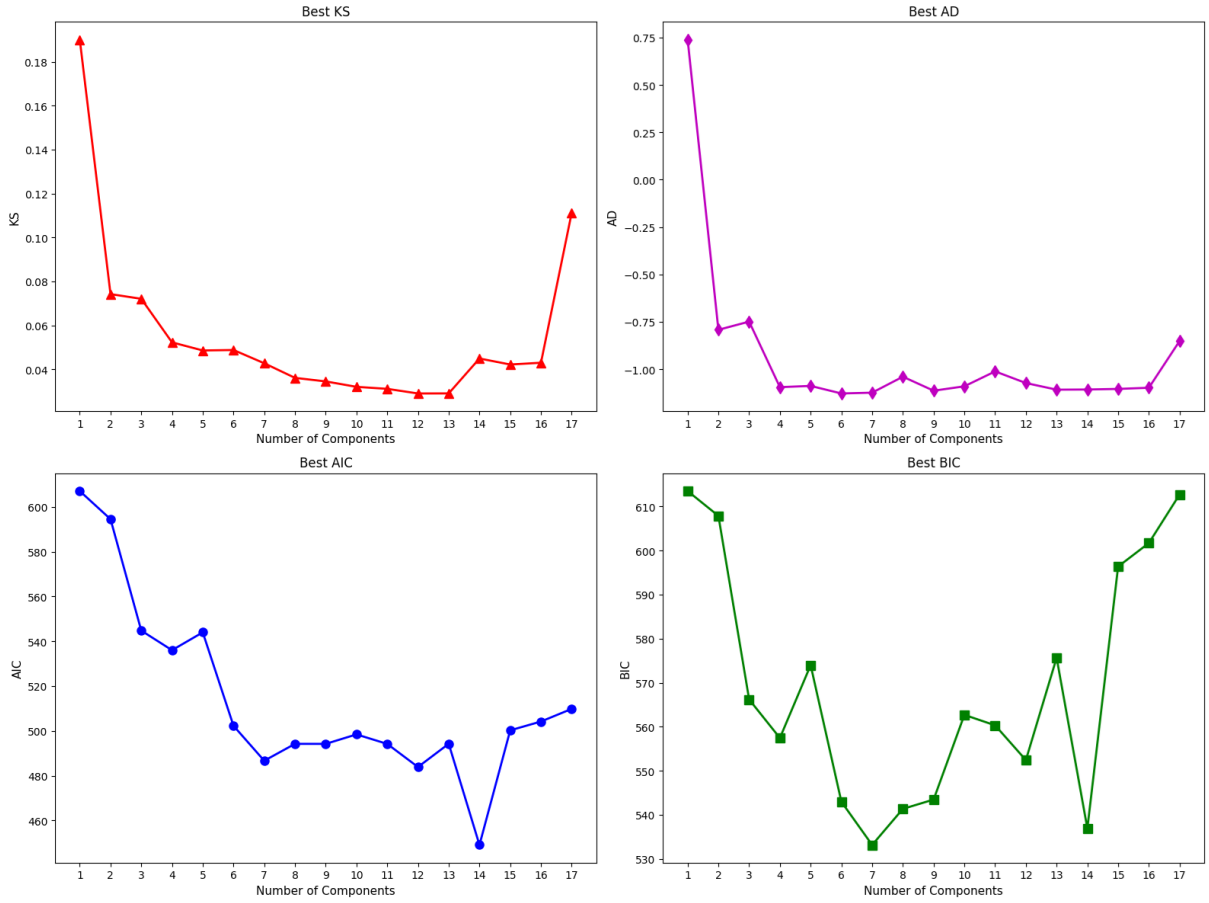

Figure 9 – G1Rac_MC0-018_iexplore_c0000005 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components.

tested, it still failed to capture the complexity present in the data.

While the gamma distribution emerged as the best-fitting single-component model among those evaluated, it remains inadequate for representing the complex behavior of time between failures. Previous studies on software reliability modeling have relied on single-distribution models, such as (LYU; NIKORA, 1991; MATIAS et al., 2014; XU; KALBARCZYK; IYER, 1999; MATIAS; OLIVEIRA; ARAUJO, 2013; HSU; HUANG, 2014; PHAM, 2003; MULLEN, 1998). These models often fail to capture the underlying structural characteristics that arise from the inherent complexity of TBFs, which can result from multiple factors, including the diversity of failure causes and variations in workload and operational profiles.

Goodness-of-fit testing highlights this difference in representational adequacy: while the gamma distribution yields extremely low p-values for both the Kolmogorov–Smirnov (KS = 0.3327, p = 1.37E-16) and Anderson–Darling (AD = 16.67, p = 0.000999) tests, indicating clear rejection of the distributional hypothesis. The mixture model achieves much stronger agreement with the data (KS = 0.0290, p = 0.999999983; AD = 0.279, p = 0.4875). And the log-likelihood comparison also shows the superior fit of the mixture model (-261.50 vs. -436.80).

The information criteria (AIC and BIC) further reinforce the inadequacy of the single distribution. As the gamma model, despite its lower complexity, shows substantially higher AIC (879.59) and BIC (888.87) values compared to the mixture model, further reinforcing its inadequacy as a representation of the observed TBF behavior.

Figure 10 illustrates the limitations of single-distribution modeling for this failure dataset. The PDF overlay reveals that the gamma model fails to represent the sharp peak near zero and completely misses the multimodal nature evident in the histogram.

The CDF comparison demonstrates significant deviations between the theoretical gamma distribution (orange crosses) and the empirical distribution (blue circles), particularly in the lower tail region (0-100 hours) where the model underestimates failure probabilities, and in the upper tail where it overestimates them.

The P-P plot shows departures from linearity, with an S-shaped curve. The Q-Q plot reveals more nonlinearities, with the data points forming a curved pattern rather than following the expected 45-degree line.

Table 14 – Mixture vs. Single Distributions: G1Rac_MC0-018_iexplore_c0000005.

GOF Metric	**Mixture Model (12-comp)**	**Gamma Simple Distribution**
KS / p-value	0.0290 / 0.999999983	0.3327 / 1.37E-16
AD / p-value	0.279 / 0.4875	16.6652 / 0.000999
Log-Likelihood	-261.50	-436.80
AIC	611.00	879.59
BIC	705.29	888.87

5.0.6.6 Computational Cost Analysis

Given that mixture models can be computationally intensive, particularly when evaluating multiple distribution types and component numbers, a computational cost analysis was conducted. This analysis examines the trade-offs between model complexity and computational requirements.

Figure 11 and 12 present the computational cost analysis, showing execution time, memory usage, CPU utilization, and operation counts across different distribution types and component numbers.

Figure 11 presents the computational cost analysis for homogeneous mixture models across varying numbers of components. The analysis reveals distinct computational characteristics for different distribution families.

Figure 12 extends the analysis to heterogeneous mixture models, where components can belong to different distribution families. This analysis is particularly relevant for the selected 12-component model, which combines lognormal, gamma, and normal distributions.

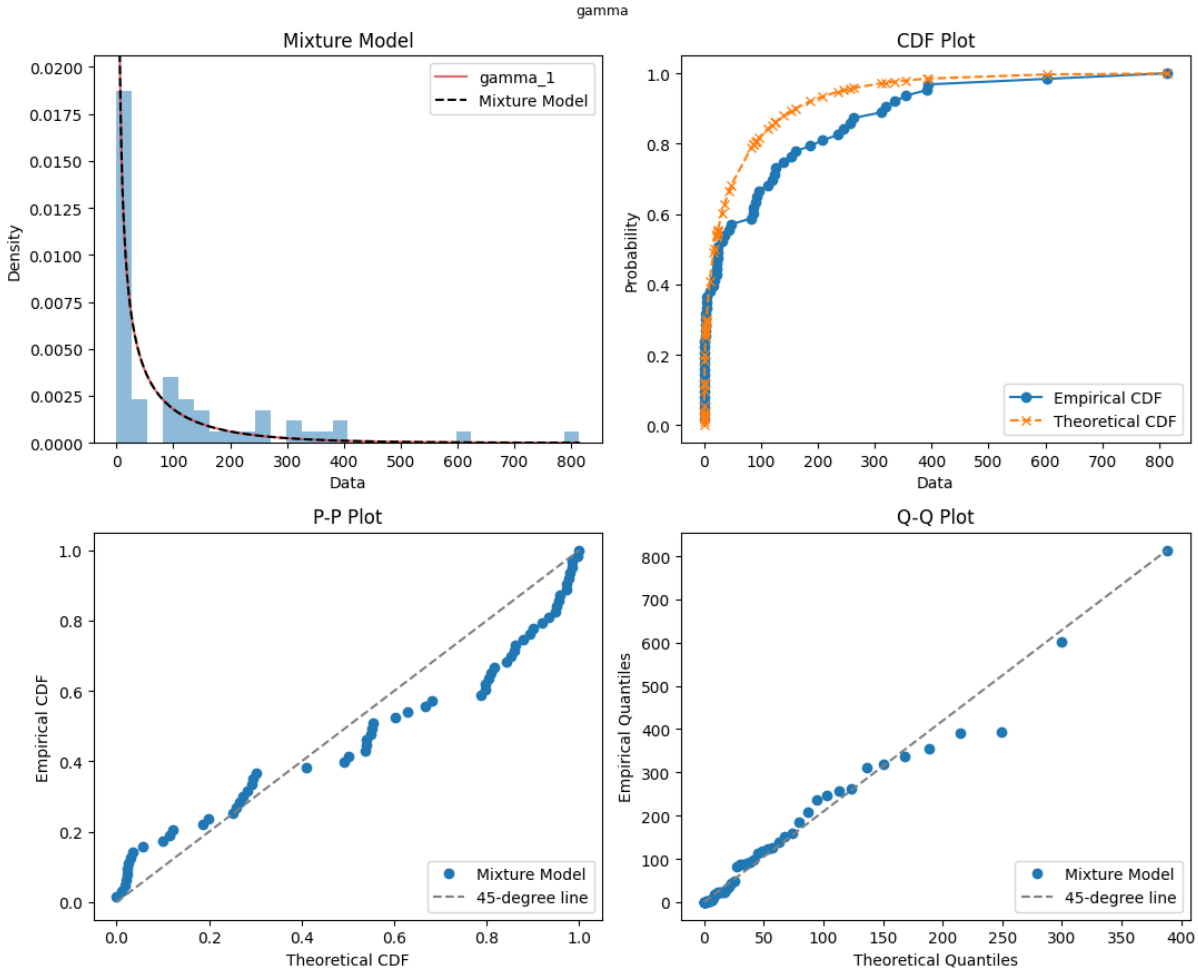

Figure 10 – G1Rac_MC0-018_iexplore_c0000005 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

The computational analysis reveals the scaling behavior of mixture model estimation as a function of both distribution complexity and the number of mixture components. These results inform practical considerations for large-scale reliability analysis applications. While the computational cost is substantial, the significant improvement in model adequacy and the critical importance of accurate failure modeling in reliability applications justify the computational investment.

5.1 Results Summary

To avoid overloading this section with dataset analyses, only one representative dataset is presented in the main text. The remaining analyses, which follow the same methodological structure, are provided in the appendices A to I for reference.

This research employed an extensive dataset to demonstrate the heterogeneity inherent in computer failure patterns, even when examining failures from the same process (`iexplore.exe`) while varying only the underlying failure cause. The failure data were extracted from computers where the failures were also analyzed, revealing no apparent systematic failure pattern that

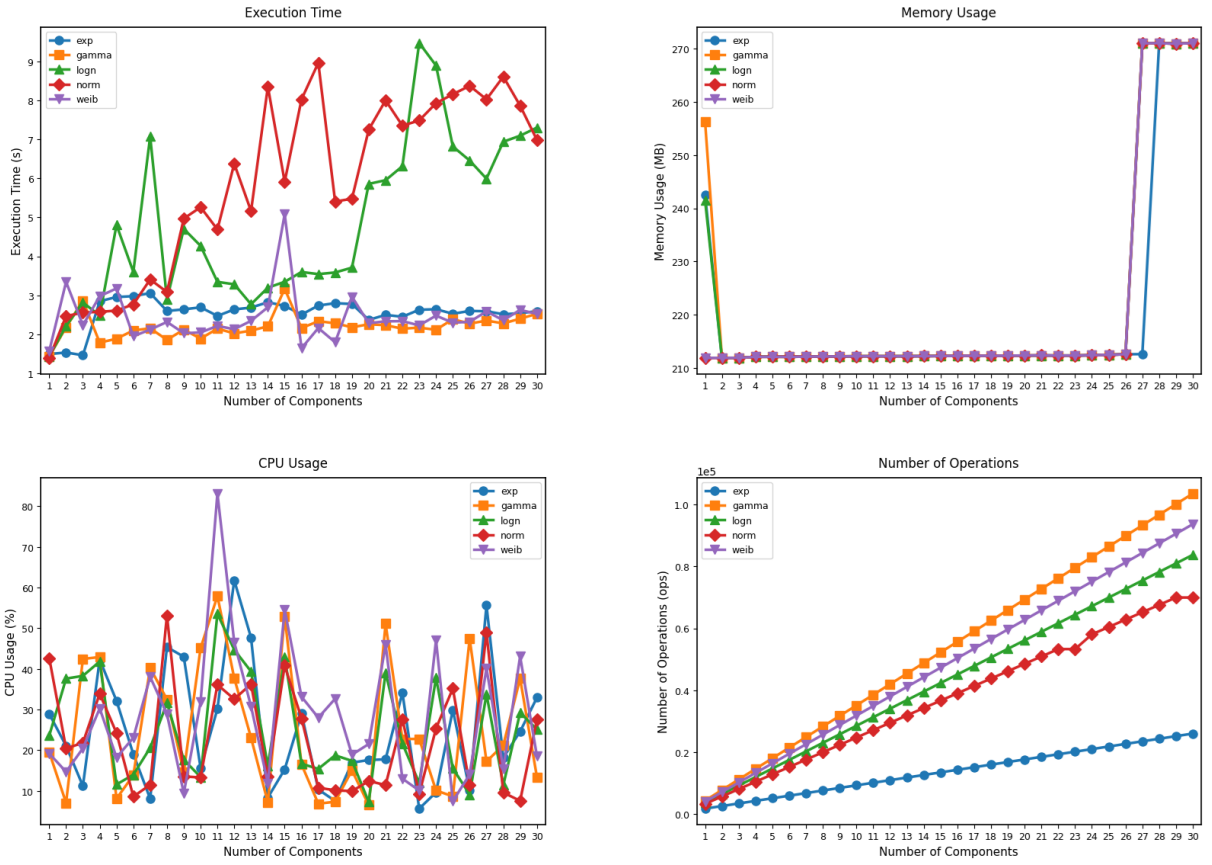

Figure 11 – G1Rac_MC0-018_iexplore_c0000005 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

could reliably distinguish the origin computer group of a given dataset. While certain distributions appear more frequently in some groups than others, these differences are insufficient to establish definitive classification patterns.

The study categorized computer environments into four distinct groups based on their operational context:

- ❏ **G1:** Undergraduate Laboratory Computers
- ❏ **G2:** University Administrative Department Computers
- ❏ **G3:** Corporate Environment Computers
- ❏ **G4:** Personal and HomeOffice Computers (Corporate, Academic, and Personal)

And four primary failure causes were identified for the `iexplore.exe` process:

- ❏ **c0000005:** Access violation (Memory addressing errors)
- ❏ **c0000374:** Heap corruption (Memory management failures)

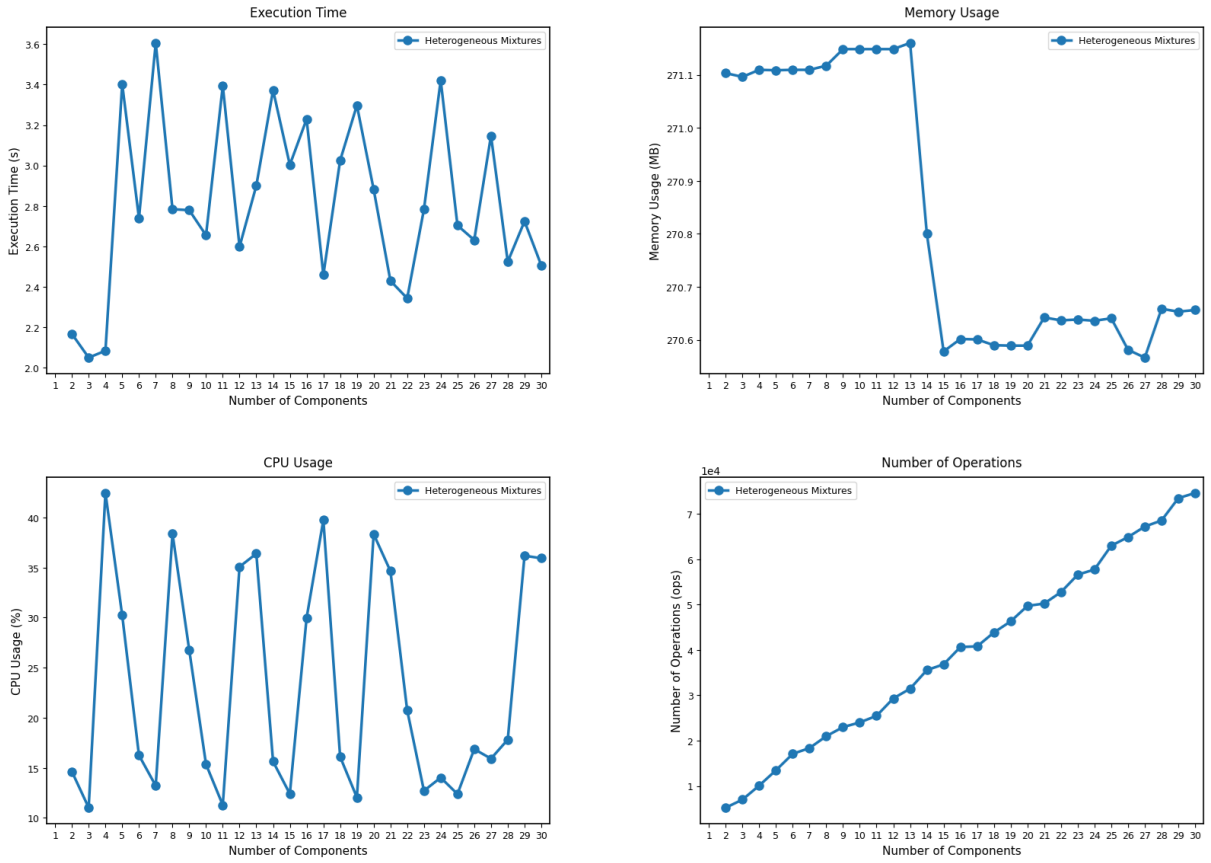

Figure 12 – G1Rac_MC0-018_iexplore_c0000005 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

- ❑ **c0000096:** Privileged instruction (Security violations)
- ❑ **e06d7363:** Unhandled C++ exception (Exception handling errors)

Table 15 presents the mixture model configurations and corresponding goodness-of-fit tests for each selected dataset. The analysis reveals that datasets following Approach 1 (general system failures) exhibit significantly more mixture components compared to `iexplore.exe`-specific failure datasets.

Table 15 – Best Mixture Models and Goodness-of-Fit Tests by Dataset.

Dataset	**Best Mixture Model Components**	**GoF Test**
G1Rac_MC0-018_iexplore_c0000005	logn-logn-logn-logn-logn-gamma-logn-norm-logn-norm-logn-norm	KS
G2Rac_MC-157400	logn-logn-logn-norm-norm-weib-logn-norm-weib-gamma-norm-norm-norm-norm-norm-norm-gamma-norm-norm-norm-norm	KS
G2Rac_MC-157400_iexplore_c0000374	logn-logn-lagn-logn-weib	KS
G2Rac_MC1-074	logn-logn-norm-weib-weib-gamma-gamma-logn-norm-norm	AD
G2Rac_MC1-074_iexplore_c0000005	gamma-gamma-gamma-norm	KS
G2Rac_MC1-074_iexplore_c0000096	logn-logn-logn-logn-norm-logn-logn-gamma	KS
G3Rac_DSK023	logn-logn-logn-logn-weib-gamma-gamma-logn-logn-weib-weib-weib-gamma	KS
G3Rac_DSK023-iexplore_c0000005	logn-logn-logn-weib-weib-norm-logn-logn-gamma-logn-norm-logn	KS
G4...bba75907480	weib-weib-weib-weib-weib-logn-logn-logn-norm-logn-weib-norm-gamma-logn-logn-norm	KS
G4...bba75907480-iexplore_e06d7363	logn-logn-logn-norm-norm-norm-norm	AD

5.2 Results and Pattern Analysis

While the previous results focused on 10 representative datasets to illustrate key findings, this section extends the analysis to encompass the complete dataset of 77 samples (41 from Approach 1 and 36 from Approach 2). The pattern analysis aims to identify relationships between failure characteristics, distributional properties, and clustering behaviors across the entire experimental dataset.

5.2.1 Comparative Group Analysis

The comprehensive analysis identified four distinct groups with different characteristics regarding sample size, statistical properties, and failure patterns. Table 16 summarizes the key statistical properties of each group, revealing significant inter-group variability.

Group 1 represents the most homogeneous cluster with a single sample, characterized by relatively low mean values (41.52) and moderate variability (70.91). This group exhibits concentrated Time Between Failures (TBF) values within the lower range, suggesting a system with consistent but frequent failure patterns.

Group 2 constitutes the largest and most heterogeneous with 53 samples, demonstrating exceptional diversity in statistical properties. The group exhibits substantial inter-sample variation. This variability indicates that G2 encompasses systems with fundamentally different operational characteristics, usage patterns, or underlying failure mechanisms.

Group 3 contains 5 samples but displays extreme statistical contrast between them. This difference suggests potentially different system configurations, software loads, or usage patterns within the same corporate environment classification.

Group 4 demonstrates sample diversity. Median values remain consistently low (5.29–59.76), indicating the presence of occasional very long inter-failure intervals alongside predominantly short intervals. This pattern suggests mixed-usage environments where systems experience both intensive and idle periods.

Table 16 – Statistical Characteristics by Computer Group.

Characteristic	G1	G2	G3	G4
Sample Size	1	53	5	18
Mean Range	41.52	33.23–193.07	31.38–184.12	52.22–208.46
Median Range	2.73	0.55–109.43	3.90–73.49	5.29–59.76
Std Dev Range	70.91	47.91–359.76	82.52–312.88	93.52–449.07
Maximum Value	813.90	2924.98	455.44	1581.13

5.2.2 Clustering Methodology Comparison

Table 17 presents the performance comparison of different clustering approaches across the complete dataset. The clustering analysis reveals revealed preferences across different approaches. K-Means and Fuzzy C-Means consistently identify 11–20 clusters, suggesting well-defined geometric structures in the high-dimensional failure characteristic space.

HDBSCAN consistently identifies fewer clusters (2–9), reflecting its density-based approach that merges closely related groups based on connectivity patterns rather than geometric proximity. Gaussian Mixture Models exhibit interesting behavioral differences based on the information criterion employed. GMM with BIC penalty produces results similar to K-Means (11–19 clusters), reflecting BIC's stronger penalty for model complexity. Conversely, GMM with AIC tends toward more complex models with higher cluster numbers, consistent with AIC's preference for model fit over parsimony.

5.2.3 Distributional Analysis and Goodness-of-Fit Patterns

The distributional analysis revealed preferences among different goodness-of-fit criteria, providing insights into the underlying nature of failure processes. Table 18 summarizes distribution preferences across all 77 samples.

The analysis reveals a fundamental dichotomy in distributional preferences between information criteria and goodness-of-fit tests. AIC and BIC criteria demonstrate strong agreement, favoring normal distributions in approximately 70% of cases with log-normal distributions accounting for 25%. This preference reflects these criteria's emphasis on likelihood maximization balanced against model complexity.

Table 17 – Clustering Method Performance Comparison.

Clustering Method	**Typical Cluster Range**	**Methodological Characteristics**
K-Means	11–20	Centroid-based
HDBSCAN	2–9	Density-based
Fuzzy C-Means	11–20	Probabilistic membership
GMM (BIC)	11–19	Model complexity penalty
GMM (AIC)	18+	Model complexity preference

Table 18 – Distribution Preferences by Goodness-of-Fit Criteria.

Distribution	**AIC (%)**	**BIC (%)**	**KS (%)**	**AD (%)**
Normal	70	70	20	25
Log-normal	25	25	50	50
Weibull	3	3	20	18
Gamma	1	1	8	5
Exponential	1	1	2	2

In stark contrast, KS and AD tests prefer log-normal distributions in approximately 50% of cases, with normal distributions accounting for only 20–25%. This discrepancy highlights the different distributional aspects emphasized by each criterion: information criteria prioritize likelihood-complexity trade-offs, while goodness-of-fit tests focus on distributional shape agreement and tail behavior.

The average number of mixture components varies systematically across criteria, with AIC suggesting 17 components, BIC recommending 14, and KS and AD indicating 15–16 components respectively. This pattern reflects the inherent complexity preferences of each evaluation approach.

5.2.4 Failure Type Characterization

The mixture model analysis reveals systematic patterns in component composition across different failure types and computer groups. Table 19 summarizes the typical component ranges and distributional preferences for different `iexplore.exe` failure codes.

Access violations (c0000005) consistently require 15–20 mixture components with log-normal distributions dominating the mixture composition. This pattern suggests complex, multiplicative failure processes where multiple independent factors contribute to failure occurrence. The high component count indicates significant heterogeneity in access violation mechanisms across different usage contexts.

Heap corruption failures (c0000374) show increased prevalence of Weibull distributions and require fewer components (8–17), suggesting more structured failure patterns with characteristic aging or wear-out behaviors.

Privileged instruction violations (c0000096) prefer normal distributions and exhibit the widest component range (7–24), indicating high variability in underlying failure mechanisms.

C++ exceptions (e06d7363) present in just one dataset with 14 components and log-normal distribution preference.

5.2.5 Proposed Statistical Classification Algorithm

No consistent failure pattern could be identified across all computer groups. Consequently, the analysis was restricted to failures associated with the `iexplore.exe` process. Within

Table 19 – Mixture Model Components by Failure Type.

Failure Code	Component Range	Dominant Distribution
c0000005 (Access Violation)	15–20	Log-normal
c0000374 (Heap Corruption)	8–17	Weibull
c0000096 (Privileged Instruction)	7–24	Normal
e06d7363 (C++ Exception)	14	Log-normal

this scope, a statistical classification algorithm was developed to distinguish between failure causes based on the statistical characteristics of the observed failures. The proposed approach achieved an overall classification accuracy of 65.7%, corresponding to 23 correctly classified samples out of 35.

- ❏ **Heap Corruption Detection:** If (Median < 3.0 AND Standard Deviation/Mean Ratio > 3.0 AND Kurtosis > 20), classify as c0000374.
- ❏ **Privileged Instruction Detection:** Else If (Mean < 25 AND Maximum < 120 AND Skewness < 3.0), classify as c0000096.
- ❏ **Access Violation Detection:** Else If (Skewness > 2.0 AND Best Distribution = Log-normal), classify as c0000005.
- ❏ **Default Classification:** Else (Classify as other failure codes).

The classification algorithm demonstrates that statistical characteristics of inter-failure intervals contain systematic information about underlying failure mechanisms. While 65.7% accuracy indicates potential for improvement, this performance represents a advancement over random classification (25% expected accuracy for four classes).

This analysis demonstrates that computer failure patterns exhibit remarkable complexity and heterogeneity, even when examining failures from identical processes running on same operating systems. The research reveals that computer usage patterns fundamentally influence failure characteristics, generating apparent randomness that masks underlying systematic behaviors.

The key finding of this research is that usage context dominates technical homogeneity in determining failure patterns. Despite employing identical operating systems and examining the same program (`iexplore.exe`), the four computer groups (laboratory, administrative, corporate, and mixed-use) exhibit fundamentally different statistical signatures. This suggests that usage patterns, software interactions, and operational contexts create unique profiles that manifest as distinct failure behaviors.

The proposed classification algorithm's 65.7% accuracy, while modest, demonstrates that statistical fingerprints of usage patterns are embedded within failure data. This finding has significant implications for system reliability engineering, suggesting that failure prediction models must incorporate usage context rather than relying solely on technical specifications.

Furthermore, the distributional analysis reveals that different failure mechanisms exhibit characteristic statistical signatures: access violations follow log-normal patterns suggesting multiplicative risk factors, heap corruptions exhibit Weibull behaviors indicating aging processes, and security violations show normal distributions reflecting random triggering events. These signatures persist across usage contexts while varying in intensity and complexity.

The analysis establishes that usage patterns create variations that appear random when viewed through purely technical lenses. Understanding and modeling these usage-induced

variations is essential for developing accurate reliability predictions and effective maintenance strategies in diverse computing environments. The apparent randomness observed in same-system failures actually reflects the hidden order of computer interaction patterns, highlighting the importance of considering usage context in software reliability engineering.

Chapter 6

Conclusion

This chapter presents the main conclusions drawn from the research, highlighting its contributions to the software reliability literature. It provides a summary of the key findings, discusses the study's limitations, and outlines potential directions for future research.

6.1 Key Research Findings

The results of this study provide evidence that finite mixture models offer a more robust and flexible representation of time between failures in software systems compared to traditional single-distribution models. Evaluation based on multiple statistical measures, including goodness-of-fit tests (Kolmogorov–Smirnov and Anderson–Darling), information criteria (Akaike Information Criterion and Bayesian Information Criterion), and graphical validations (PDF, CDF, Q–Q, and P–P plots), corroborated the superiority of mixture models in capturing the complex statistical behavior of TBFs.

These models effectively addressed the inherent multimodal nature of the data, which could not be adequately represented by simple distributions. Therefore, the findings confirm that the main objective of this research was successfully achieved, demonstrating that mixture models can indeed provide a more robust and flexible framework for representing software failure behavior in systems affected by multiple failure causes and operational variability.

Although the results demonstrate that mixture models provide a flexible and effective framework for representing heterogeneous failure behavior, they do not constitute a universal solution for all datasets. In several cases, simpler single-distribution models achieved comparable goodness-of-fit, indicating that the additional complexity of mixture models is not always necessary. Therefore, the contribution of this work is not to advocate mixture models as universally superior, but rather to clarify the conditions under which their use is justified and beneficial.

The findings revealed that software failure times exhibit significant heterogeneity, challenging the traditional view of a single underlying failure mechanism. Instead, the data supported the hypothesis that multiple failure causes coexist within software systems. Further-

more, the analysis indicated that models with 10 to 15 mixture components often offered better statistical accuracy. Within this range, no substantial improvements were observed in the applied GoF tests, suggesting that additional components did not yield meaningful gains in performance.

To assess the practicality of applying mixture models in real-world settings, a computational cost analysis was conducted. The results showed that, even as model complexity increased, the computational resource usage remained within acceptable limits, allowing all models to be successfully fitted. In the worst-case scenarios, the computation time did not exceed 30 seconds, memory consumption remained below 400 MB, CPU usage peaked at approximately 85%, and the total number of operations did not surpass the order of 10^5. Thus, the use of mixture models is not computationally prohibitive, further supporting their applicability in operational environments.

Beyond its specific application to software failure data, an additional contribution of this dissertation lies in the generality of the proposed methodology. The modeling, clustering, and validation framework adopted in this work is not limited to failure time data and can be applied to other datasets involving continuous variables. As such, the approach may be useful for analyzing heterogeneous behaviors in broader empirical contexts, extending its relevance beyond traditional software reliability studies.

6.1.1 Operational Environment Impact

One of the findings of the study was the clear influence of the operational environment on TBF patterns. The analysis across four different operational groups revealed distinct and characteristic behaviors that reflect the underlying usage patterns and system conditions:

G1 (University Laboratories): These systems operate in academic environments, their usage tends to be less structured, as laboratory computers are often used by multiple users for a wide variety of tasks. This can introduce variability in workload patterns. However, in this study, the failure patterns observed for G1 appeared relatively homogeneous and were represented by simpler mixture models with fewer components, which may be explained by the smaller number of samples available for this group compared to the others, potentially masking the possible diversity of behaviors.

G2 (University Administrative Departments): In contrast, systems from administrative departments operate in more stable and narrowly defined contexts, with well-established routines and a limited range of applications. Despite this more controlled operational environment, this group exhibited the greatest diversity in failure configurations, with mixture models reaching up to 30 components. This group contributed the largest number of samples to the dataset, and the abundance of data revealed a wide variety of failure configurations. Consequently, the mixture models for this group showed the highest complexity, with up to 30 components, indicating that the larger dataset captured a broader spectrum of failure behaviors within this environment.

G3 (Corporate Environment): Presented behavior similar to group G4, characterized by the presence of lognormal, Weibull, normal, and gamma distributions in the mixture components. The corporate environment showed intermediate complexity, yet with sufficient diversity to require multi-component mixture models.

G4 (Personal and HomeOffice Computers): Exhibited the greatest diversity of distributions in the mixture models, notably presenting the exponential distribution, which was not commonly observed in other groups. This unique characteristic suggests that mixed-usage environments create conditions that lead to diverse failure behavior.

Therefore, even under a common operating system architecture, the usage profile and workload environment significantly impact the failure behavior of software systems.

6.1.2 Failure Pattern Characterization

The study uncovered patterns in the TBFs, particularly for failures associated with specific software components such as Internet Explorer. Failures including access violation, heap corruption, privileged instruction violation and unhandled C++ exception showed consistent preferences for specific distribution types, such as lognormal, Weibull and normal distributions, depending on the failure category and underlying failure mechanism. These findings suggest the possibility of identifying and characterizing failure root causes based solely on the statistical properties of the TBFs, contributing to more precise failure classification and potentially predictive maintenance strategies.

The finding of these patterns across different datasets and operational environments indicates that certain failure types have inherent statistical signatures that can be leveraged for diagnostic purposes.

6.1.3 Multi-Modal Failure Behavior

The presence of multiple failure causes was demonstrated through clustering analyses conducted using various unsupervised learning techniques, including K-Means, HDBSCAN, Fuzzy C-Means, and Gaussian Mixture Models with both AIC and BIC selection criteria. In many cases, the optimal number of clusters corresponded closely to the number of components in the best-fitting mixture models, providing support for the interpretation of these components as distinct failure behaviors rather than statistical artifacts.

This convergence of results from different analytical approaches strengthens the conclusion that software systems exhibit multi-modal failure behavior. The clustering analyses revealed that failures naturally group into distinct categories based on their temporal characteristics.

6.1.4 Distributional Preferences and Model Complexity

The research led to the identification of preferred probability distributions for modeling software failure processes. The lognormal distribution emerged as the most prevalent, appearing in approximately 70% of the best-fitting mixture models, followed by the Normal and Weibull distributions. The dominance of the lognormal distribution suggests that many software failure processes follow multiplicative rather than additive patterns.

It is noteworthy that the Lognormal and Weibull distributions are not frequently used in software reliability studies. Among the 16 related works reviewed, only about 20% employed either of these distributions. Specifically, (VINEYARD; AMOAKO-GYAMPAH; MEREDITH, 1999) applied Weibull and Lognormal models to represent failure and repair times, (SCHROEDER; GIBSON, 2009) used Weibull and Gamma distributions to describe Time Between Failures (TBF) and Time To Repair (TTR), and (KUMAR; JAIN, 2023) combined Lognormal and Weibull distributions within homogeneous and heterogeneous mixture frameworks. In line with these studies, the present research confirms the effectiveness of these distributions, particularly within mixture model structures, for representing complex and heterogeneous software failure patterns.

Sensitivity analyses indicated that models with 10 to 15 components offered good statistical accuracy. Within this range, no substantial improvements were observed in the applied Goodness-of-Fit (GoF) tests, suggesting that increasing the number of components beyond this point yields diminishing returns in terms of model performance. To complement this evaluation, a computational cost analysis was conducted to determine whether the use of mixture models is computationally intensive. The results showed that, despite increasing model complexity, the computational demands remained within acceptable limits. Thus, the application of mixture models is not prohibitively costly from a computational standpoint, supporting their feasibility for real-world implementation.

6.2 Research Limitations

This work acknowledges several limitations that should be considered. First, the scope of the dataset, while comprehensive within its domain, is based on Windows 7 Reliability Analysis Component (RAC) data and is primarily focused on specific software applications and operating environments. This raises concerns regarding dataset dependence and the generalizability of the conclusions. Although the operating system and data collection mechanism are platform-specific, the observed failure patterns are driven by fundamental interactions between software components, workloads and operational profiles. Consequently, while numerical results and distributional configurations may differ across platforms or more recent systems, the proposed modeling methodology and analytical framework are expected to remain applicable. Broader datasets encompassing more diverse platforms, application types,

and operational contexts would allow for more generalizable conclusions and potentially reveal additional failure patterns not observed in the current study.

Additionally, the modeling approach assumed statistical independence between consecutive failures and considered only univariate TBF data. While this assumption is common in reliability modeling, real software systems may exhibit temporal dependencies between failures that could provide additional modeling opportunities. The independence assumption may be particularly limiting in cases where failures are caused by cumulative effects.

Another acknowledged limitation relates to the interpretability of mixture components. While mixture models provide statistical flexibility and fitting capability, they become progressively less interpretable as the number of components increases, especially when different distribution types are combined within a single model.

6.3 Future Research Directions

While the present study demonstrates that mixture distributions are effective in characterizing heterogeneous failure patterns, an important next step is to understand the underlying mechanisms that give rise to these distinct failure classes. Advancing in this direction could transform the proposed approach from primarily descriptive to more prescriptive, enabling deeper insights into failure causation and system behavior. Building upon the findings of this research, several possibilities for future work emerge:

Dataset Generalization: Applying the proposed methodology to other datasets from different domains, platforms and application types would test the generalizability of the conclusions and potentially reveal new failure patterns.

Multivariate Extensions: Incorporating additional variables such as workload intensity, system age, user behavior patterns into the analysis may enhance the mixture models predictive capability.

Temporal Modeling: Expanding the framework to account for temporal dependencies in failure data could improve the modeling of complex systems where failures are not independent.

Real-time Applications: Integrating mixture models into real-time monitoring systems could offer valuable tools for failure prediction in operational environments.

Machine Learning Integration: Combining mixture modeling approaches with machine learning techniques could lead to hybrid models that leverage the interpretability of statistical models and the flexibility of machine learning algorithms.

Bibliography

ANDERSON, T. W. Anderson-darling tests of goodness-of-fit. **International encyclopedia of statistical science**, v. 1, p. 52–54, 2011. Disponível em: <https://doi.org/10.1007/978-3-642-04898-2_118>.

AVIZIENIS, A.; LAPRIE, J.-C.; RANDELL, B.; LANDWEHR, C. Basic concepts and taxonomy of dependable and secure computing. **IEEE transactions on dependable and secure computing**, IEEE, v. 1, n. 1, p. 11–33, 2004. Disponível em: <https://doi.org/10.1109/TDSC.2004.2>.

BAUDRY, J.-P.; RAFTERY, A. E.; CELEUX, G.; LO, K.; GOTTARDO, R. Combining mixture components for clustering. **Journal of computational and graphical statistics**, Taylor & Francis, v. 19, n. 2, p. 332–353, 2010. Disponível em: <https://doi.org/10.1198/jcgs.2010.08111>.

BERGER, V. W.; ZHOU, Y. Kolmogorov–smirnov test: Overview. **Wiley statsref: Statistics reference online**, Wiley Online Library, 2014. Disponível em: <https://doi.org/10.1002/9781118445112.stat06558>.

BIERNACKI, C.; CELEUX, G.; GOVAERT, G. Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models. **Computational Statistics & Data Analysis**, Elsevier, v. 41, n. 3-4, p. 561–575, 2003. Disponível em: <https://doi.org/10.1016/S0167-9473(02)00163-9>.

BILMES, J. A. et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. **International computer science institute**, Berkeley, CA, v. 4, n. 510, p. 126, 1998.

BIRD, C.; RANGANATH, V.-P.; ZIMMERMANN, T.; NAGAPPAN, N.; ZELLER, A. Extrinsic influence factors in software reliability: A study of 200,000 windows machines. In: **Companion Proceedings of the 36th International Conference on Software Engineering**. [s.n.], 2014. p. 205–214. Disponível em: <https://doi.org/10.1145/2591062.2591173>.

BLOSTEIN, M.; MILJKOVIC, T. On modeling left-truncated loss data using mixtures of distributions. **Insurance: Mathematics and Economics**, Elsevier, v. 85, p. 35–46, 2019. Disponível em: <https://doi.org/10.1016/j.insmatheco.2018.12.001>.

BUČAR, T.; NAGODE, M.; FAJDIGA, M. Reliability approximation using finite weibull mixture distributions. **Reliability Engineering & System Safety**, Elsevier, v. 84, n. 3, p. 241–251, 2004. Disponível em: <https://doi.org/10.1016/j.ress.2003.11.008>.

CAI, W.; ZHAO, J.; ZHU, M. A real time methodology of cluster-system theory-based reliability estimation using k-means clustering. **Reliability Engineering & System Safety**, Elsevier, v. 202, p. 107045, 2020. Disponível em: <https://doi.org/10.1016/j.ress.2020.107045>.

CELEBI, M. E.; KINGRAVI, H. A.; VELA, P. A. A comparative study of efficient initialization methods for the k-means clustering algorithm. **Expert systems with applications**, Elsevier, v. 40, n. 1, p. 200–210, 2013. Disponível em: <https://doi.org/10.1016/j.eswa.2012.07.021>.

DIGIUSEPPE, N.; JONES, J. A. Software behavior and failure clustering: An empirical study of fault causality. In: IEEE. **2012 IEEE Fifth International Conference on Software Testing, Verification and Validation**. 2012. p. 191–200. Disponível em: <https://doi.org/10.1109/ICST.2012.99>.

ELMAHDY, E. E. A new approach for weibull modeling for reliability life data analysis. **Applied Mathematics and computation**, Elsevier, v. 250, p. 708–720, 2015. Disponível em: <https://doi.org/10.1016/j.amc.2014.10.036>.

_____. Modelling reliability data with finite weibull or lognormal mixture distributions. **Appl. Math. Inf. Sci**, v. 11, n. 9, p. 1081–1089, 2017. Disponível em: <https://doi.org/10.18576/amis/110414>.

ELMAHDY, E. E.; ABOUTAHOUN, A. W. A new approach for parameter estimation of finite weibull mixture distributions for reliability modeling. **Applied Mathematical Modelling**, Elsevier, v. 37, n. 4, p. 1800–1810, 2013. Disponível em: <https://doi.org/10.1016/j.apm.2012.04.023>.

GERNAND, L.; FENSKE, N. Understanding aic and bic in model selection. **Handreichungen zum Vortrag vom**, v. 20, p. 1–18, 2009.

GNANADESIKAN, R.; WILK, M. B. Probability plotting methods for the analysis of data. **Biometrika**, v. 55, n. 1, p. 1–17, 1968. Disponível em: <https://doi.org/10.1093/biomet/55.1.1>.

GOEL, A. L. Software reliability models: Assumptions, limitations, and applicability. **IEEE Transactions on software engineering**, IEEE, n. 12, p. 1411–1423, 1985. Disponível em: <https://doi.org/10.1109/TSE.1985.232177>.

GOKHALE, S. S.; TRIVEDI, K. S. A time/structure based software reliability model. **Annals of Software Engineering**, Springer, v. 8, n. 1, p. 85–121, 1999. Disponível em: <https://doi.org/10.1023/A:1018923329647>.

GOSEVA-POPSTOJANOVA, K.; TRIVEDI, K. S. Failure correlation in software reliability models. **IEEE Transactions on Reliability**, IEEE, v. 49, n. 1, p. 37–48, 2000. Disponível em: <https://doi.org/10.1109/24.855535>.

GUPTA, M. R.; CHEN, Y. et al. Theory and use of the em algorithm. **Foundations and Trends® in Signal Processing**, Now Publishers, Inc., v. 4, n. 3, p. 223–296, 2011. Disponível em: <https://doi.org/10.1561/2000000034>.

HALKIDI, M.; BATISTAKIS, Y.; VAZIRGIANNIS, M. On clustering validation techniques. **Journal of intelligent information systems**, Springer, v. 17, p. 107–145, 2001. Disponível em: <https://doi.org/10.1023/A:1012801612483>.

HAMILL, M.; GOSEVA-POPSTOJANOVA, K. Common trends in software fault and failure data. **IEEE Transactions on Software Engineering**, IEEE, v. 35, n. 4, p. 484–496, 2009. Disponível em: <https://doi.org/10.1109/TSE.2009.3>.

HARTIGAN, J. A. Statistical theory in clustering. **Journal of classification**, Springer, v. 2, n. 1, p. 63–76, 1985. Disponível em: <https://doi.org/10.1007/BF01908064>.

HSU, C.-J.; HUANG, C.-Y. Optimal weighted combinational models for software reliability estimation and analysis. **IEEE Transactions on Reliability**, IEEE, v. 63, n. 3, p. 731–749, 2014. Disponível em: <https://doi.org/10.1109/TR.2014.2315966>.

IEEE. Ieee standard glossary of software engineering terminology. **IEEE Std**, v. 610, p. 12, 1990.

JAIN, A. K. Data clustering: 50 years beyond k-means. **Pattern recognition letters**, Elsevier, v. 31, n. 8, p. 651–666, 2010. Disponível em: <https://doi.org/10.1016/j.patrec.2009.09.011>.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. **ACM computing surveys (CSUR)**, Acm New York, NY, USA, v. 31, n. 3, p. 264–323, 1999. Disponível em: <https://doi.org/10.1145/331499.331504>.

KEENE, S.; LANE, C. Combined hardware and software aspects of reliability. **Quality and reliability engineering international**, Wiley Online Library, v. 8, n. 5, p. 419–426, 1992. Disponível em: <https://doi.org/10.1002/qre.4680080506>.

KUMAR, S.; JAIN, M. Heterogeneous mixture model for software reliability prediction. In: SPRINGER. **International conference on soft computing for problem-solving**. 2023. p. 867–879. Disponível em: <https://doi.org/10.1007/978-981-97-3292-0_59>.

KUMAR, S.; JAIN, M.; GANGOPADHAY, A. A mixture model of the mixed distributions to analyze some reliability indices and survival data. In: **Proceedings-26th international conference on reliability and quality in design (RQD), international society of science and applied technologies (ISSAT), Florida USA**. [S.l.: s.n.], 2021. p. 245–250.

LI, P. L.; NI, M.; XUE, S.; MULLALLY, J. P.; GARZIA, M.; KHAMBATTI, M. Reliability assessment of mass-market software: insights from windows vista®. In: IEEE. **2008 19th International Symposium on Software Reliability Engineering (ISSRE)**. 2008. p. 265–270. Disponível em: <https://doi.org/10.1109/ISSRE.2008.60>.

LYU, M. R. Software reliability engineering: A roadmap. In: IEEE. **Future of Software Engineering (FOSE'07)**. 2007. p. 153–170. Disponível em: <https://doi.org/10.1109/FOSE.2007.24>.

LYU, M. R.; NIKORA, A. P. A heuristic approach for software reliability prediction: the equally-weighted linear combination model. In: **ISSRE**. [s.n.], 1991. p. 172–181. Disponível em: <https://doi.org/10.1109/ISSRE.1991.145376>.

MATIAS, R.; OLIVEIRA, G. D.; ARAUJO, L. B. de. Operating system reliability from the quality of experience viewpoint: an exploratory study. In: **Proceedings of the 28th Annual ACM Symposium on Applied Computing**. [s.n.], 2013. p. 1644–1649. Disponível em: <https://doi.org/10.1145/2480362.2480669>.

MATIAS, R.; PRINCE, M.; BORGES, L.; SOUSA, C.; HENRIQUE, L. An empirical exploratory study on operating system reliability. In: **Proceedings of the 29th Annual ACM Symposium on Applied Computing**. [s.n.], 2014. p. 1523–1528. Disponível em: <https://doi.org/10.1145/2554850.2555021>.

MCLACHLAN, G. J.; KRISHNAN, T. **The EM algorithm and extensions**. John Wiley & Sons, 2008. Disponível em: <https://doi.org/10.1002/9780470191613>.

MCLACHLAN, G. J.; LEE, S. X.; RATHNAYAKE, S. I. Finite mixture models. **Annual review of statistics and its application**, Annual Reviews, v. 6, n. 1, p. 355–378, 2019. Disponível em: <https://doi.org/10.1146/annurev-statistics-031017-100325>.

MEILĂ, M.; HECKERMAN, D. An experimental comparison of model-based clustering methods. **Machine learning**, Springer, v. 42, p. 9–29, 2001. Disponível em: <https://doi.org/10.1023/A:1007648401407>.

MELNYKOV, V.; MELNYKOV, I. Initializing the em algorithm in gaussian mixture models with an unknown number of components. **Computational Statistics & Data Analysis**, Elsevier, v. 56, n. 6, p. 1381–1395, 2012. Disponível em: <https://doi.org/10.1016/j.csda.2011.11.002>.

MENARDI, G. Density-based silhouette diagnostics for clustering methods. **Statistics and Computing**, Springer, v. 21, p. 295–308, 2011. Disponível em: <https://doi.org/10.1007/s11222-010-9169-0>.

MICHAEL, S.; MELNYKOV, V. An effective strategy for initializing the em algorithm in finite mixture models. **Advances in Data Analysis and Classification**, Springer, v. 10, p. 563–583, 2016. Disponível em: <https://doi.org/10.1007/s11634-016-0264-8>.

MICROSOFT. **Compatibility and Reliability**. 2020. <https://learn.microsoft.com/en-us/windows/win32/win7devguide/compatibility-and-reliability>. Accessed: April 2025.

MOON, T. K. The expectation-maximization algorithm. **IEEE Signal processing magazine**, IEEE, v. 13, n. 6, p. 47–60, 1996. Disponível em: <https://doi.org/10.1109/79.543975>.

MULLEN, R. E. The lognormal distribution of software failure rates: application to software reliability growth modeling. In: IEEE. **Proceedings Ninth International Symposium on Software Reliability Engineering (Cat. No. 98TB100257)**. 1998. p. 134–142. Disponível em: <https://doi.org/10.1109/ISSRE.1998.730872>.

MUSA, J. D. Software reliability measurement. **Journal of Systems and Software**, Elsevier, v. 1, p. 223–241, 1979. Disponível em: <https://doi.org/10.1016/0164-1212(79)90023-2>.

_____. **Software reliability engineering: more reliable software faster and cheaper**. [S.l.]: AuthorHouse, 2004.

OHBA, M. Software reliability analysis models. **IBM Journal of research and Development**, IBM, v. 28, n. 4, p. 428–443, 1984. Disponível em: <https://doi.org/10.1147/rd.284.0428>.

OKAMURA, H.; DOHI, T. Application of em algorithm to nhpp-based software reliability assessment with generalized failure count data. **Mathematics**, MDPI, v. 9, n. 9, p. 985, 2021. Disponível em: <https://doi.org/10.3390/math9090985>.

OKAMURA, H.; MURAYAMA, A.; DOHI, T. Em algorithm for discrete software reliability models: a unified parameter estimation method. In: IEEE. **Eighth IEEE International Symposium on High Assurance Systems Engineering, 2004. Proceedings.** 2004. p. 219–228. Disponível em: <https://doi.org/10.1109/HASE.2004.1281746>.

OKAMURA, H.; WATANABE, Y.; DOHI, T. Estimating mixed software reliability models based on the em algorithm. In: IEEE. **Proceedings international symposium on empirical software engineering**. 2002. p. 69–78. Disponível em: <https://doi.org/10.1109/ISESE.2002.1166927>.

PAI, G. J. A survey of software reliability models. **arXiv preprint arXiv:1304.4539**, 2013.

PANIĆ, B.; KLEMENC, J.; NAGODE, M. Improved initialization of the em algorithm for mixture model parameter estimation. **Mathematics**, MDPI, v. 8, n. 3, p. 373, 2020. Disponível em: <https://doi.org/10.3390/math8030373>.

PHAM, H. Software reliability and cost models: Perspectives, comparison, and practice. **European Journal of operational research**, Elsevier, v. 149, n. 3, p. 475–489, 2003. Disponível em: <https://doi.org/10.1016/S0377-2217(02)00498-8>.

_____. **System software reliability**. [S.l.]: Springer Science & Business Media, 2007.

RAZALI, A. M.; SALIH, A. A. Combining two weibull distributions using a mixing parameter. **European Journal of Scientific Research**, v. 31, n. 2, p. 296–305, 2009.

RAZALI, N. M.; WAH, Y. B. et al. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. **Journal of statistical modeling and analytics**, v. 2, n. 1, p. 21–33, 2011.

RUHI, S.; SARKER, S.; KARIM, M. Mixture models for analyzing product reliability data: a case study. **SpringerPlus**, Springer, v. 4, p. 1–14, 2015. Disponível em: <https://doi.org/10.1186/s40064-015-1420-x>.

SAMMAKNEJAD, N.; ZHAO, Y.; HUANG, B. A review of the expectation maximization algorithm in data-driven process identification. **Journal of process control**, Elsevier, v. 73, p. 123–136, 2019. Disponível em: <https://doi.org/10.1016/j.jprocont.2018.12.010>.

SANTOS, C.; MATIAS, R.; TRIVEDI, K. A multisite characterization study on failure causes in system and applications software. In: IEEE. **2021 XI Brazilian Symposium on Computing Systems Engineering (SBESC)**. 2021. p. 1–8. Disponível em: <https://doi.org/10.1109/SBESC53686.2021.9628276>.

SANTOS, C. A. R. dos; MATIAS, R. Failure patterns in operating systems: An exploratory and observational study. **Journal of Systems and Software**, Elsevier, v. 137, p. 512–530, 2018. Disponível em: <https://doi.org/10.1016/j.jss.2017.03.058>.

SCHOLZ, F. W.; STEPHENS, M. A. K-sample anderson–darling tests. **Journal of the American Statistical Association**, Taylor & Francis, v. 82, n. 399, p. 918–924, 1987. Disponível em: <https://doi.org/10.1080/01621459.1987.10478517>.

SCHROEDER, B.; GIBSON, G. A. A large-scale study of failures in high-performance computing systems. **IEEE transactions on Dependable and Secure Computing**, Ieee, v. 7, n. 4, p. 337–350, 2009. Disponível em: <https://doi.org/10.1109/TDSC.2009.4>.

VINEYARD, M.; AMOAKO-GYAMPAH, K.; MEREDITH, J. R. Failure rate distributions for flexible manufacturing systems: An empirical study. **European journal of operational research**, Elsevier, v. 116, n. 1, p. 139–155, 1999. Disponível em: <https://doi.org/10.1016/S0377-2217(98)00096-4>.

WANG, H.; KHOSHGOFTAAR, T. M.; NAPOLITANO, A. A comparative study of ensemble feature selection techniques for software defect prediction. In: IEEE. **2010 Ninth International Conference on Machine Learning and Applications**. 2010. p. 135–140. Disponível em: <https://doi.org/10.1109/ICMLA.2010.27>.

WANG, H.; KHOSHGOFTAAR, T. M.; SELIYA, N. et al. How many software metrics should be selected for defect prediction? In: **FLAIRS**. [S.l.: s.n.], 2011.

WU, C. J. On the convergence properties of the em algorithm. **The Annals of statistics**, JSTOR, p. 95–103, 1983. Disponível em: <https://doi.org/10.1214/aos/1176346060>.

WU, Y.; LU, Z.; WU, J.; LIANG, X. Reliability evaluation of components with multiple failure modes based on mixture weibull distribution using expectation maximization algorithm. **Journal of Mechanical Science and Technology**, Springer, v. 38, n. 2, p. 649–660, 2024. Disponível em: <https://doi.org/10.1007/s12206-024-0113-1>.

XU, J.; KALBARCZYK, Z.; IYER, R. K. Networked windows nt system field failure data analysis. In: IEEE. **Proceedings 1999 Pacific Rim International Symposium on Dependable Computing**. [S.l.], 1999. p. 178–185.

ZHANG, B.; ZHANG, C.; YI, X. Competitive em algorithm for finite mixture models. **Pattern recognition**, Elsevier, v. 37, n. 1, p. 131–144, 2004. Disponível em: <https://doi.org/10.1016/S0031-3203(03)00140-7>.

ZHAO, K.; STEFFEY, D. Practical applications of mixture models to complex time-to-failure data. In: IEEE. **2013 Proceedings Annual Reliability and Maintainability Symposium (RAMS)**. 2013. p. 1–6. Disponível em: <https://doi.org/10.1109/RAMS.2013.6517714>.

ZHONG, S.; KHOSHGOFTAAR, T. M.; SELIYA, N. Analyzing software measurement data with clustering techniques. **IEEE Intelligent Systems**, IEEE, v. 19, n. 2, p. 20–27, 2004. Disponível em: <https://doi.org/10.1109/MIS.2004.1274907>.

Appendix

APPENDIX A

Results G2Rac-MC1-074 (Approach 1)

This case refers to failures observed throughout the operational history of the computer MC1-074, which operates within Group 2 (Graduate Laboratory environment).

A.0.0.1 Statistical Characterization

Table 20 presents the statistical characteristics of the TBF data for this sample. The descriptive statistics reveal a highly heterogeneous failure pattern characterized by extreme asymmetry and heavy-tailed behavior. The difference between mean (36.13 hours) and median (8.07 hours) indicates a strongly right-skewed distribution, which is further confirmed by the high positive skewness value of 5.11.

The extremely low mode value (0.0014 hours) combined with the minimum observation (0.0006 hours) indicates that some failures occur in rapid succession, possibly representing cascading failures or immediate re-failures after system recovery attempts.

The extremely high kurtosis value (41.50) indicates a distribution with heavy tails and a

Table 20 – Descriptive Statistics: G2Rac_MC1-074.

Statistic	**Value**
Count	246
Mean (hours)	36.13
Median (hours)	8.07
Mode (hours)	0.0014
Standard Deviation	69.45
Minimum	0.0006
Maximum	733.91
First Quartile (Q1)	0.77
Third Quartile (Q3)	37.32
Interquartile Range (IQR)	36.55
Skewness	5.11
Kurtosis	41.50
Main Data Range	0.01 – 160.75

sharp peak, suggesting the coexistence of multiple modes. The large interquartile range (36.55 hours) relative to the median further emphasizes the high variability in failure times.

A.0.0.2 Clustering Analysis

To support the selection of the number of mixture components, clustering algorithms were applied to the dataset. Table 21 summarizes the number of clusters recommended by each approach.

K-Means and Fuzzy C-means algorithms demonstrate convergence, both suggesting 8 clusters, indicating moderate granularity in failure pattern recognition. HDBSCAN's recommendation of 16 clusters suggests the presence of numerous density-based groupings, reflecting its sensitivity to local density variations and its ability to identify micro-clusters within the failure pattern. The Gaussian Mixture Model (GMM) approaches, using both BIC and AIC, recommend intermediate values of 9 and 15 components, respectively.

Figure 13 provides the visualization of the clustering analysis results across five different algorithms. The scatter plots reveal distinct clustering behaviors: K-Means (k=8) identifies multiple clusters with many observations concentrated in the low TBF region (0-50 hours) and several isolated high-TBF outliers. HDBSCAN (k=16) demonstrates its noise-handling capabilities by identifying higher TBF values as noise points (represented by gray crosses), suggesting that extremely long intervals may represent outliers rather than members of failure clusters.

Fuzzy C-Means (k=8) produces a clustering pattern similar to K-Means but with slightly different cluster boundaries, while the GMM approaches (BIC with k=9 and AIC with k=15) show intermediate clustering between HDBSCAN's fine partitioning and K-Means' moderate grouping. The Elbow and Silhouette analysis in the bottom-right panel demonstrates the trade-off between cluster quality metrics, with the Elbow method showing a sharp decline up to 8 components followed by gradual improvement, while the Silhouette score peaks around 2-3 clusters.

Table 21 – Cluster Results: G2Rac_MC1-074.

Clustering Approach	**Recommended Clusters**
K-Means	8
HDBSCAN	16
Fuzzy C-means	8
GMM (BIC)	9
GMM (AIC)	15

A.0.0.3 Mixture Model Selection and Validation

The selection of the best-fitting mixture model was based on the top five models identified by each GoF test, combined with a visual comparison of the graphical results for each candidate model. This procedure was applied across all samples. The final model selected via the Expectation-Maximization (EM) algorithm consisted of 10 components with the following distributional structure: logn-logn-norm-weib-weib-gamma-gamma-logn-norm-norm. This configuration was selected based on the outcomes of the AD goodness-of-fit test.

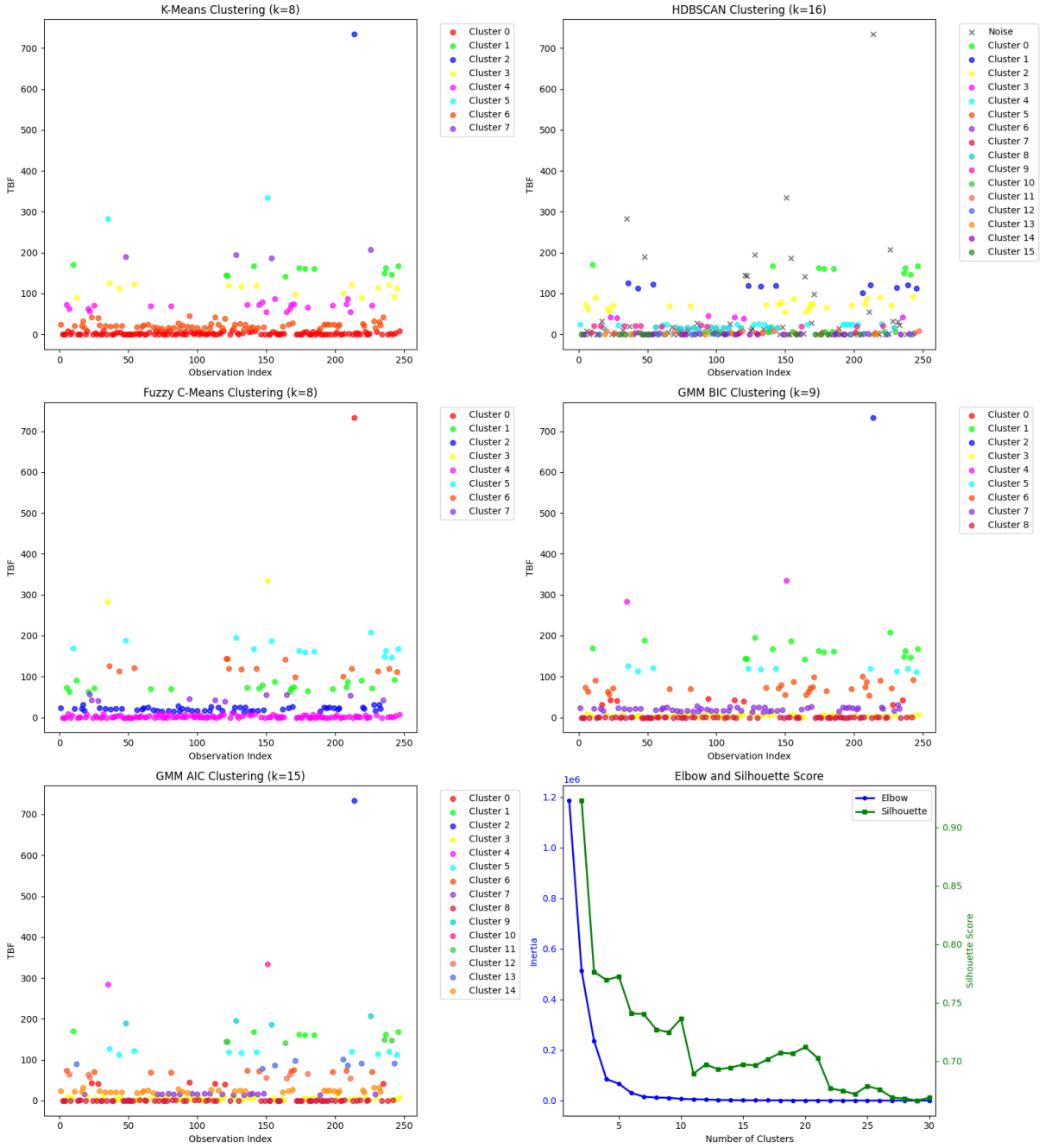

Figure 13 – Cluster Evaluation Plots for the Sample G2Rac_MC1-074.

Figure 14 demonstrates the fit achieved by the 10-component mixture model. The probability density function plot shows a highly multimodal distribution with a dominant peak near zero and several smaller peaks at intermediate values (50-150 hours) and longer intervals (300-700 hours).

The cumulative distribution function comparison reveals agreement between the empirical and theoretical distributions, with the theoretical CDF (orange crosses) closely overlaying the empirical CDF (blue circles) across the entire range. The probability-probability (P-P) plot shows linearity along the 45-degree line, indicating that the mixture model accurately represents the cumulative probability structure at all quantile levels.

The quantile-quantile plot further confirms model adequacy, with linear correspondence between theoretical and empirical quantiles. Minor deviations are observed only at the upper tail (around 700 hours). The overall linear pattern validates the mixture model's ability to capture both the central tendency and the extreme behavior of the failure process.

Table 22 summarizes the goodness-of-fit results for the selected model. The GoF test results

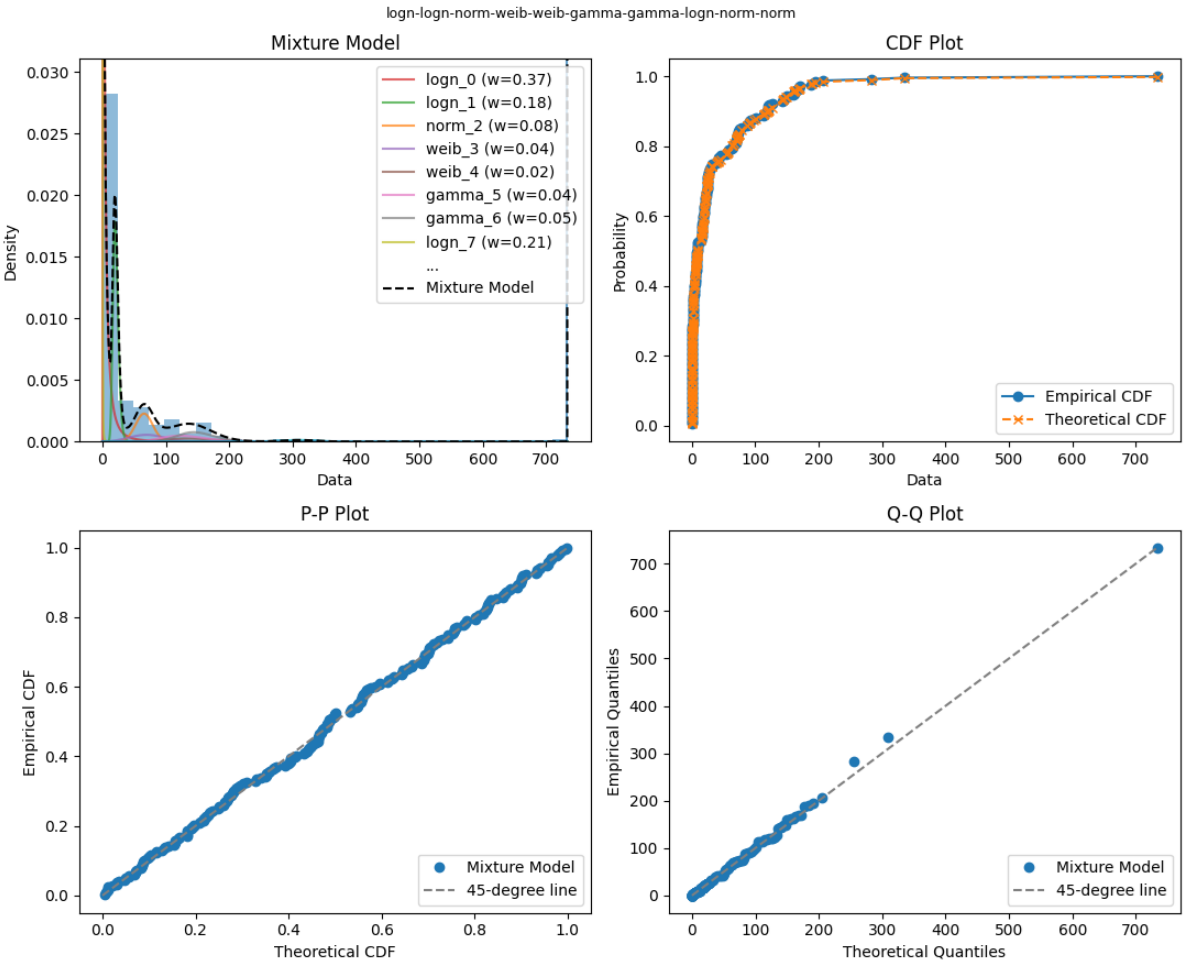

Figure 14 – G2Rac_MC1-074 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

provide statistical evidence for the adequacy of the 10-component mixture model. The KS test yields a low test statistic (0.0313) with a high p-value (0.9636), indicating that the null hypothesis of distributional equivalence cannot be rejected with confidence. The AD test, which is more sensitive to deviations in the distribution tails, produces a test statistic of -1.118 with a p-value of 0.9990. This result further confirms the model adequacy.

The log-likelihood value of -862.08 reflects the model's overall fit and provides a basis for comparison with alternative models. In addition, the AIC and the BIC values are 1796.16 and 1922.35, respectively, these criteria penalize model complexity.

Having established the adequacy of the selected mixture model in terms of global fit, the next step is to examine the estimated parameters associated with each mixture component. Table 23 presents the estimated parameters of the fitted mixture model.

Each distribution follows its standard statistical parameterization, respectively:

- ❏ Normal distribution uses the location (μ) and scale (σ) parameters.
- ❏ Lognormal distribution is parameterized by the shape (σ), location (μ), and scale (θ) parameters.
- ❏ Weibull distribution is characterized by the shape (k), location (μ), and scale (λ) parameters.

Table 22 – Goodness-of-fit Test Results: G2Rac_MC1-074.

Model	**Test**	**Statistic / p-value**
10-Component Mixture	KS	0.0313 / 0.9636
	AD	-1.118 / 0.9990
	Log-Likelihood	-862.08
	AIC	1796.16
	BIC	1922.35

Table 23 – Mixture Model Parameters: G2Rac_MC1-074.

Component	**Distribution**	**Weight**	**Param 1**	**Param 2**	**Param 3**
logn_0	Lognormal	0.3680	1.2799	1E-10	3.20
logn_1	Lognormal	0.1832	0.2015	1E-10	20.74
norm_2	Normal	0.0764	65.72	13.44	—
weib_3	Weibull	0.0363	3.0507	1E-10	79.37
weib_4	Weibull	0.0249	3.7468	1E-10	144.48
gamma_5	Gamma	0.0400	14.46	1E-10	9.12
gamma_6	Gamma	0.0541	26.60	1E-10	5.63
logn_7	Lognormal	0.2050	1.7491	1E-10	0.020
norm_8	Normal	0.0080	309.36	25.56	—
norm_9	Normal	0.0041	733.91	1E-10	—

- ❑ Gamma distribution is characterized by the shape (α), location (μ), and scale (θ) parameters.

The mixture weights indicate the relative importance of each failure subpopulation, with several dominant components: logn_0 (36.80%), logn_7 (20.50%), and logn_1 (18.32%). These high-weight components correspond to different failure regimes, from rapid succession failures (logn_7 with scale parameter 0.020) to intermediate stability periods (logn_0 with scale parameter 3.20).

The lognormal components dominate the mixture (3 out of 10 components, accounting for 75.62% of total weight), reflecting the multiplicative nature of the failure process. The presence of three normal components (norm_2, norm_8, norm_9) with high location parameters (65.72, 309.36, and 733.91 hours, respectively) captures the extended TBFs. The Weibull components (weib_3, weib_4) with shape parameters around 3.0-3.7 suggest wear-out behavior, while gamma components (gamma_5, gamma_6) provide additional flexibility for modeling intermediate failure patterns.

A.0.0.4 Sensitivity Analysis

To assess the robustness of the selected 10-component model and evaluate how model performance responds to changes in complexity, a sensitivity analysis was performed. This analysis investigated the behavior of GoF metrics as the number of mixture components increased. The sensitivity analysis was conducted by plotting the best GoF statistics obtained for each number of mixture components.

Figure 15 presents the sensitivity analysis results, examining the behavior of key goodness-of-fit statistics across component numbers. The KS test sensitivity analysis reveals a performance trajectory that supports moderate complexity selections. Starting from single-component performance, the statistic shows improvement through the initial component additions. The most significant improvement occurs in the 2-10 component range, with the selected 10-component model showing stable performance.

The sensitivity of the AD test provides complementary evidence by emphasizing distribution's tail behavior. As the number of components increases, the AD statistic shows improvement in the model's ability to represent distribution's tail behavior more accurately in multi-component configurations. However, oscillatory behavior becomes evident beyond 20 components, indicating instability at higher complexity levels.

Information criteria sensitivity analysis reveals the trade-off inherent in model selection. The AIC trajectory shows consistent improvement through moderate component numbers, with the minimum AIC value observed around 10-15 components. In contrast, the BIC analysis, which imposes a stronger penalty for model complexity, reaches its minimum at lower component numbers, reflecting its preference for parsimony.

Additionally, graphical analyses of mixture model fits were conducted by inspecting the best-fitting result according to each individual GoF test. The distributions selected as optimal by each test: AD, KS, AIC, and BIC rarely coincided. In particular, the models selected by AIC and BIC consistently exhibited poorer visual and statistical fit compared to those chosen by the AD and KS tests. This performance gap persisted even in the cases where AIC and BIC selected models with a higher number of components.

A.0.0.5 Comparison with Single-Component Models

For comparison purposes, the Weibull distribution provided the best fit among the single-component models evaluated, as reported in Table 24. Despite its superior performance relative to the other single distributions, the model was unable to adequately capture the underlying complexity of the data. This result indicates that, although suitable as a baseline, a single Weibull distribution is insufficient to represent the heterogeneous behavior observed in the time between failures.

Goodness-of-fit testing highlights this difference in representational adequacy: while the

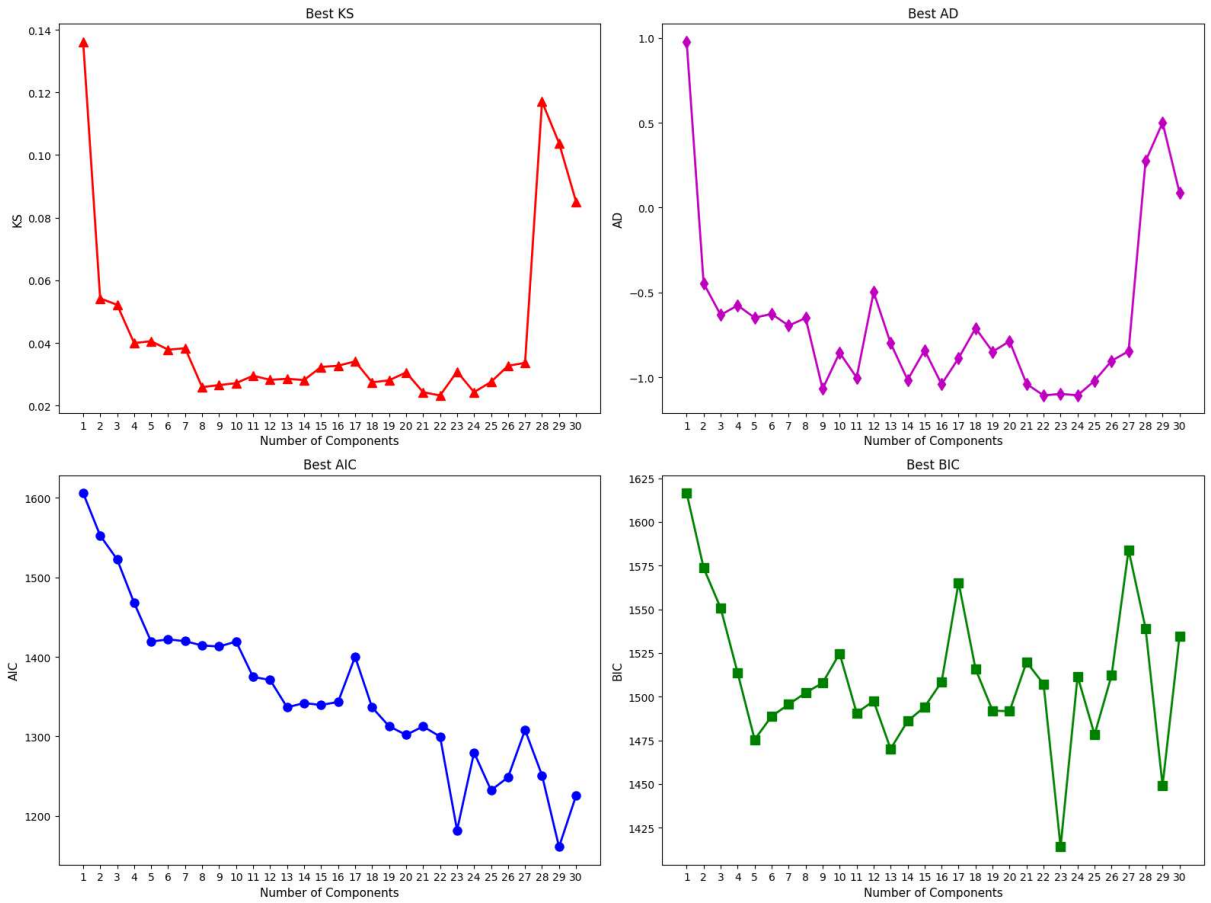

Figure 15 – G2Rac_MC1-074 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components.

Weibull distribution yields low p-values for both the Kolmogorov–Smirnov (KS = 0.0940, p = 0.0241) and Anderson–Darling (AD = 1.583, p = 0.0679) tests, indicating marginal adequacy. The mixture model achieves much stronger agreement with the data (KS = 0.0313, p = 0.9636; AD = -1.118, p = 0.9990). And the log-likelihood comparison also shows the superior fit of the mixture model (-862.08 vs. -922.75).

The information criteria (AIC and BIC) provide a more nuanced comparison. While the AIC strongly favors the mixture model (1796.16 vs. 1851.49), the BIC comparison shows the Weibull model achieving slightly better performance (1862.01 vs. 1922.35) due to stronger complexity penalties. However, this BIC advantage for the simple model comes at the cost of substantially poorer distributional fit, as evidenced by the likelihood and goodness-of-fit test results.

Figure 16 illustrates the limitations of single-distribution modeling for this failure dataset. The PDF overlay reveals that the Weibull model fails to represent the sharp peak near zero and completely misses the multimodal nature evident in the histogram.

The CDF comparison demonstrates significant deviations between the theoretical Weibull distribution (orange crosses) and the empirical distribution (blue circles), particularly in the lower tail region (0-100 hours) where the model underestimates failure probabilities, and in the upper tail where it overestimates them.

The P-P plot shows departures from linearity, with an S-shaped curve. The Q-Q plot reveals more nonlinearities, with the data points forming a curved pattern rather than following the expected 45-degree line.

A.0.0.6 Computational Cost Analysis

Given that mixture models can be computationally intensive, particularly when evaluating multiple distribution types and component numbers, a computational cost analysis was conducted. This analysis examines the trade-offs between model complexity and computational requirements.

Figure 17 and 18 present the computational cost analysis, showing execution time, memory usage, CPU utilization, and operation counts across different distribution types and component numbers.

Table 24 – Mixture vs. Single Distributions: G2Rac_MC1-074.

GOF Metric	**Mixture Model (10-comp)**	**Weibull Simple Distribution**
KS / p-value	0.0313 / 0.9636	0.0940 / 0.0241
AD / p-value	-1.118 / 0.9990	1.583 / 0.0679
Log-Likelihood	-862.08	-922.75
AIC	1796.16	1851.49
BIC	1922.35	1862.01

Figure 17 presents the computational cost analysis for homogeneous mixture models across varying numbers of components. The analysis reveals distinct computational characteristics for different distribution families.

Figure 18 extends the analysis to heterogeneous mixture models, where components can belong to different distribution families. This analysis is particularly relevant for the selected 10-component model, which combines lognormal, normal, Weibull, and gamma distributions.

The computational analysis reveals the scaling behavior of mixture model estimation as a function of both distribution complexity and the number of mixture components. These results inform practical considerations for large-scale reliability analysis applications. While the computational cost is substantial, the significant improvement in model adequacy and the critical importance of accurate failure modeling in reliability applications justify the computational investment.

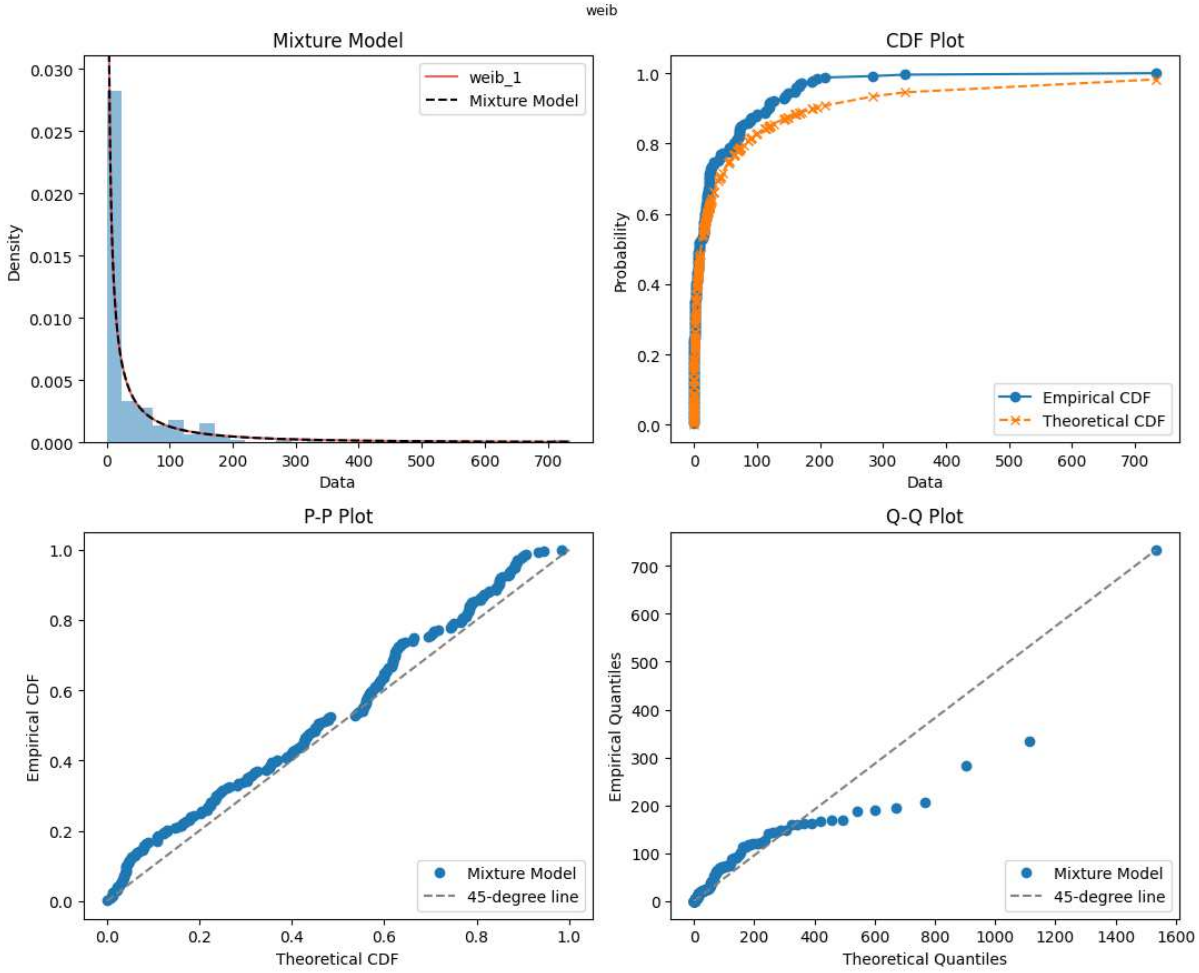

Figure 16 – G2Rac_MC1-074 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

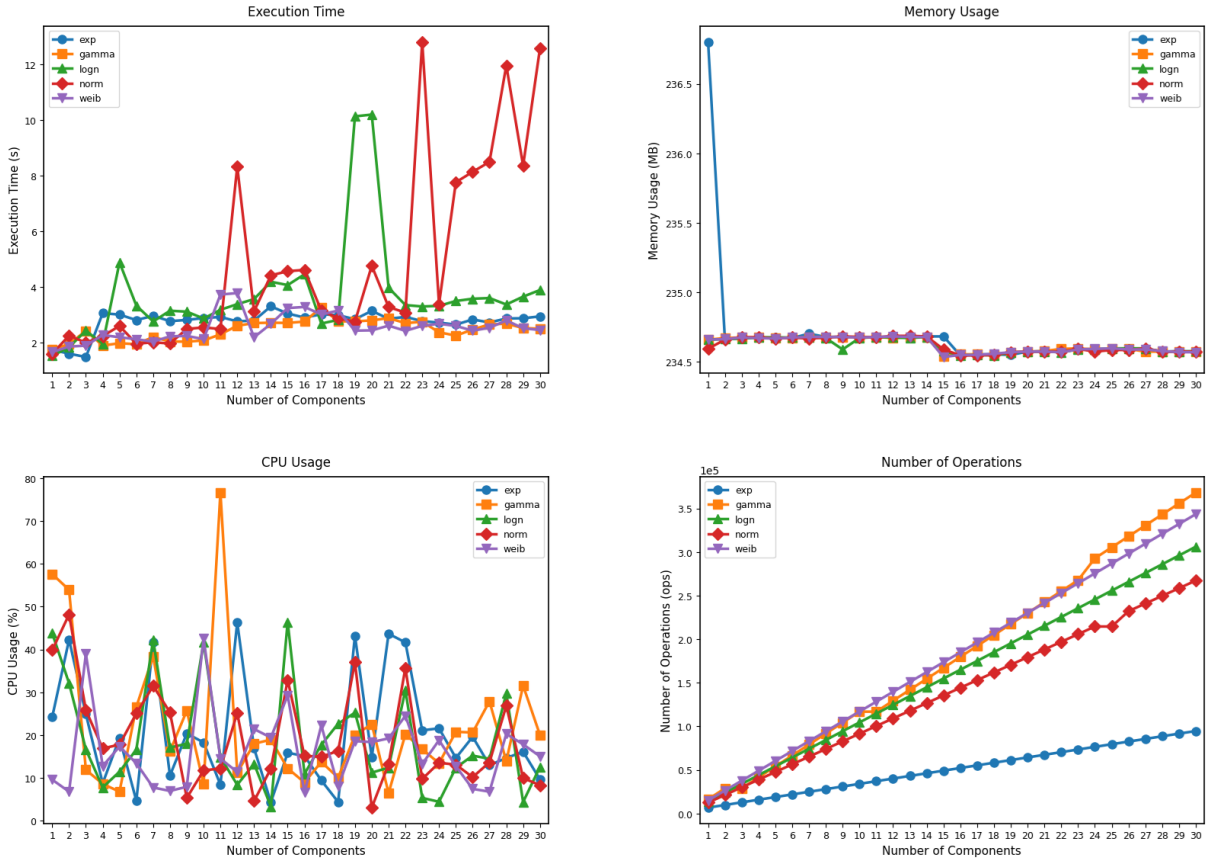

Figure 17 – G2Rac_MC1-074 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

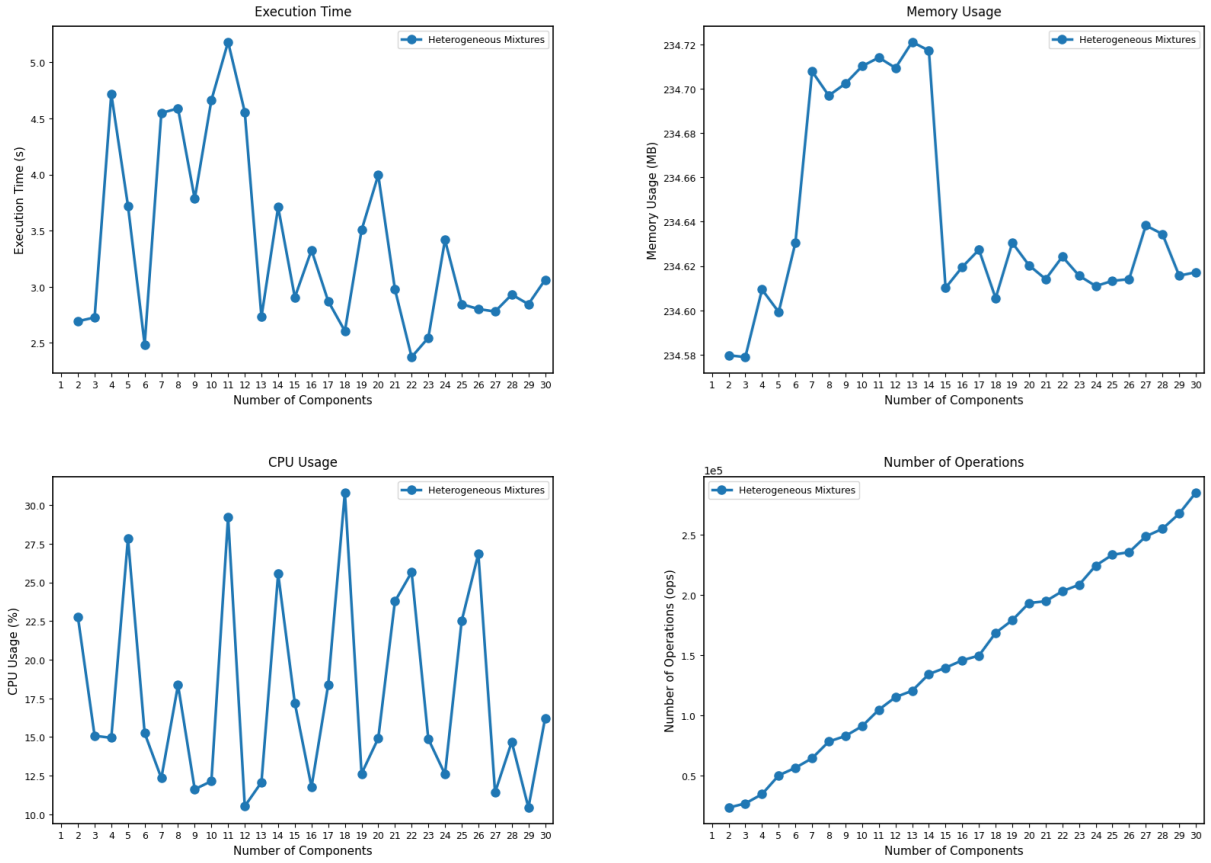

Figure 18 – G2Rac_MC1-074 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

APPENDIX **B**

G2Rac_MC1-074_iexplore_c0000005 (Approach 2)

This case refers to failures related to access violations (c0000005), associated with invalid memory addressing, occurring in the `iexplore.exe`'s process on computer MC1-074, part of the Group 2 (Graduate Laboratory environment).

B.0.0.1 Statistical Characterization

Table 25 presents the statistical characteristics of the TBF data for this sample. The descriptive statistics reveal a heterogeneous failure pattern characterized by significant asymmetry and heavy-tailed behavior. The difference between mean (197.72 hours) and median (106.50 hours) indicates a right-skewed distribution, which is further confirmed by the positive skewness value of 1.35.

The extremely low mode value (0.003 hours) combined with the minimum observation

Table 25 – Descriptive Statistics: G2Rac_MC1-074_iexplore_c0000005.

Statistic	**Value**
Count	36
Mean (hours)	197.72
Median (hours)	106.50
Mode (hours)	0.003
Standard Deviation	227.60
Minimum	0.003
Maximum	908.48
First Quartile (Q1)	23.43
Third Quartile (Q3)	304.43
Interquartile Range (IQR)	280.99
Skewness	1.35
Kurtosis	1.04
Main Data Range	0.46 – 631.92

(0.003 hours) indicates that some failures occur in rapid succession, possibly representing cascading failures or immediate re-failures after system recovery attempts.

The moderate kurtosis value (1.04) indicates a distribution with moderate tail behavior and peak sharpness. The large interquartile range (280.99 hours) relative to the median further emphasizes the high variability in failure times.

B.0.0.2 Clustering Analysis

To support the selection of the number of mixture components, clustering algorithms were applied to the dataset. Table 26 summarizes the number of clusters recommended by each approach.

K-Means and Fuzzy C-means algorithms demonstrate convergence, both suggesting 13 clusters, indicating moderate granularity in failure pattern recognition. HDBSCAN failed to identify meaningful clusters, returning an indeterminate result, suggesting that the clusters in this dataset are better characterized by distance-based rather than density-based criteria. The Gaussian Mixture Model (GMM) approaches, using both BIC and AIC, recommend higher values of 18 and 25 components, respectively.

Figure 19 provides the visualization of the clustering analysis results across five different algorithms. The scatter plots reveal distinct clustering behaviors: K-Means (k=13) identifies numerous clusters with clear separation between low, medium, and high TBF groups. HDBSCAN shows no meaningful clustering pattern, with most data points unclassified.

Fuzzy C-Means (k=13) produces a clustering pattern similar to K-Means but with slightly different cluster boundaries, while the GMM approaches (BIC with k=18 and AIC with k=25) show finer granularity in partitioning the data space. The Elbow and Silhouette analysis in the bottom-right panel demonstrates the trade-off between cluster quality metrics, with the Elbow method showing gradual improvement, while the Silhouette score peaks at lower cluster numbers.

B.0.0.3 Mixture Model Selection and Validation

The selection of the best-fitting mixture model was based on the top five models identified by each GoF test, combined with a visual comparison of the graphical results for each

Table 26 – Cluster Results: G2Rac_MC1-074_iexplore_c0000005.

Clustering Approach	Recommended Clusters
K-Means	13
HDBSCAN	N/A
Fuzzy C-means	13
GMM (BIC)	18
GMM (AIC)	25

candidate model. This procedure was applied across all samples. The final model selected via the Expectation-Maximization (EM) algorithm consisted of 4 components with the following distributional structure: gamma-gamma-gamma-norm. This configuration was selected based on the outcomes of the KS goodness-of-fit test.

Figure 20 demonstrates the fit achieved by the 4-component mixture model. The probability density function plot shows a multimodal distribution with peaks distributed across the TBF spectrum (0-900 hours).

The cumulative distribution function comparison reveals agreement between the empiri-

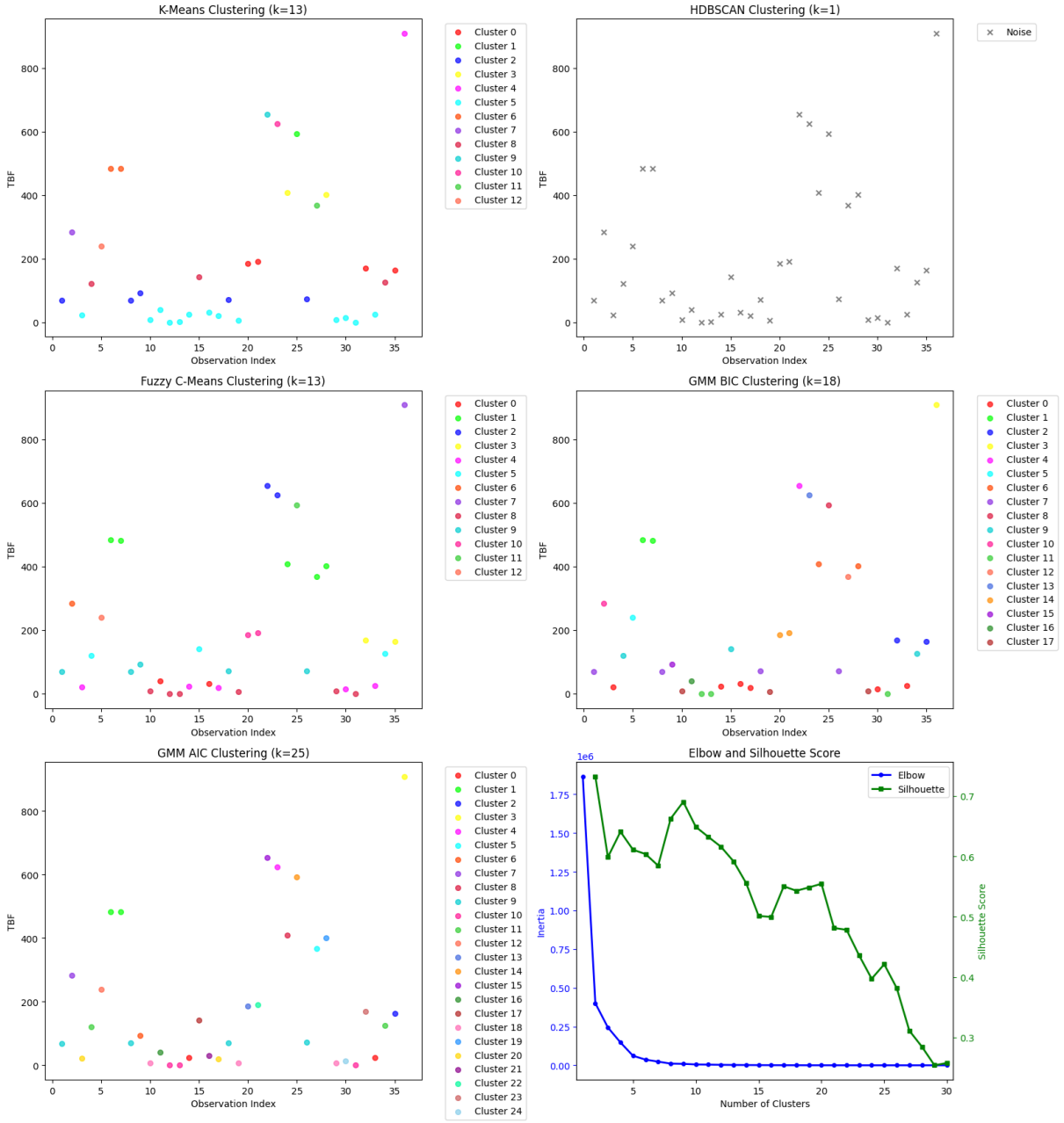

Figure 19 – Cluster Evaluation Plots for the Sample G2Rac_MC1-074_iexplore_c0000005.

cal and theoretical distributions, with the theoretical CDF (orange crosses) closely overlaying the empirical CDF (blue circles) across the entire range. The probability-probability (P-P) plot shows linearity along the 45-degree line, indicating that the mixture model accurately represents the cumulative probability structure at all quantile levels.

The quantile-quantile plot further confirms model adequacy, with linear correspondence between theoretical and empirical quantiles. Minor deviations are observed only at the extreme tail regions. The overall linear pattern validates the mixture model's ability to capture both the central tendency and the extreme behavior of the failure process.

Table 27 summarizes the goodness-of-fit results for the selected model. The GoF test results provide statistical evidence for the adequacy of the 4-component mixture model. The KS test yields a low test statistic (0.052) with an extremely high p-value (0.9999), indicating that the null hypothesis of distributional equivalence cannot be rejected with confidence. The AD test, which is more sensitive to deviations in the distribution tails, produces a test statistic of -0.633 with a p-value of 0.708. This result further confirms the model adequacy.

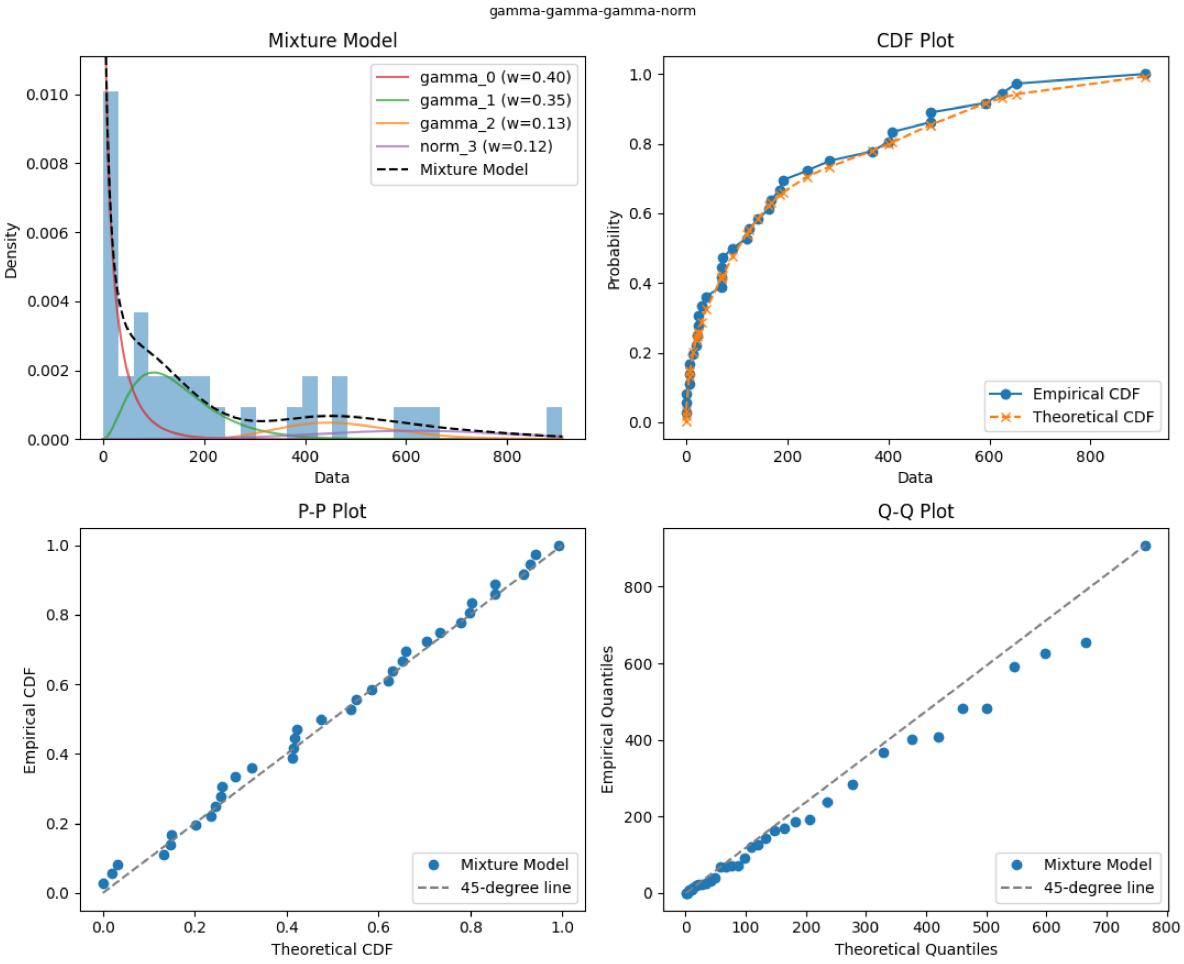

Figure 20 – G2Rac_MC1-074_iexplore_c0000005 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

The AIC and the BIC values are 462.64 and 484.81, respectively, these criteria penalize model complexity.

Having established the adequacy of the selected mixture model in terms of global fit, the next step is to examine the estimated parameters associated with each mixture component. Table 28 presents the estimated parameters of the fitted mixture model.

Each distribution follows its standard statistical parameterization, respectively:

- ❏ Normal distribution uses the location (μ) and scale (σ) parameters.
- ❏ Gamma distribution is characterized by the shape (α), location (μ), and scale (θ) parameters.

The mixture weights indicate the relative importance of each failure subpopulation, with several dominant components: gamma_0 (40.0%), gamma_1 (34.9%), and gamma_2 (13.5%). These high-weight components correspond to different failure regimes, from highly variable short-to-moderate execution times (gamma_0 with shape parameter 0.587) to more consistent moderate-range patterns (gamma_1 with shape parameter 3.110).

The gamma components dominate the mixture (3 out of 4 components, accounting for 88.4% of total weight), reflecting the multiplicative nature of the failure process. The presence of one normal component (norm_3) with high location parameter (614.96 hours) captures the extended TBFs in the upper tail region. The varying gamma shape parameters (0.587, 3.110, and 16.646) suggest different degrees of variability, with higher shape values indicating more concentrated distributions around the mode.

Table 27 – Goodness-of-fit Test Results: G2Rac_MC1-074_iexplore_c0000005.

Model	**Test**	**Statistic / p-value**
4-Component Mixture	KS	0.052 / 0.9999
	AD	-0.633 / 0.708
	Log-Likelihood	-217.32
	AIC	462.64
	BIC	484.81

Table 28 – Mixture Model Parameters: G2Rac_MC1-074_iexplore_c0000005.

Component	**Distribution**	**Weight**	**Param 1**	**Param 2**	**Param 3**
gamma_0	Gamma	0.400	0.587	1E-10	47.23
gamma_1	Gamma	0.349	3.110	1E-10	47.68
gamma_2	Gamma	0.135	16.646	1E-10	28.43
norm_3	Normal	0.116	614.96	191.75	—

B.0.0.4 Sensitivity Analysis

To assess the robustness of the selected 4-component model and evaluate how model performance responds to changes in complexity, a sensitivity analysis was performed. This analysis investigated the behavior of GoF metrics as the number of mixture components increased. The sensitivity analysis was conducted by plotting the best GoF statistics obtained for each number of mixture components.

Figure 21 presents the sensitivity analysis results, examining the behavior of key goodness-of-fit statistics across component numbers. The KS test sensitivity analysis reveals stable performance across the 4-20 component range, supporting the selection of a simpler model that maintains adequate fit quality.

The sensitivity of the AD test provides complementary evidence by emphasizing distribution's tail behavior. The AD statistic exhibits greater variability than the KS statistic, achieving optimal performance around 15-17 components before showing increased variation at higher component numbers, indicating instability at higher complexity levels.

Information criteria sensitivity analysis reveals the trade-off inherent in model selection.

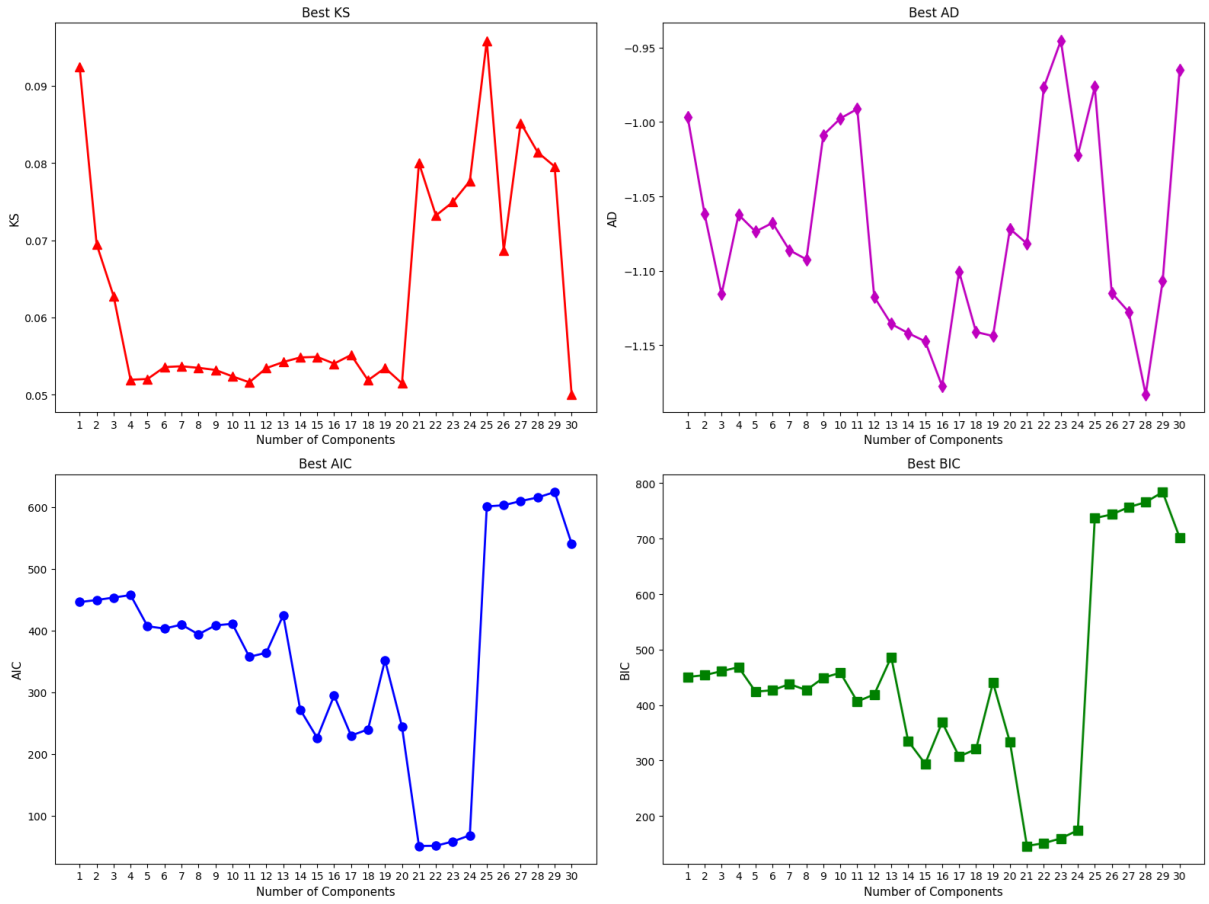

Figure 21 – G2Rac_MC1-074_iexplore_c0000005 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components.

The AIC trajectory shows a clear and pronounced minimum at 21 components, increasing for both lower and higher component counts. In contrast, the BIC analysis, which imposes a stronger penalty for model complexity, exhibits a similar pattern, achieving minimum values around 21-24 components before increasing sharply.

Additionally, graphical analyses of mixture model fits were conducted by inspecting the best-fitting result according to each individual GoF test. The distributions selected as optimal by each test: AD, KS, AIC, and BIC rarely coincided. In particular, the models selected by AIC and BIC consistently exhibited poorer visual and statistical fit compared to those chosen by the AD and KS tests. This performance gap persisted even in the cases where AIC and BIC selected models with a higher number of components.

B.0.0.5 Comparison with Single-Component Models

For comparison, the best-fitting single distribution was the lognormal model, as shown in Table 29. In this particular case, the single-component model achieved comparable performance to the mixture model while maintaining greater parsimony.

While mixture models often provide superior fit for complex failure patterns, this case demonstrates that single-distribution models can be adequate when the underlying data structure is less complex. In this particular sample, the simpler lognormal model provides sufficient representational adequacy.

Goodness-of-fit testing reveals comparable performance: while the mixture model achieves excellent fit (KS = 0.052, p = 0.9999; AD = -0.633, p = 0.708), the lognormal distribution also demonstrates adequate fit (KS = 0.1222, p = 0.6121; AD = -0.451, p = 0.6067). Both models show no significant departure from the empirical distribution.

The information criteria (AIC and BIC) favor the simpler model in this case. The lognormal distribution achieves lower AIC (458.30 vs. 462.64) and BIC (463.05 vs. 484.81) values compared to the mixture model, indicating that the additional complexity of the mixture approach is not justified by sufficient improvement in fit quality.

Figure 22 illustrates the fit quality of single-distribution modeling for this failure dataset. The PDF overlay reveals that the lognormal model captures the main characteristics of the distribution, though some minor deviations are visible in the histogram representation.

Table 29 – Mixture vs. Single Distributions: G2Rac_MC1-074_iexplore_c0000005.

GOF Metric	Mixture Model (4-comp)	Lognormal Simple Distribution
KS / p-value	0.052 / 0.9999	0.1222 / 0.6121
AD / p-value	-0.633 / 0.708	-0.451 / 0.6067
Log-Likelihood	-217.32	-290.13
AIC	462.64	458.30
BIC	484.81	463.05

The CDF comparison demonstrates good agreement between the theoretical lognormal distribution (orange crosses) and the empirical distribution (blue circles) across most of the data range. The P-P plot shows reasonable linearity along the 45-degree line, with minor departures. The Q-Q plot reveals acceptable correspondence between theoretical and empirical quantiles, with deviations primarily in the tail regions.

B.0.0.6 Computational Cost Analysis

Given that mixture models can be computationally intensive, particularly when evaluating multiple distribution types and component numbers, a computational cost analysis was conducted. This analysis examines the trade-offs between model complexity and computational requirements.

Figure 23 and 24 present the computational cost analysis, showing execution time, memory usage, CPU utilization, and operation counts across different distribution types and component numbers.

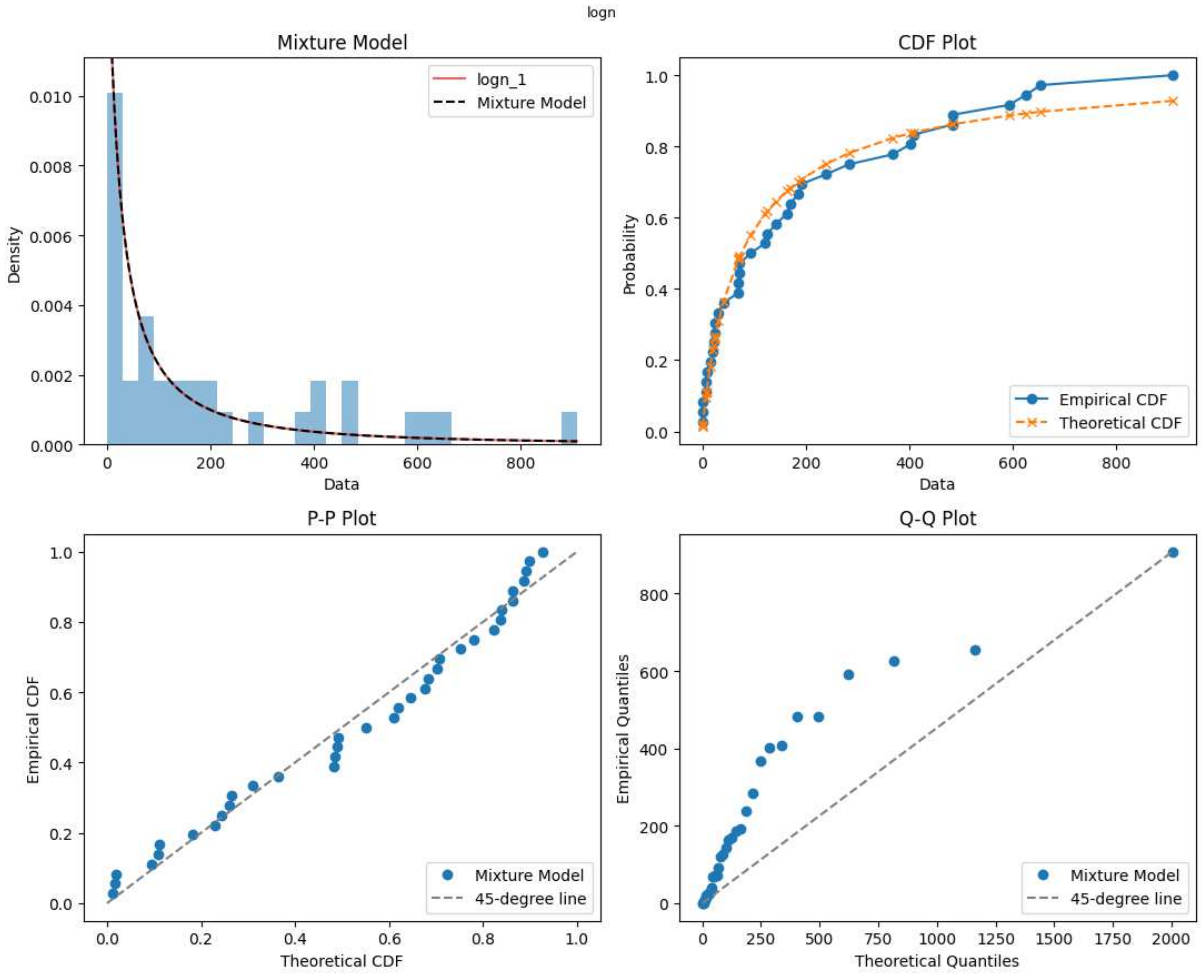

Figure 22 – G2Rac_MC1-074_iexplore_c0000005 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

Figure 23 presents the computational cost analysis for homogeneous mixture models across varying numbers of components. The analysis reveals distinct computational characteristics for different distribution families.

Figure 24 extends the analysis to heterogeneous mixture models, where components can belong to different distribution families. This analysis is particularly relevant for the selected 4-component model, which combines gamma and normal distributions.

The computational analysis reveals the scaling behavior of mixture model estimation as a function of both distribution complexity and the number of mixture components. These results inform practical considerations for large-scale reliability analysis applications. While the computational cost is substantial, the significant improvement in model adequacy and the critical importance of accurate failure modeling in reliability applications justify the computational investment.

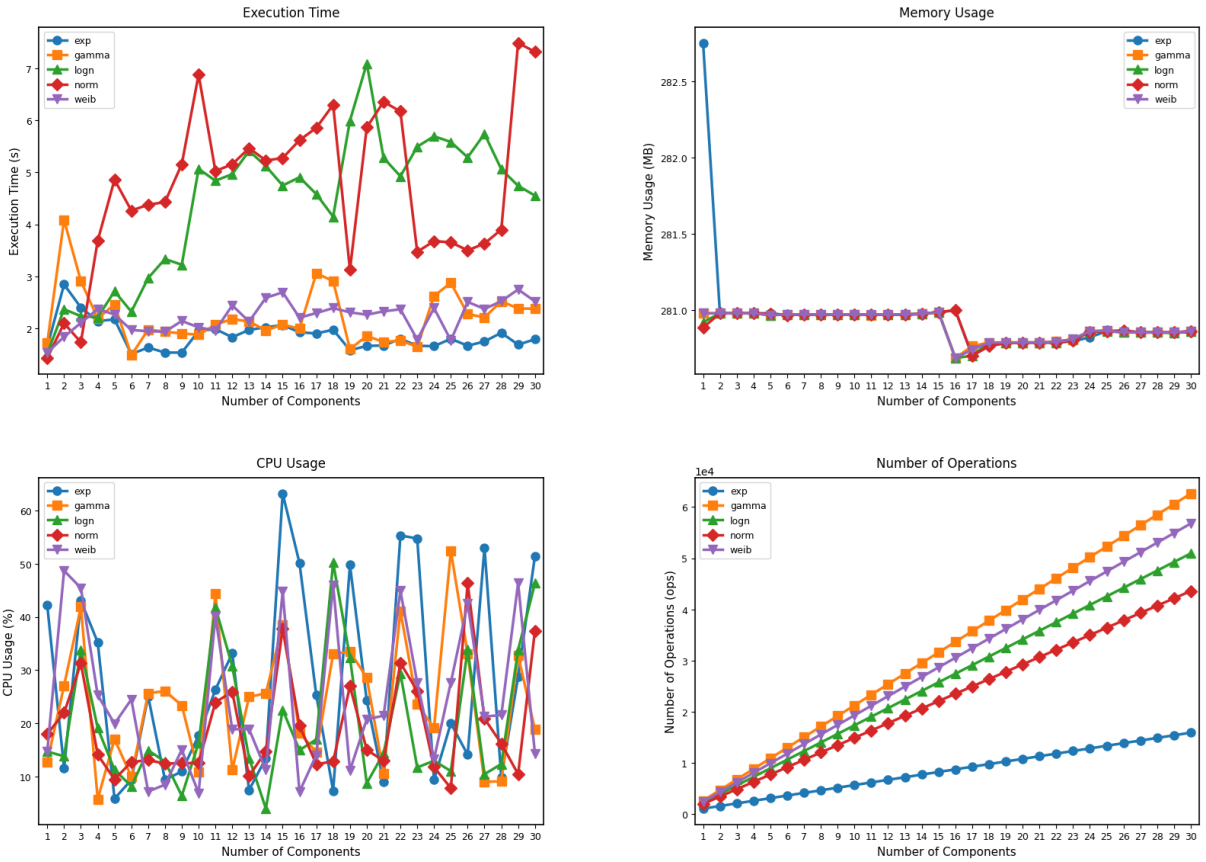

Figure 23 – G2Rac_MC1-074_iexplore_c0000005 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

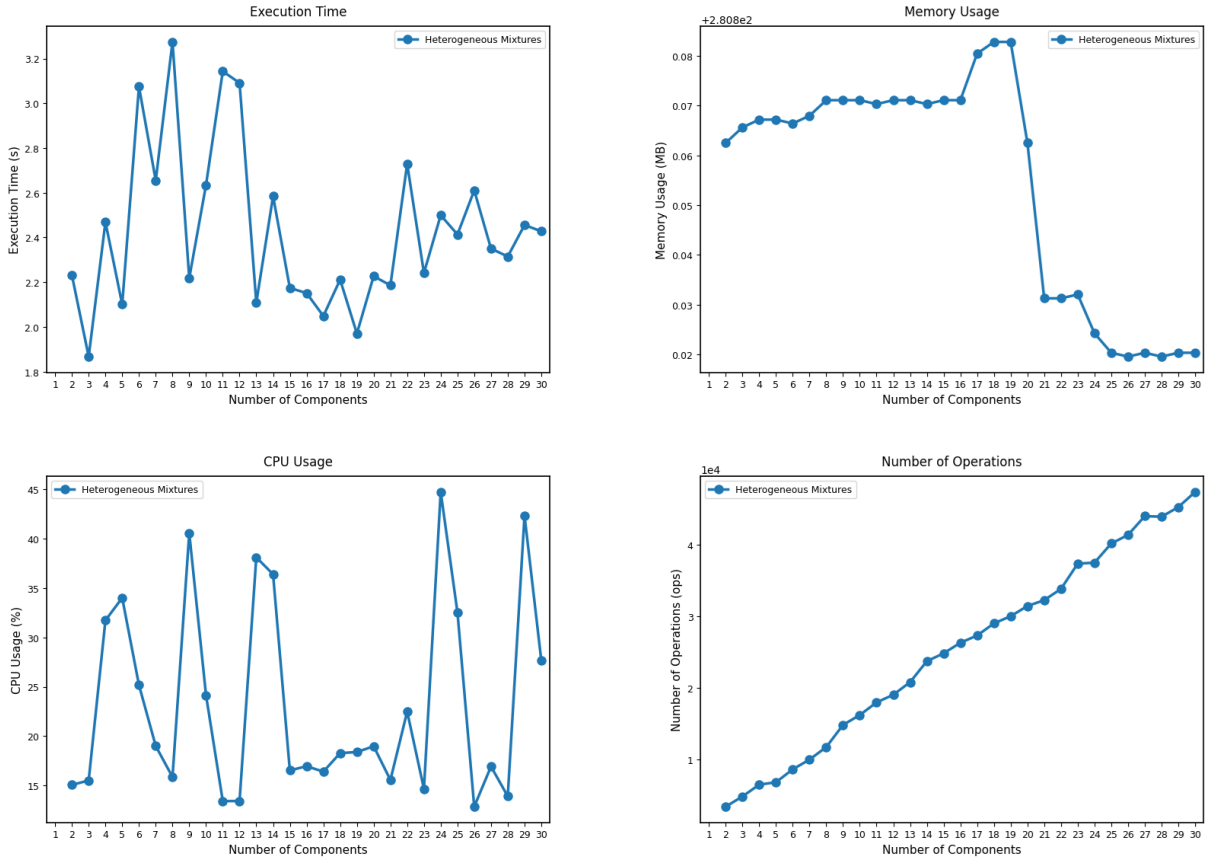

Figure 24 – G2Rac_MC1-074_iexplore_c0000005 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

APPENDIX C

Results G2Rac-MC1-074-iexplore-c0000096 (Approach 2)

This case refers to failures related to privileged instruction violation (c0000096), occurring in the `iexplore.exe`'s process on computer MC1-074, part of the Group 2 (University Administrative Department).

C.0.0.1 Statistical Characterization

Table 30 presents the statistical characteristics of the TBF data for this sample. The descriptive statistics reveal a heterogeneous failure pattern characterized by significant asymmetry and moderate heavy-tailed behavior. The difference between mean (20.15 hours) and median (16.26 hours) indicates a right-skewed distribution, which is further confirmed by the positive skewness value of 1.66.

The extremely low mode value (0.001 hours) combined with the minimum observation (0.001 hours) indicates that some failures occur in rapid succession, possibly representing cascading failures or immediate re-failures after system recovery attempts.

The moderate kurtosis value (2.43) indicates a distribution with moderate tail behavior and peak sharpness. The large interquartile range (23.53 hours) relative to the median further emphasizes the high variability in failure times.

C.0.0.2 Clustering Analysis

To support the selection of the number of mixture components, clustering algorithms were applied to the dataset. Table 31 summarizes the number of clusters recommended by each approach.

Fuzzy C-means and GMM (BIC) algorithms demonstrate convergence, both suggesting 9 clusters, indicating moderate granularity in failure pattern recognition. HDBSCAN's recom-

mendation of 2 clusters suggests the presence of two major density-based groupings, reflecting its conservative approach to cluster identification. K-Means suggests 8 clusters, while the Gaussian Mixture Model (GMM) with AIC recommends 11 components.

Figure 25 provides the visualization of the clustering analysis results across five different algorithms. The scatter plots reveal distinct clustering behaviors: K-Means (k=8) identifies multiple clusters with well-separated groups and clear boundaries. HDBSCAN (k=2) shows a conservative approach, identifying only two main clusters plus noise points (represented by gray crosses), focusing on high-density regions.

Fuzzy C-Means (k=9) produces a clustering pattern with slightly finer granularity, while the GMM approaches (BIC with k=9 and AIC with k=11) show similar moderate clustering. The Elbow and Silhouette analysis in the bottom-right panel demonstrates the trade-off between cluster quality metrics, with the Elbow method showing improvement up to 8-9 components, while the Silhouette score indicates optimal performance at lower cluster numbers.

C.0.0.3 Mixture Model Selection and Validation

The selection of the best-fitting mixture model was based on the top five models identified by each GoF test, combined with a visual comparison of the graphical results for each candi-

Table 30 – Descriptive Statistics: G2Rac_MC1-074_iexplore_c0000096.

Statistic	Value
Count	36
Mean (hours)	20.15
Median (hours)	16.26
Mode (hours)	0.001
Standard Deviation	23.61
Minimum	0.001
Maximum	95.60
First Quartile (Q1)	1.49
Third Quartile (Q3)	25.02
Interquartile Range (IQR)	23.53
Skewness	1.66
Kurtosis	2.43
Main Data Range	0.001 – 95.60

Table 31 – Cluster Results: G2Rac_MC1-074_iexplore_c0000096.

Clustering Approach	Recommended Clusters
K-Means	8
HDBSCAN	2
Fuzzy C-means	9
GMM (BIC)	9
GMM (AIC)	11

date model. This procedure was applied across all samples. The final model selected via the Expectation-Maximization (EM) algorithm consisted of 8 components with the following distributional structure: logn-logn-logn-logn-norm-logn-logn-gamma. This configuration was selected based on the outcomes of the AD goodness-of-fit test.

Figure 26 demonstrates the fit achieved by the 8-component mixture model. The probability density function plot shows a multimodal distribution with a dominant peak near zero and several smaller peaks at intermediate values (10-50 hours) and longer intervals (60-90 hours).

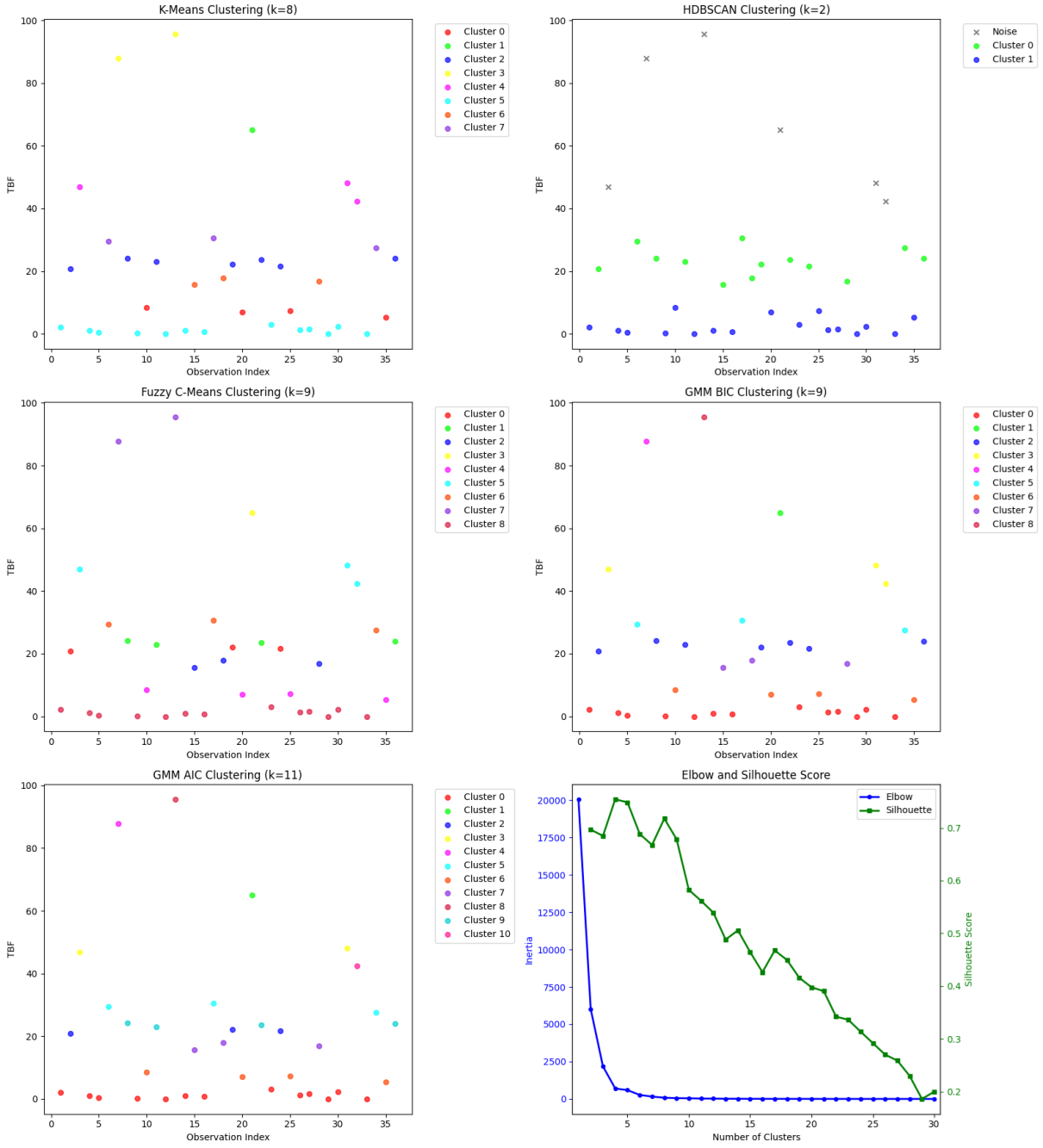

Figure 25 – Cluster Evaluation Plots for the Sample G2Rac_MC1-074_iexplore_c0000096.

The cumulative distribution function comparison reveals agreement between the empirical and theoretical distributions, with the theoretical CDF (orange crosses) closely overlaying the empirical CDF (blue circles) across the entire range. The probability-probability (P-P) plot shows linearity along the 45-degree line, indicating that the mixture model accurately represents the cumulative probability structure at all quantile levels.

The quantile-quantile plot further confirms model adequacy, with linear correspondence between theoretical and empirical quantiles. Minor deviations are observed only at the extreme tail regions. The overall linear pattern validates the mixture model's ability to capture both the central tendency and the extreme behavior of the failure process.

Table 32 summarizes the goodness-of-fit results for the selected model. The GoF test results provide statistical evidence for the adequacy of the 8-component mixture model. The KS test yields a low test statistic (0.055) with an extremely high p-value (0.9997), indicating that the null hypothesis of distributional equivalence cannot be rejected with confidence. The AD test, which is more sensitive to deviations in the distribution tails, produces a test statistic of -1.151

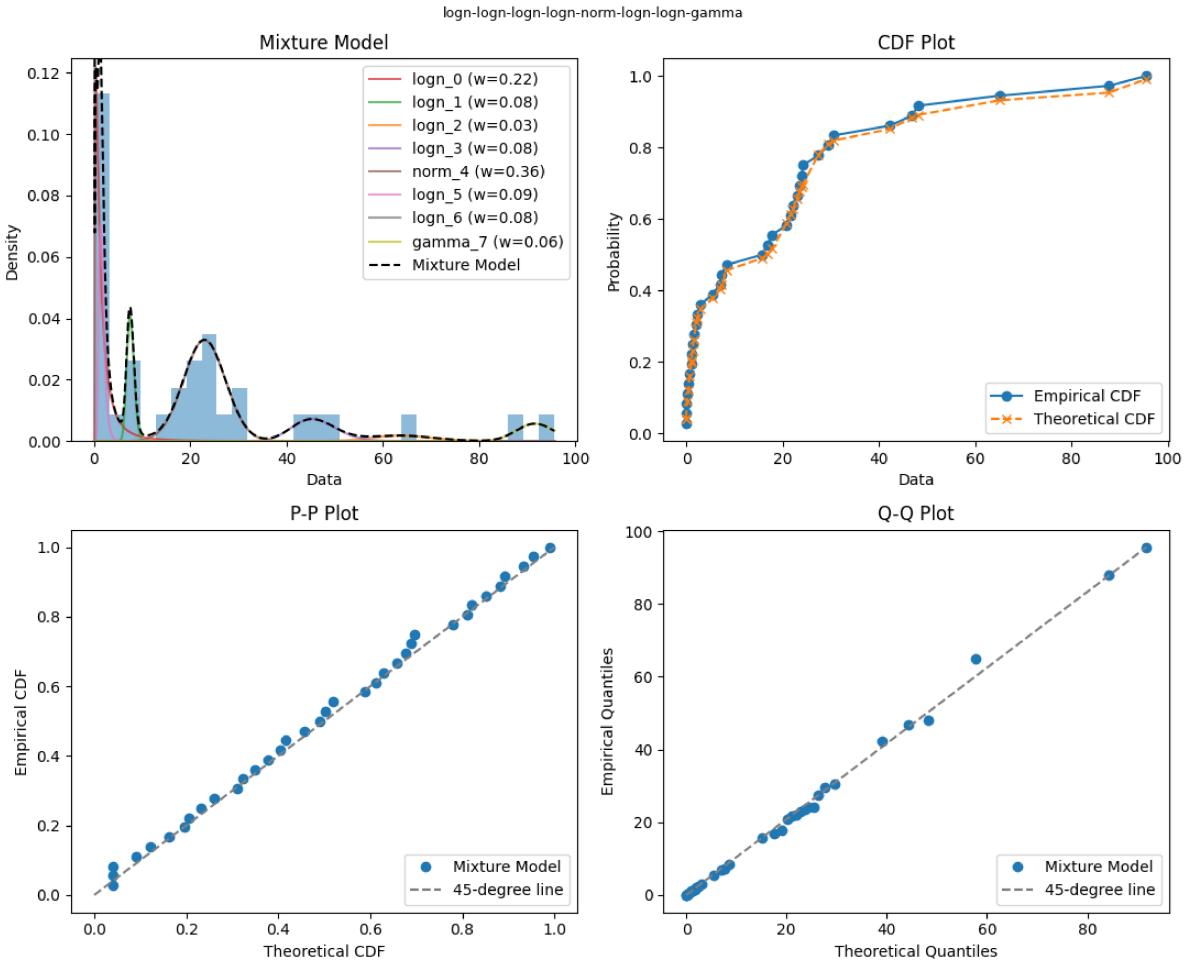

Figure 26 – G2Rac_MC1-074_iexplore_c0000096 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

with a p-value of 0.989. This result further confirms the model adequacy.

The log-likelihood value of -106.94 reflects the model's overall fit and provides a basis for comparison with alternative models. In addition, the AIC and the BIC values are 273.88 and 321.39, respectively, these criteria penalize model complexity.

Having established the adequacy of the selected mixture model in terms of global fit, the next step is to examine the estimated parameters associated with each mixture component. Table 33 presents the estimated parameters of the fitted mixture model.

Each distribution follows its standard statistical parameterization, respectively:

- ❏ Normal distribution uses the location (μ) and scale (σ) parameters.
- ❏ Lognormal distribution is parameterized by the shape (σ), location (μ), and scale (θ) parameters.
- ❏ Gamma distribution is characterized by the shape (α), location (μ), and scale (θ) parameters.

The mixture weights indicate the relative importance of each failure subpopulation, with several dominant components: norm_4 (35.8%), logn_0 (22.1%), and logn_5 (9.5%). These high-weight components correspond to different failure regimes, from stable operational periods (norm_4 with mean 22.917 hours) to multiplicative failure patterns (logn_0 with scale parameter 1.208).

Table 32 – Goodness-of-fit Test Results: G2Rac_MC1-074_iexplore_c0000096.

Model	**Test**	**Statistic / p-value**
8-Component Mixture	KS	0.055 / 0.9997
	AD	-1.151 / 0.989
	Log-Likelihood	-106.94
	AIC	273.88
	BIC	321.39

Table 33 – Mixture Model Parameters: G2Rac_MC1-074_iexplore_c0000096.

Component	**Distribution**	**Weight**	**Param 1**	**Param 2**	**Param 3**
logn_0	Lognormal	0.221	1.135	1E-10	1.208
logn_1	Lognormal	0.077	0.100	1E-10	7.532
logn_2	Lognormal	0.029	0.100	1E-10	64.198
logn_3	Lognormal	0.082	0.100	1E-10	45.715
norm_4	Normal	0.358	22.917	4.340	—
logn_5	Lognormal	0.095	0.376	1E-10	1.540
logn_6	Lognormal	0.083	0.100	1E-10	0.001
gamma_7	Gamma	0.055	552.614	1E-10	0.166

The lognormal components dominate the mixture (6 out of 8 components, accounting for 58.7% of total weight), reflecting the multiplicative nature of the failure process. The presence of one normal component (norm_4) with moderate location parameter (22.917 hours) captures the central tendency with relatively low variability (standard deviation 4.340). The single gamma component (gamma_7) with an extremely high shape parameter (552.614) captures a highly peaked distribution with very low variability, likely representing highly consistent, short execution times.

C.0.0.4 Sensitivity Analysis

To assess the robustness of the selected 8-component model and evaluate how model performance responds to changes in complexity, a sensitivity analysis was performed. This analysis investigated the behavior of GoF metrics as the number of mixture components increased. The sensitivity analysis was conducted by plotting the best GoF statistics obtained for each number of mixture components.

Figure 27 presents the sensitivity analysis results, examining the behavior of key goodness-

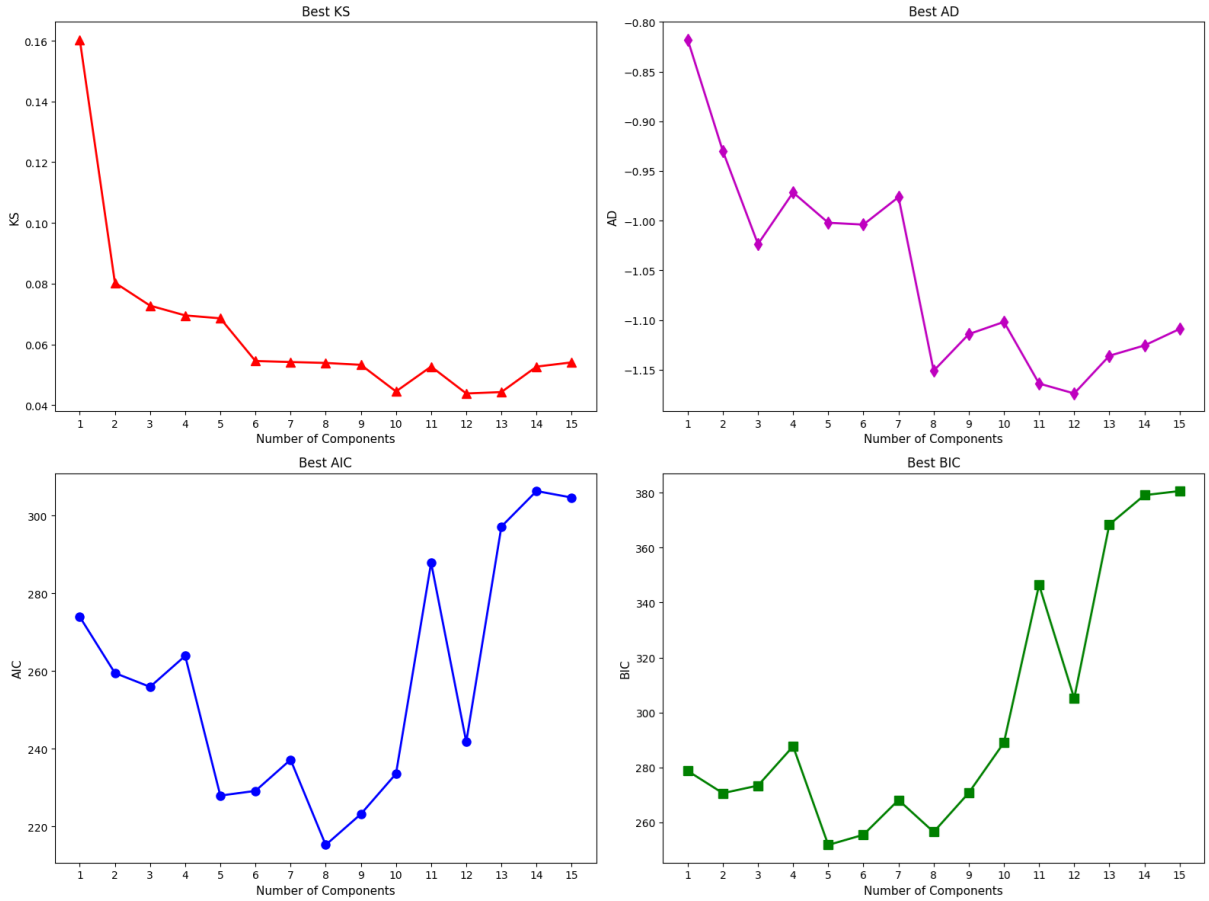

Figure 27 – G2Rac_MC1-074_iexplore_c0000096 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components.

of-fit statistics across component numbers. The KS test sensitivity analysis reveals rapid improvement as component numbers increase from 1 to 6, reflecting the progressive ability to capture distributional nuances. Beyond 6 components, the KS statistics stabilize around 0.05-0.06, suggesting that additional components provide marginal improvements in overall fit quality.

The sensitivity of the AD test provides complementary evidence by emphasizing distribution's tail behavior. The AD statistic shows improvement through moderate component numbers, with the selected 8-component model achieving strong performance. However, irregular patterns emerge at higher complexity levels, indicating potential instability.

Information criteria sensitivity analysis reveals the trade-off inherent in model selection. The AIC trajectory shows improvement through moderate component numbers, with optimal performance around 8 components. Beyond this point, AIC values show irregular patterns, suggesting potential overfitting where additional parameters no longer provide proportional improvements in model likelihood. In contrast, the BIC analysis, which imposes a stronger penalty for model complexity, follows similar trends but with steeper increases at high component numbers, reaching minimum values around 8-9 components before increasing again.

Additionally, graphical analyses of mixture model fits were conducted by inspecting the best-fitting result according to each individual GoF test. The distributions selected as optimal by each test: AD, KS, AIC, and BIC rarely coincided. In particular, the models selected by AIC and BIC consistently exhibited poorer visual and statistical fit compared to those chosen by the AD and KS tests. This performance gap persisted even in the cases where AIC and BIC selected models with a higher number of components.

C.0.0.5 Comparison with Single-Component Models

For comparison, the best-fitting single distribution was the Weibull model, as shown in Table 34. Although it achieved the best performance among the single-component models tested, it still failed to capture the complexity present in the data.

While the Weibull distribution emerged as the best-fitting single-component model among those evaluated, it remains inadequate for representing the complex behavior of time between failures. These models often fail to capture the underlying structural characteristics that arise

Table 34 – Mixture vs. Single Distributions: G2Rac_MC1-074_iexplore_c0000096.

GOF Metric	Mixture Model (8-comp)	Weibull Simple Distribution
KS / p-value	0.055 / 0.9997	0.1393 / 0.4469
AD / p-value	-1.151 / 0.989	-0.239 / 0.4631
Log-Likelihood	-106.94	-153.75
AIC	273.88	254.96
BIC	321.39	259.71

from the inherent complexity of TBFs, which can result from multiple factors, including the diversity of failure causes and variations in workload and operational profiles.

Goodness-of-fit testing highlights this difference in representational adequacy: while the Weibull distribution yields moderate p-values for both the Kolmogorov–Smirnov (KS = 0.1393, p = 0.4469) and Anderson–Darling (AD = -0.239, p = 0.4631) tests, indicating acceptable but not excellent fit. The mixture model achieves much stronger agreement with the data (KS = 0.055, p = 0.9997; AD = -1.151, p = 0.989). The Kolmogorov-Smirnov statistic of 0.1393 for the single distribution is more than twice as large as the mixture model's 0.055, indicating systematic deviations between the empirical data and the fitted model.

The information criteria (AIC and BIC) provide a more nuanced comparison. The Weibull model achieves lower AIC (254.96 vs. 273.88) and BIC (259.71 vs. 321.39) values due to its much lower complexity. However, this advantage in parsimony comes at the cost of substantially poorer distributional fit, as evidenced by the goodness-of-fit test results.

Figure 28 illustrates the limitations of single-distribution modeling for this failure dataset.

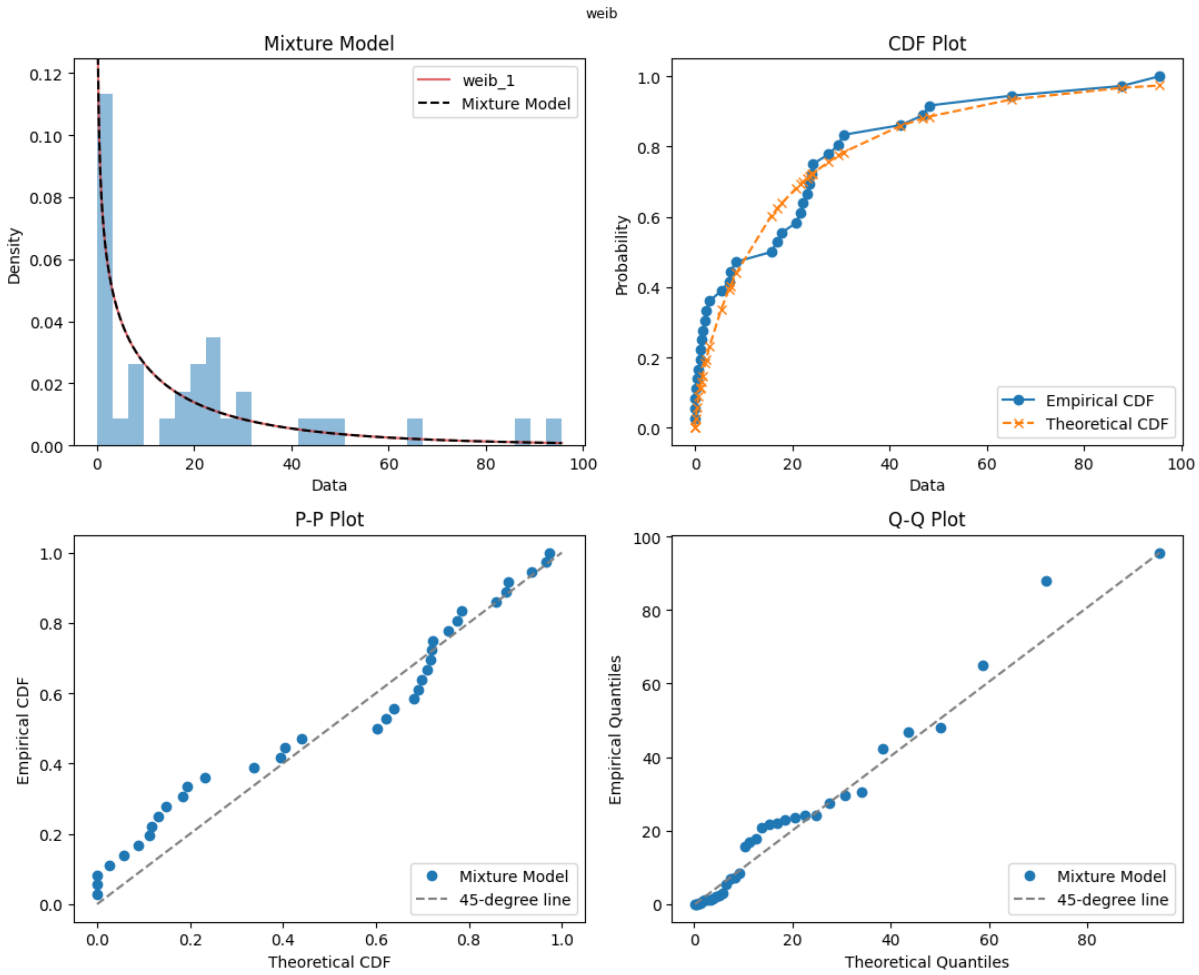

Figure 28 – G2Rac_MC1-074_iexplore_c0000096 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

The PDF overlay reveals that the Weibull model fails to represent the sharp peak near zero and completely misses the multimodal nature evident in the histogram.

The CDF comparison demonstrates significant deviations between the theoretical Weibull distribution (orange crosses) and the empirical distribution (blue circles), particularly in the lower and upper tail regions where the model systematically over or underestimates failure probabilities.

The P-P plot shows departures from linearity, with an S-shaped curve indicating consistent over and under prediction in different regions. The Q-Q plot reveals more nonlinearities, with the data points forming a curved pattern rather than following the expected 45-degree line.

C.0.0.6 Computational Cost Analysis

Given that mixture models can be computationally intensive, particularly when evaluating multiple distribution types and component numbers, a computational cost analysis was conducted. This analysis examines the trade-offs between model complexity and computational requirements.

Figure 29 and 30 present the computational cost analysis, showing execution time, memory usage, CPU utilization, and operation counts across different distribution types and component numbers.

Figure 29 presents the computational cost analysis for homogeneous mixture models across varying numbers of components. The analysis reveals distinct computational characteristics for different distribution families.

Figure 30 extends the analysis to heterogeneous mixture models, where components can belong to different distribution families. This analysis is particularly relevant for the selected 8-component model, which combines lognormal, normal, and gamma distributions.

The computational analysis reveals the scaling behavior of mixture model estimation as a function of both distribution complexity and the number of mixture components. These results inform practical considerations for large-scale reliability analysis applications. While the computational cost is substantial, the significant improvement in model adequacy and the critical importance of accurate failure modeling in reliability applications justify the computational investment.

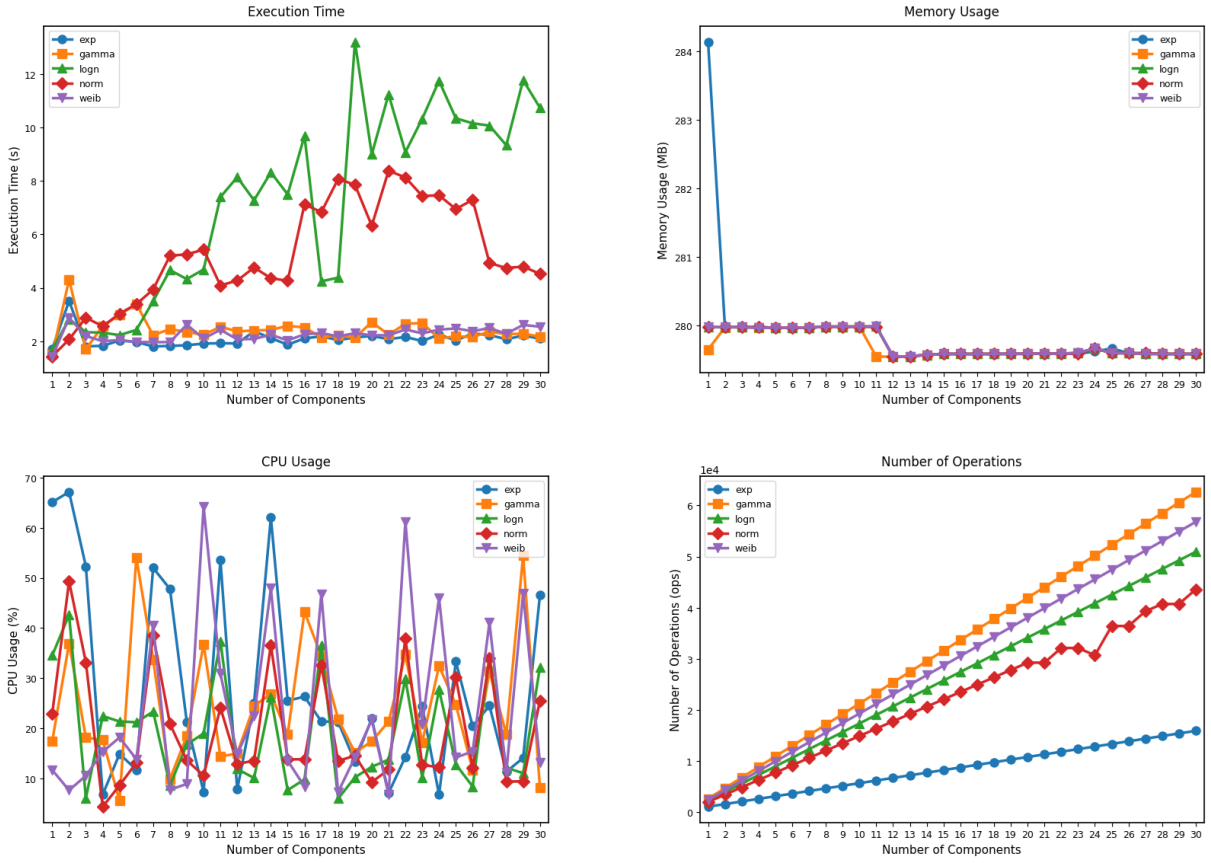

Figure 29 – G2Rac_MC1-074_iexplore_c0000096 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

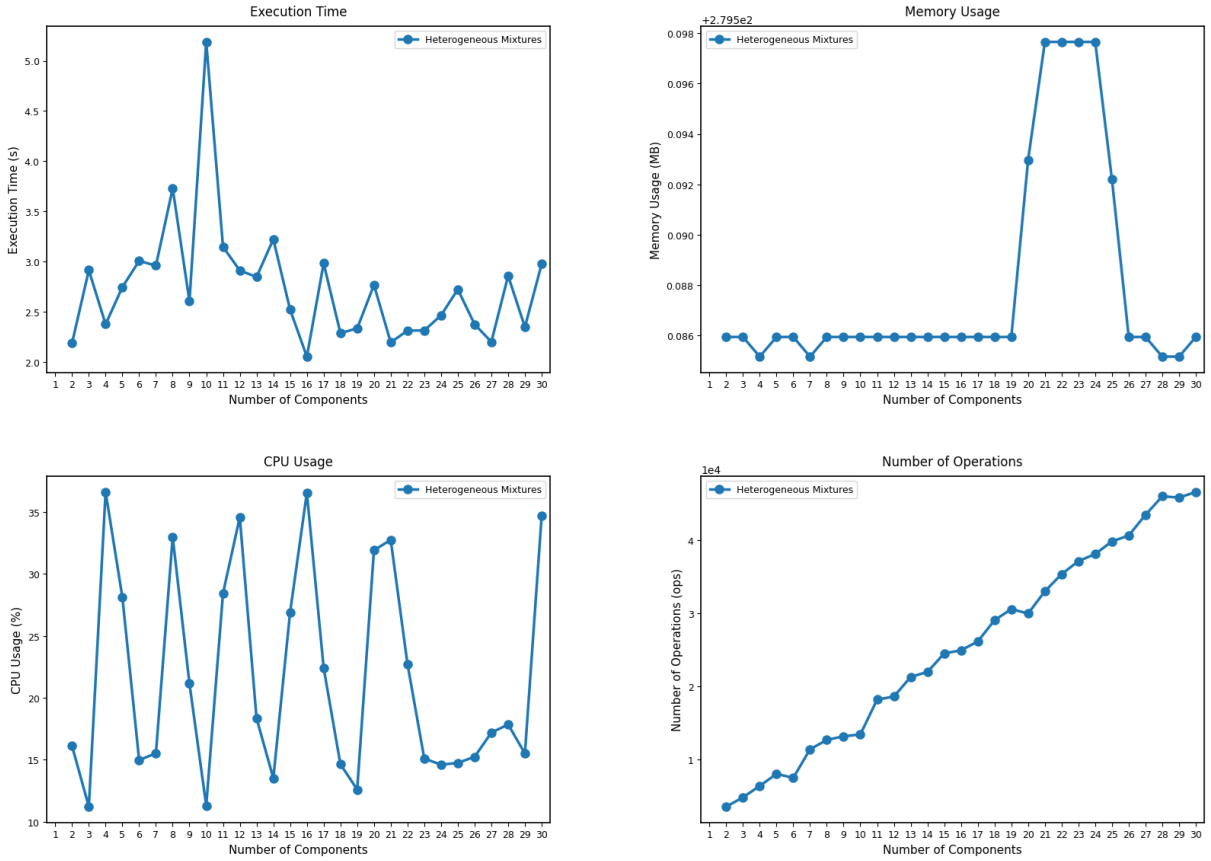

Figure 30 – G2Rac_MC1-074_iexplore_c0000096 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

APPENDIX **D**

Results G2Rac-MC-157400 (Approach 1)

This case refers to failures observed throughout the operational history of the computer MC-157400, which operates within Group 2 (University Administrative Department).

D.0.0.1 Statistical Characterization

Table 35 presents the statistical characteristics of the TBF data for this sample. The descriptive statistics reveal an extremely heterogeneous failure pattern characterized by extreme asymmetry and very heavy-tailed behavior. The difference between mean (35.98 hours) and median (2.82 hours) indicates a strongly right-skewed distribution, which is further confirmed by the extremely high positive skewness value of 9.33.

The extremely low mode value (0.02 hours) combined with the minimum observation (0.001 hours) indicates that some failures occur in rapid succession, possibly representing cascading failures or immediate re-failures after system recovery attempts.

The extremely high kurtosis value (111.71) indicates a distribution with very heavy tails

Table 35 – Descriptive Statistics: G2Rac_MC-157400.

Statistic	**Value**
Count	248
Mean (hours)	35.98
Median (hours)	2.82
Mode (hours)	0.02
Standard Deviation	108.15
Minimum	0.001
Maximum	1434.31
First Quartile (Q1)	0.15
Third Quartile (Q3)	26.00
Interquartile Range (IQR)	25.85
Skewness	9.33
Kurtosis	111.71
Main Data Range	0.01 – 154.67

and an extremely sharp peak, suggesting the coexistence of multiple modes and the presence of extreme outliers. The large interquartile range (25.85 hours) relative to the median further emphasizes the high variability in failure times.

D.0.0.2 Clustering Analysis

To support the selection of the number of mixture components, clustering algorithms were applied to the dataset. Table 36 summarizes the number of clusters recommended by each approach.

The clustering results demonstrate reasonable convergence among most methods, with four of the five algorithms suggesting between 15-18 clusters, indicating substantial granularity in failure pattern recognition. HDBSCAN's recommendation of 16 clusters suggests the presence of numerous density-based groupings. The Gaussian Mixture Model (GMM) approaches, using both BIC and AIC, recommend values of 17 and 25 components, respectively, with AIC suggesting a more complex structure.

Figure 31 provides the visualization of the clustering analysis results across five different algorithms. The scatter plots reveal distinct clustering behaviors: K-Means (k=15) identifies numerous clusters with many observations concentrated in the low TBF region (0-50 hours) and several isolated high-TBF outliers. HDBSCAN (k=16) shows similar partitioning with slightly different cluster boundaries.

Fuzzy C-Means (k=18) produces a clustering pattern with finer granularity, while the GMM approaches (BIC with k=17 and AIC with k=25) show progressive increase in partitioning complexity. The Elbow and Silhouette analysis in the bottom-right panel demonstrates the trade-off between cluster quality metrics, with the Elbow method showing improvement through moderate component numbers, while the Silhouette score peaks at lower cluster numbers.

D.0.0.3 Mixture Model Selection and Validation

The selection of the best-fitting mixture model was based on the top five models identified by each GoF test, combined with a visual comparison of the graphical results for each candidate model. This procedure was applied across all samples. The final model selected via the Expectation-Maximization (EM) algorithm consisted of 21 components with the following dis-

Table 36 – Cluster Results: G2Rac_MC-157400.

Clustering Approach	Recommended Clusters
K-Means	15
HDBSCAN	16
Fuzzy C-means	18
GMM (BIC)	17
GMM (AIC)	25

tributional structure: logn-logn-logn-norm-norm-weib-logn-norm-weib-gamma-norm-norm-norm-norm-norm-norm-gamma-norm-norm-norm-norm. This configuration was selected based on the outcomes of the KS goodness-of-fit test.

Figure 32 demonstrates the fit achieved by the 21-component mixture model. The probability density function plot shows a highly multimodal distribution with a dominant peak near zero and several smaller peaks at intermediate values (50-300 hours) and longer intervals (500-1400 hours).

The cumulative distribution function comparison reveals agreement between the empiri-

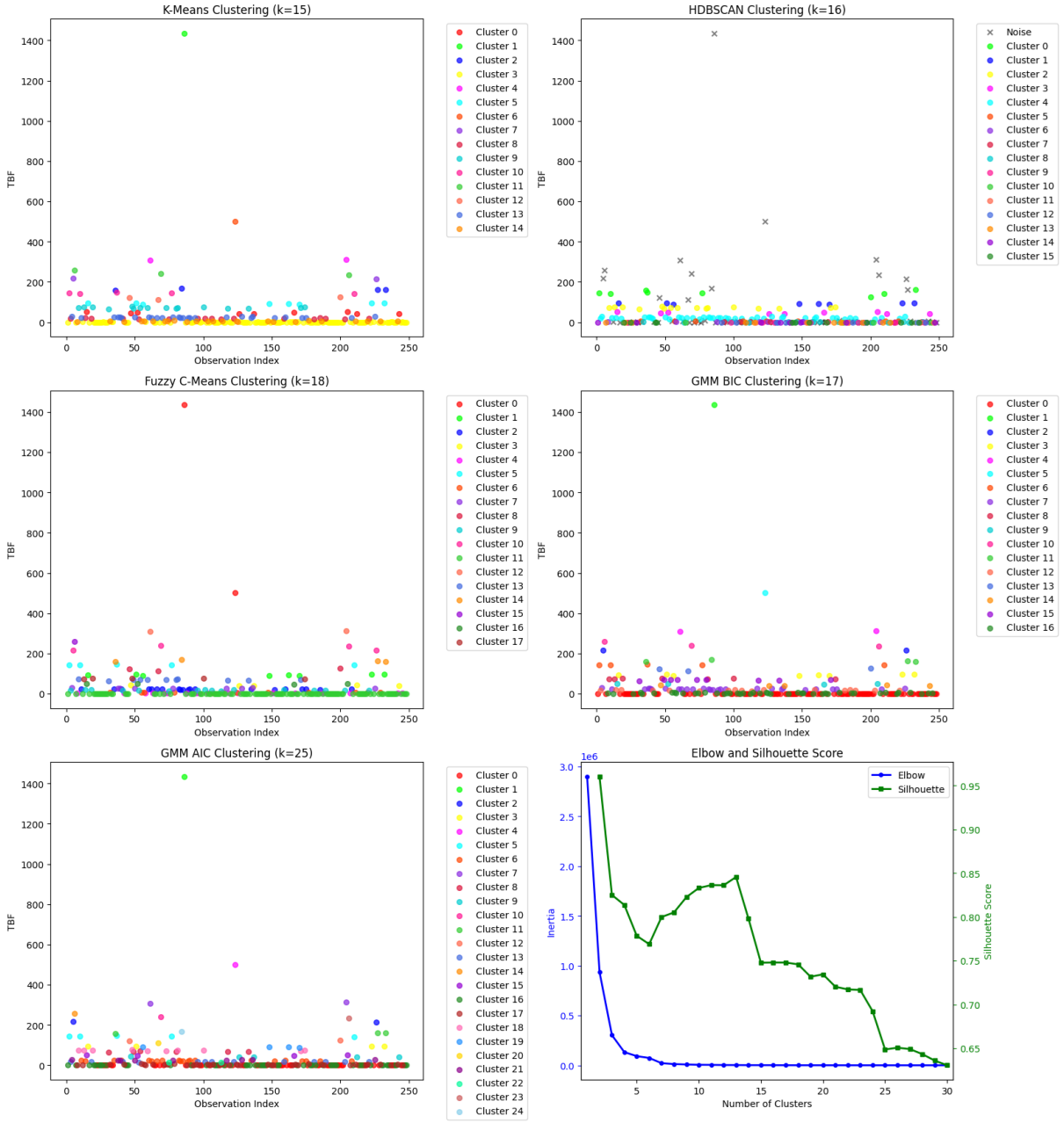

Figure 31 – Cluster Evaluation Plots for the Sample G2Rac_MC-157400.

cal and theoretical distributions, with the theoretical CDF (orange crosses) closely overlaying the empirical CDF (blue circles) across the entire range. The probability-probability (P-P) plot shows linearity along the 45-degree line, indicating that the mixture model accurately represents the cumulative probability structure at all quantile levels.

The quantile-quantile plot further confirms model adequacy, with linear correspondence between theoretical and empirical quantiles. Minor deviations are observed only at the extreme upper tail (around 1400 hours). The overall linear pattern validates the mixture model's ability to capture both the central tendency and the extreme behavior of the failure process.

Table 37 summarizes the goodness-of-fit results for the selected model. The GoF test results provide statistical evidence for the adequacy of the 21-component mixture model. The KS test yields a low test statistic (0.024) with a high p-value (0.998), indicating that the null hypothesis of distributional equivalence cannot be rejected with confidence. The AD test, which is more sensitive to deviations in the distribution tails, produces a test statistic of -1.033 with a p-value of 0.984. This result further confirms the model adequacy.

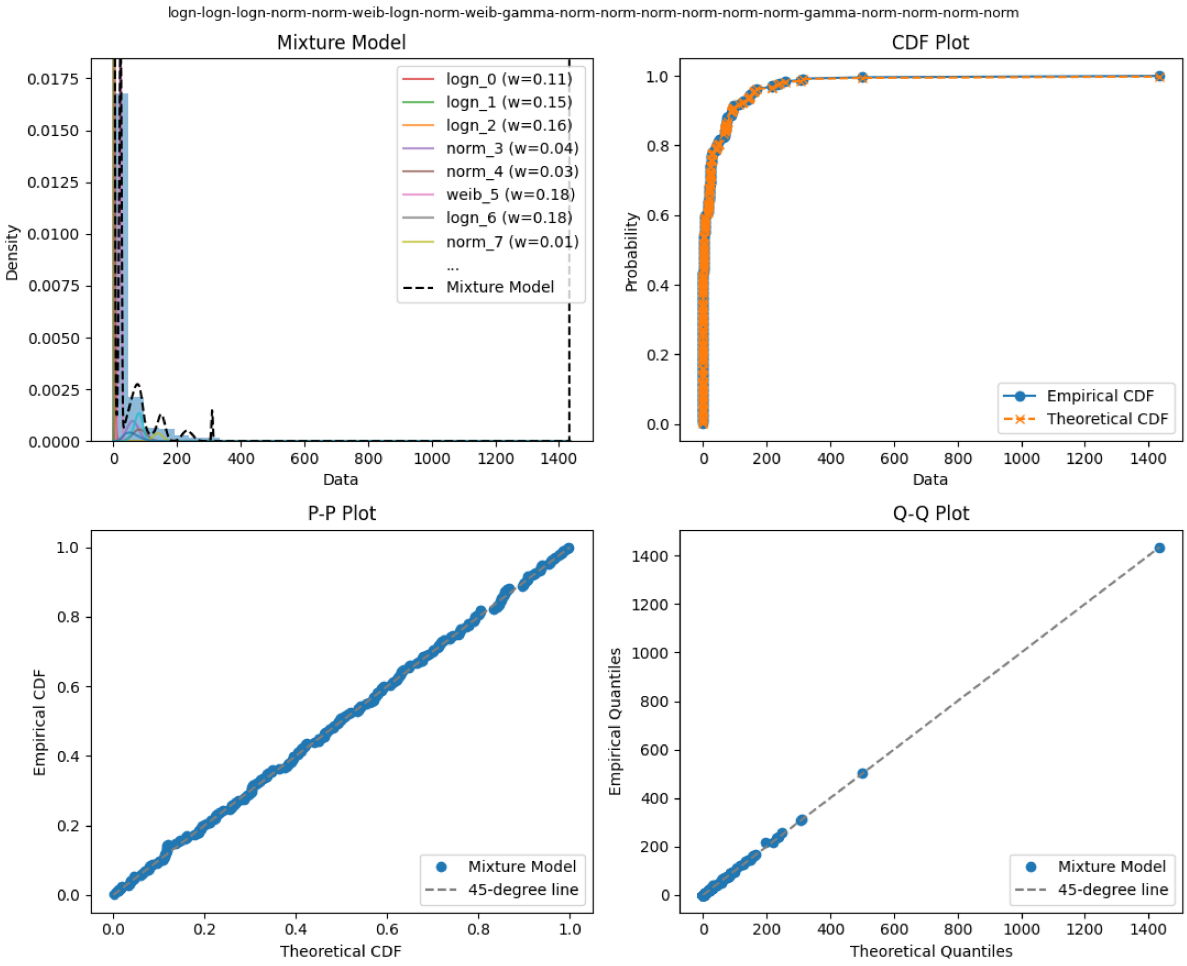

Figure 32 – G2Rac_MC-157400 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

The log-likelihood value of -671.25 reflects the model's overall fit and provides a basis for comparison with alternative models. In addition, the AIC and the BIC values are 1482.50 and 1728.44, respectively, these criteria penalize model complexity.

Having established the adequacy of the selected mixture model in terms of global fit, the next step is to examine the estimated parameters associated with each mixture component. Table 38 presents the estimated parameters of the fitted mixture model.

Each distribution follows its standard statistical parameterization, respectively:

- ❏ Normal distribution uses the location (μ) and scale (σ) parameters.
- ❏ Lognormal distribution is parameterized by the shape (σ), location (μ), and scale (θ) parameters.

Table 37 – Goodness-of-fit Test Results: G2Rac_MC-157400.

Model	**Test**	**Statistic / p-value**
21-Component Mixture	KS	0.024 / 0.998
	AD	-1.033 / 0.984
	Log-Likelihood	-671.25
	AIC	1482.50
	BIC	1728.44

Table 38 – Mixture Model Parameters: G2Rac_MC-157400.

Component	**Distribution**	**Weight**	**Param 1**	**Param 2**	**Param 3**
logn_0	Lognormal	0.114	0.380	1E-10	4.201
logn_1	Lognormal	0.148	0.798	1E-10	0.565
logn_2	Lognormal	0.156	1.734	1E-10	0.118
norm_3	Normal	0.045	60.917	18.214	—
norm_4	Normal	0.030	82.371	21.177	—
weib_5	Weibull	0.177	6.776	1E-10	23.527
logn_6	Lognormal	0.183	1.477	1E-10	0.022
norm_7	Normal	0.014	140.194	15.178	—
weib_8	Weibull	0.045	6.705	1E-10	82.976
gamma_9	Gamma	0.024	5.977	1E-10	10.103
norm_10	Normal	0.027	152.436	10.944	—
norm_11	Normal	0.0004	131.516	43.157	—
norm_12	Normal	0.0006	151.810	52.232	—
norm_13	Normal	0.0023	234.928	16.060	—
norm_14	Normal	0.0036	237.519	15.692	—
norm_15	Normal	0.0032	240.280	15.330	—
gamma_16	Gamma	0.0098	284.151	1E-10	0.803
norm_17	Normal	0.0011	242.589	14.748	—
norm_18	Normal	0.0081	310.624	2.095	—
norm_19	Normal	0.0040	500.769	1E-10	—
norm_20	Normal	0.0040	1434.308	1E-10	—

- ❏ Weibull distribution is characterized by the shape (k), location (μ), and scale (λ) parameters.
- ❏ Gamma distribution is characterized by the shape (α), location (μ), and scale (θ) parameters.

The mixture weights indicate the relative importance of each failure subpopulation, with several dominant components: logn_6 (18.3%), weib_5 (17.7%), and logn_2 (15.6%). These high-weight components correspond to different failure regimes, from rapid succession failures (logn_6 with scale parameter 0.022) to intermediate stability periods (weib_5 with scale parameter 23.527).

The lognormal components account for 4 out of 21 components (60.1% of total weight), reflecting the multiplicative nature of the failure process. The presence of twelve normal components with location parameters ranging from 60.917 to 1434.308 hours captures extended TBFs across multiple operational regimes. The Weibull components (weib_5, weib_8) with shape parameters around 6.7-6.8 suggest wear-out behavior, while gamma components (gamma_9, gamma_16) provide additional flexibility for modeling intermediate failure patterns. Several components exhibit extremely small weights (< 0.005) but occupy important positions in the parameter space, particularly in the extreme tail regions (norm_19, norm_20), essential for capturing the extreme kurtosis observed in the data.

D.0.0.4 Sensitivity Analysis

To assess the robustness of the selected 21-component model and evaluate how model performance responds to changes in complexity, a sensitivity analysis was performed. This analysis investigated the behavior of GoF metrics as the number of mixture components increased. The sensitivity analysis was conducted by plotting the best GoF statistics obtained for each number of mixture components.

Figure 33 presents the sensitivity analysis results, examining the behavior of key goodness-of-fit statistics across component numbers. The KS test sensitivity analysis reveals rapid improvement as component numbers increase from 1 to approximately 10-15 components. Beyond this range, the KS statistic stabilizes around 0.024-0.027 for models containing 15-25 components, indicating that additional components provide marginal improvements in overall distributional fit.

The sensitivity of the AD test provides complementary evidence by emphasizing distribution's tail behavior. The AD statistic shows similar patterns of improvement through moderate component numbers, with the selected 21-component model achieving strong performance in representing tail behavior.

Information criteria sensitivity analysis reveals the trade-off inherent in model selection. The AIC trajectory shows initial rapid decrease followed by gradual increase as model complexity penalties begin to outweigh likelihood improvements. The minimum AIC occurs around

10-16 components. In contrast, the BIC analysis, which imposes a stronger penalty for model complexity, reaches its minimum at lower component numbers, reflecting its preference for parsimony.

Additionally, graphical analyses of mixture model fits were conducted by inspecting the best-fitting result according to each individual GoF test. The distributions selected as optimal by each test: AD, KS, AIC, and BIC rarely coincided. In particular, the models selected by AIC and BIC consistently exhibited poorer visual and statistical fit compared to those chosen by the AD and KS tests. This performance gap persisted even in the cases where AIC and BIC selected models with a higher number of components.

D.0.0.5 Comparison with Single-Component Models

For comparison, the best-fitting single distribution was the lognormal model, as shown in Table 39. Although it achieved the best performance among the single-component models tested, it still failed to capture the complexity present in the data.

While the lognormal distribution emerged as the best-fitting single-component model

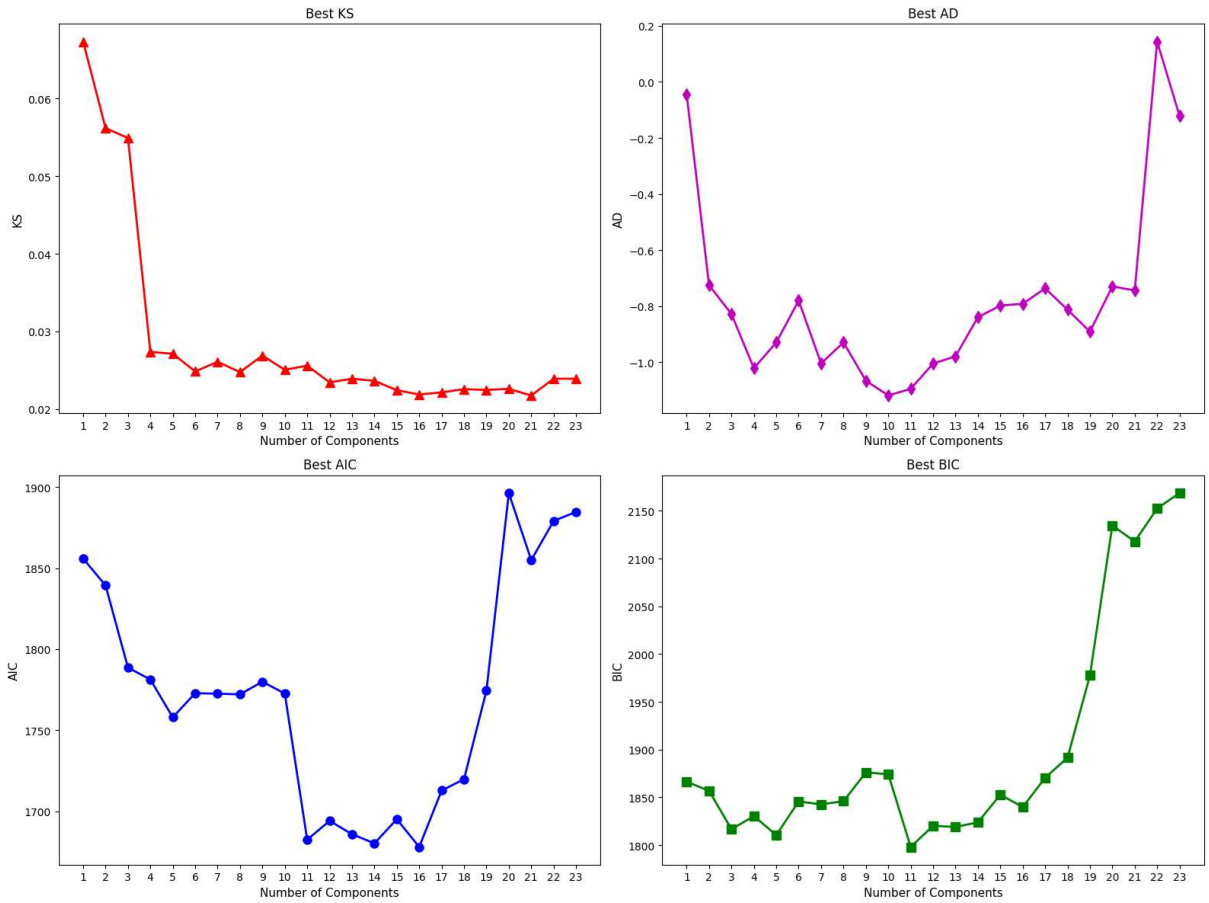

Figure 33 – G2Rac_MC-157400 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components.

among those evaluated, it remains inadequate for representing the complex behavior of time between failures. These models often fail to capture the underlying structural characteristics that arise from the inherent complexity of TBFs, which can result from multiple factors, including the diversity of failure causes and variations in workload and operational profiles.

Goodness-of-fit testing highlights this difference in representational adequacy: while the lognormal distribution yields low p-values for both the Kolmogorov–Smirnov (KS = 0.135, p = 0.0002) and Anderson–Darling (AD = 4.190, p = 0.010) tests, indicating clear rejection of the distributional hypothesis. The mixture model achieves much stronger agreement with the data (KS = 0.024, p = 0.998; AD = -1.033, p = 0.984). The Kolmogorov-Smirnov statistic of 0.135 for the single distribution is more than five times larger than the mixture model's 0.024, indicating systematic and substantial deviations between the empirical data and the fitted model.

The information criteria (AIC and BIC) provide a more nuanced comparison. While the AIC strongly favors the mixture model (1482.50 vs. 1602.05), representing an improvement of 119.55 AIC units, the BIC comparison shows the lognormal model achieving slightly better performance (1612.59 vs. 1728.44) due to stronger complexity penalties. However, this BIC advantage for the simple model is completely overshadowed by the differences in statistical fit quality, and becomes irrelevant when the simpler model fails basic goodness-of-fit requirements.

Figure 34 illustrates the limitations of single-distribution modeling for this failure dataset. The PDF overlay reveals that the lognormal model fails to represent the sharp peak near zero and completely misses the multimodal nature evident in the histogram.

The CDF comparison demonstrates significant deviations between the theoretical lognormal distribution (orange crosses) and the empirical distribution (blue circles), particularly in the lower tail region (0-100 hours) where the model underestimates failure probabilities, and in the upper tail where it overestimates them.

The P-P plot shows departures from linearity, with an S-shaped curve. The Q-Q plot reveals more nonlinearities, with the data points forming a curved pattern rather than following the expected 45-degree line.

Table 39 – Mixture vs. Single Distributions: G2Rac_MC-157400.

GOF Metric	**Mixture Model (21-comp)**	**Lognormal Simple Distribution**
KS / p-value	0.024 / 0.998	0.135 / 0.0002
AD / p-value	-1.033 / 0.984	4.190 / 0.010
Log-Likelihood	-671.25	-782.86
AIC	1482.50	1602.05
BIC	1728.44	1612.59

D.0.0.6 Computational Cost Analysis

Given that mixture models can be computationally intensive, particularly when evaluating multiple distribution types and component numbers, a computational cost analysis was conducted. This analysis examines the trade-offs between model complexity and computational requirements.

Figure 35 and 36 present the computational cost analysis, showing execution time, memory usage, CPU utilization, and operation counts across different distribution types and component numbers.

Figure 35 presents the computational cost analysis for homogeneous mixture models across varying numbers of components. The analysis reveals distinct computational characteristics for different distribution families.

Figure 36 extends the analysis to heterogeneous mixture models, where components can belong to different distribution families. This analysis is particularly relevant for the selected 21-component model, which combines lognormal, normal, Weibull, and gamma distributions.

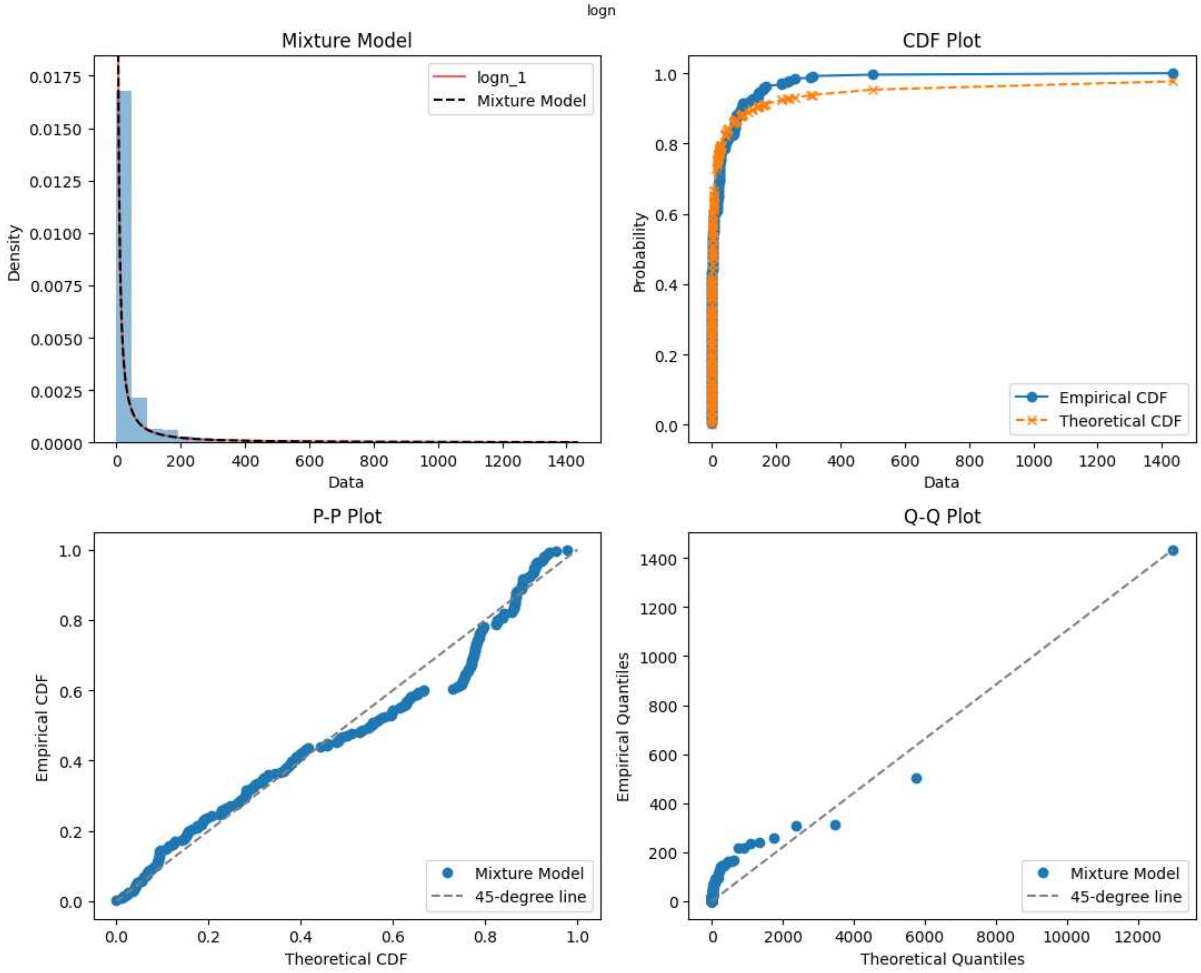

Figure 34 – G2Rac_MC-157400 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

The computational analysis reveals the scaling behavior of mixture model estimation as a function of both distribution complexity and the number of mixture components. These results inform practical considerations for large-scale reliability analysis applications. While the computational cost is substantial, the significant improvement in model adequacy and the critical importance of accurate failure modeling in reliability applications justify the computational investment.

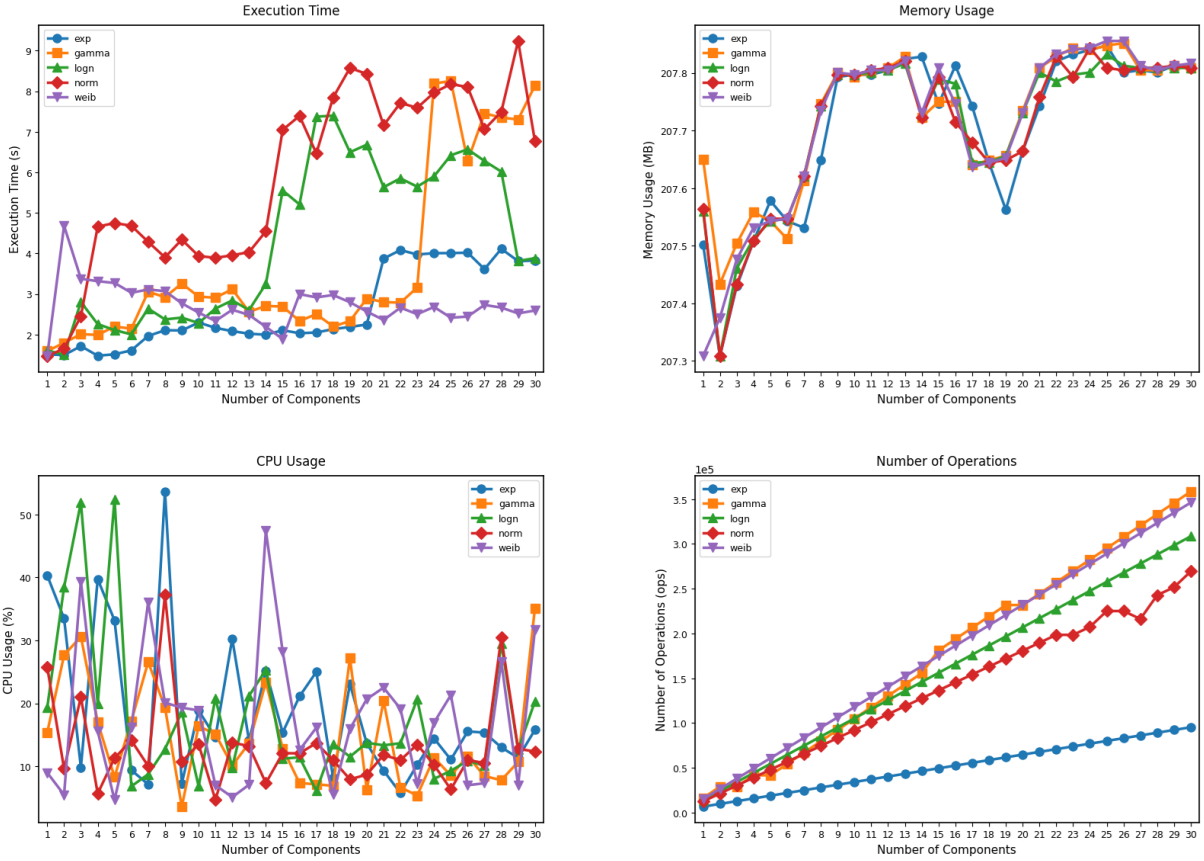

Figure 35 – G2Rac_MC-157400 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

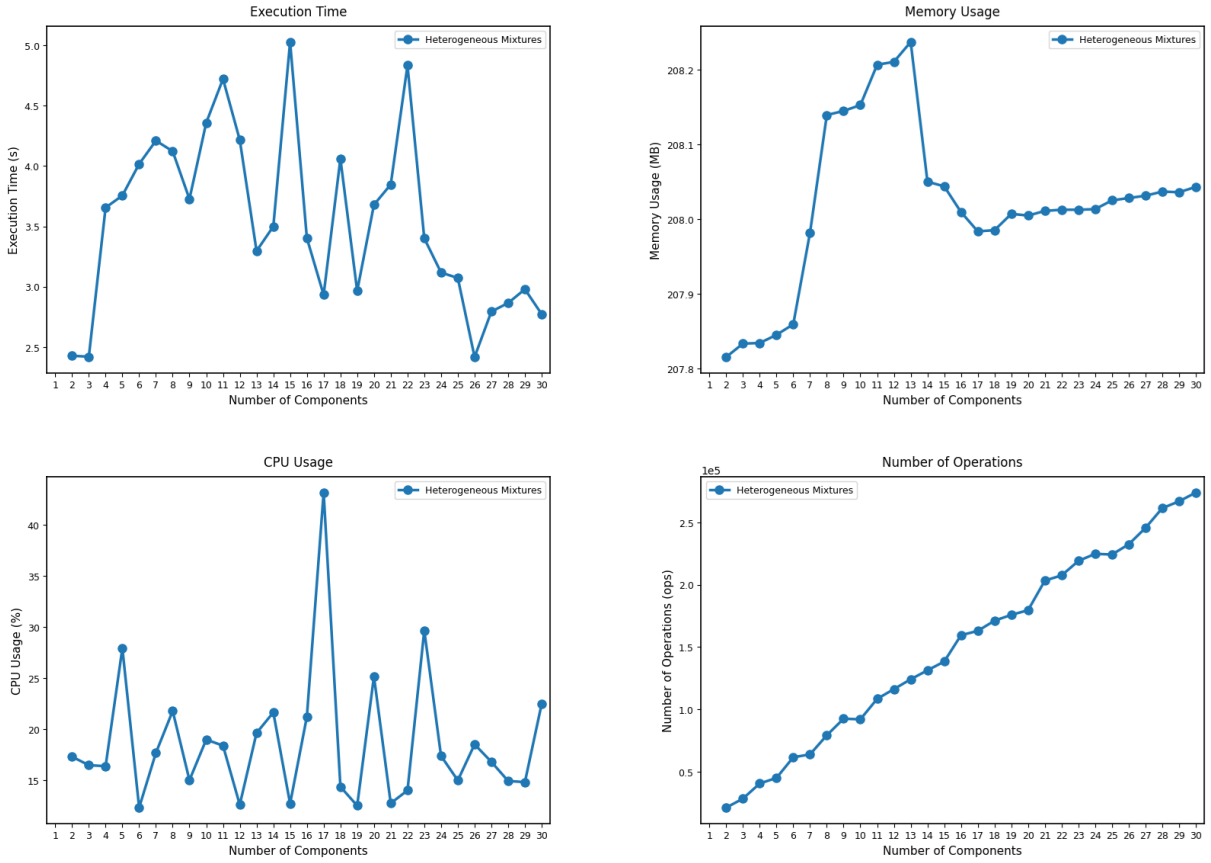

Figure 36 – G2Rac_MC-157400 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

APPENDIX **E**

Results G2Rac-MC-157400-iexplore-c0000374 (Approach 2)

This case refers to failures related to unhandled C++ exception (c0000374), occurring in the `iexplore.exe`'s process on computer MC-157400, part of the Group 2 (Graduate Laboratory environment).

E.0.0.1 Statistical Characterization

Table 40 presents the statistical characteristics of the TBF data for this sample. The descriptive statistics reveal a highly heterogeneous failure pattern characterized by extreme asymmetry and very heavy-tailed behavior. The difference between mean (136.94 hours) and median (2.53 hours) indicates a strongly right-skewed distribution, which is further confirmed by the high positive skewness value of 5.33.

The extremely low minimum value (0.006 hours) indicates that some failures occur in rapid succession, possibly representing cascading failures or immediate re-failures after system recovery attempts.

The extremely high kurtosis value (28.21) indicates a distribution with very heavy tails and a sharp peak, suggesting the coexistence of multiple modes and the presence of extreme outliers. The large interquartile range (70.31 hours) relative to the median further emphasizes the high variability in failure times.

E.0.0.2 Clustering Analysis

To support the selection of the number of mixture components, clustering algorithms were applied to the dataset. Table 41 summarizes the number of clusters recommended by each approach.

K-Means and Fuzzy C-means algorithms demonstrate convergence, both suggesting 9 clusters, indicating moderate granularity in failure pattern recognition. HDBSCAN failed to identify meaningful cluster structure, likely attributable to the algorithm's sensitivity to density variations and its inability to handle the extreme outliers effectively. The Gaussian Mixture Model (GMM) approaches, using both BIC and AIC, recommend 10 components.

Figure 37 provides the visualization of the clustering analysis results across five different algorithms. The scatter plots reveal distinct clustering behaviors: K-Means (k=9) identifies multiple clusters with reasonably clear separation patterns. HDBSCAN shows no meaningful clustering pattern, with most data points unclassified, highlighting the limitations of density-based clustering methods when applied to datasets with extreme scale variations.

Fuzzy C-Means (k=9) produces a clustering pattern consistent with K-Means results, while the GMM approaches (BIC with k=10 and AIC with k=10) show slight disagreement with the hard clustering methods but remain within reasonable proximity. The Elbow and Silhouette analysis in the bottom-right panel demonstrates the trade-off between cluster quality metrics, with both methods indicating optimal performance at moderate cluster numbers.

Table 40 – Descriptive Statistics: G2Rac_MC-157400_iexplore_c0000374.

Statistic	**Value**
Count	38
Mean (hours)	136.94
Median (hours)	2.53
Mode (hours)	0.006
Standard Deviation	475.22
Minimum	0.006
Maximum	2924.98
First Quartile (Q1)	0.164
Third Quartile (Q3)	70.47
Interquartile Range (IQR)	70.31
Skewness	5.33
Kurtosis	28.21
Main Data Range	0.006 – 2924.98

Table 41 – Cluster Results: G2Rac_MC-157400_iexplore_c0000374.

Clustering Approach	**Recommended Clusters**
K-Means	9
HDBSCAN	N/A
Fuzzy C-means	9
GMM (BIC)	10
GMM (AIC)	10

E.0.0.3 Mixture Model Selection and Validation

The selection of the best-fitting mixture model was based on the top five models identified by each GoF test, combined with a visual comparison of the graphical results for each candidate model. This procedure was applied across all samples. The final model selected via the Expectation-Maximization (EM) algorithm consisted of 5 components with the following distributional structure: logn-logn-logn-logn-weib. This configuration was selected based on the outcomes of the KS goodness-of-fit test.

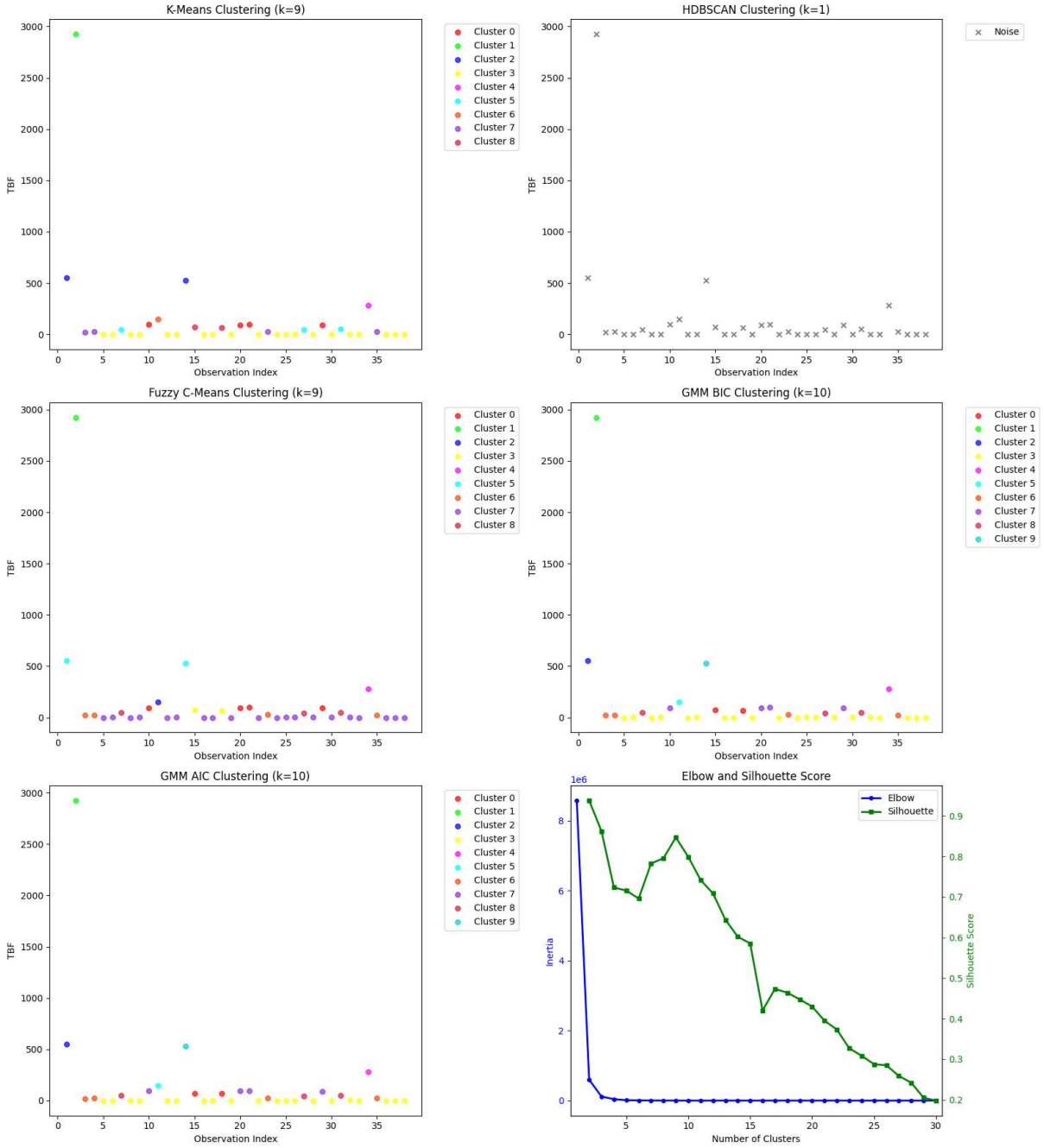

Figure 37 – Cluster Evaluation Plots: G2Rac_MC-157400_iexplore_c0000374.

Figure 38 demonstrates the fit achieved by the 5-component mixture model. The probability density function plot shows a multimodal distribution with a dominant peak near zero and several smaller peaks at intermediate and extreme values (500-3000 hours).

The cumulative distribution function comparison reveals agreement between the empirical and theoretical distributions, with the theoretical CDF (orange crosses) closely overlaying the empirical CDF (blue circles) across the entire range. The probability-probability (P-P) plot shows linearity along the 45-degree line, indicating that the mixture model accurately represents the cumulative probability structure at all quantile levels.

The quantile-quantile plot further confirms model adequacy, with linear correspondence between theoretical and empirical quantiles. Minor deviations are observed only at the extreme tail regions. The overall linear pattern validates the mixture model's ability to capture both the central tendency and the extreme behavior of the failure process.

Table 42 summarizes the goodness-of-fit results for the selected model. The GoF test results provide statistical evidence for the adequacy of the 5-component mixture model. The KS test

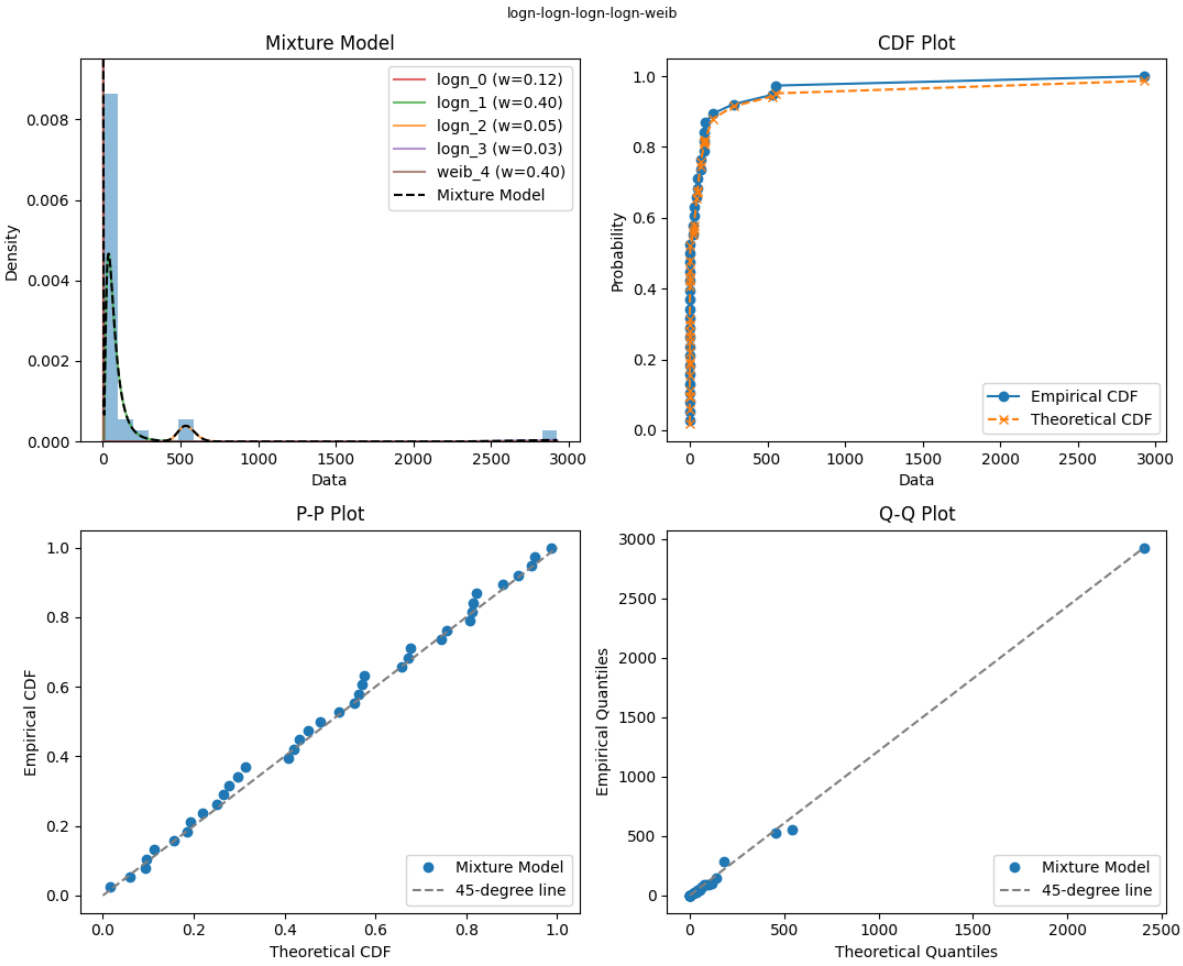

Figure 38 – G2Rac_MC-157400_iexplore_c0000374 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

yields a low test statistic (0.055) with an extremely high p-value (0.999), indicating that the null hypothesis of distributional equivalence cannot be rejected with confidence. The AD test, which is more sensitive to deviations in the distribution tails, produces a test statistic of -0.757 with a p-value of 0.798. This result further confirms the model adequacy.

The AIC and the BIC values are 292.23 and 323.35, respectively, these criteria penalize model complexity. The relatively modest difference between these information criteria suggests that the five-component model achieves good fit without excessive parameterization.

Having established the adequacy of the selected mixture model in terms of global fit, the next step is to examine the estimated parameters associated with each mixture component. Table 43 presents the estimated parameters of the fitted mixture model.

Each distribution follows its standard statistical parameterization, respectively:

- ❏ Lognormal distribution is parameterized by the shape (σ), location (μ), and scale (θ) parameters.
- ❏ Weibull distribution is characterized by the shape (k), location (μ), and scale (λ) parameters.

The mixture weights indicate the relative importance of each failure subpopulation, with two dominant components: weib_4 (40.2%) and logn_1 (39.5%). These high-weight components correspond to different failure regimes. Together, these two components represent nearly 80% of the data, indicating that the majority of observations arise from these two primary modes.

The lognormal components dominate the mixture (4 out of 5 components, accounting for 59.7% of total weight), reflecting the multiplicative nature of the failure process. The Weibull

Table 42 – Goodness-of-fit Test Results: G2Rac_MC-157400_iexplore_c0000374.

Model	**Test**	**Statistic / p-value**
5-Component Mixture	KS	0.055 / 0.999
	AD	-0.757 / 0.798
	Log-Likelihood	-127.11
	AIC	292.23
	BIC	323.35

Table 43 – Mixture Model Parameters: G2Rac_MC-157400_iexplore_c0000374.

Component	**Distribution**	**Weight**	**Param 1**	**Param 2**	**Param 3**
logn_0	Lognormal	0.124	0.348	1E-10	0.011
logn_1	Lognormal	0.395	0.711	1E-10	61.23
logn_2	Lognormal	0.052	0.100	1E-10	539.42
logn_3	Lognormal	0.026	0.100	1E-10	2924.98
weib_4	Weibull	0.402	0.771	1E-10	0.548

component with its substantial weight of 40.2% appears to model the lower-value observations with its characteristic decreasing hazard rate (shape parameter 0.771). The dominant lognormal component (logn_1 with 39.5% weight) captures the main body of intermediate values with scale parameter 61.23.

E.0.0.4 Sensitivity Analysis

To assess the robustness of the selected 5-component model and evaluate how model performance responds to changes in complexity, a sensitivity analysis was performed. This analysis investigated the behavior of GoF metrics as the number of mixture components increased. The sensitivity analysis was conducted by plotting the best GoF statistics obtained for each number of mixture components.

Figure 39 presents the sensitivity analysis results, examining the behavior of key goodness-of-fit statistics across component numbers. The KS test sensitivity analysis reveals that the statistic achieves its first minimum value at five components, with model performance improvements becoming marginal beyond this point.

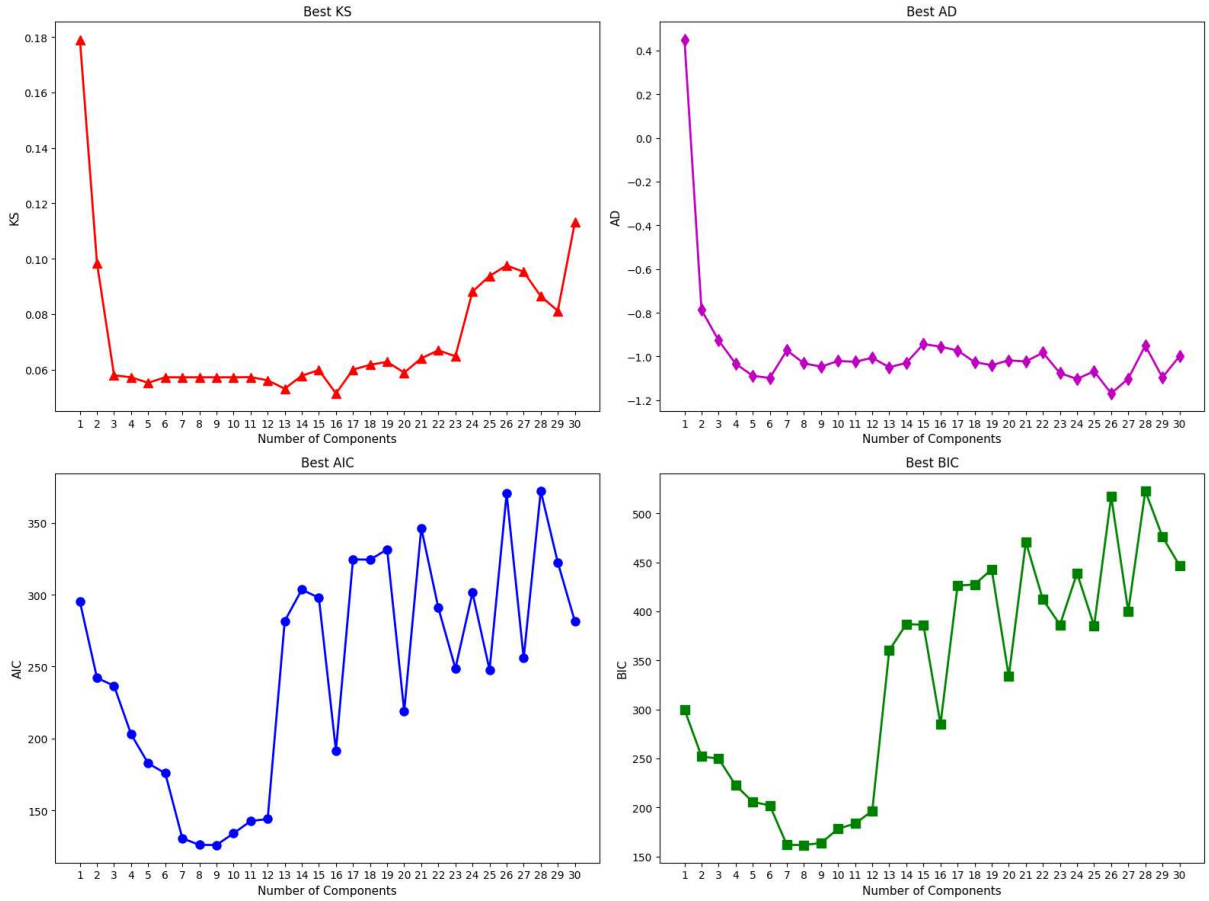

Figure 39 – G2Rac_MC-157400_iexplore_c0000374 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components.

The sensitivity of the AD test provides complementary evidence by emphasizing distribution's tail behavior. The AD statistic shows improvement through moderate component numbers, with the selected 5-component model achieving strong performance in representing tail behavior.

Information criteria sensitivity analysis reveals the trade-off inherent in model selection. The AIC trajectory reaches its minimum value at nine components, with subsequent increases indicating that additional components introduce unnecessary complexity without proportional improvement in model fit. In contrast, the BIC analysis, which imposes a stronger penalty for model complexity, achieves its optimum at 7 components, reflecting its preference for parsimony.

Additionally, graphical analyses of mixture model fits were conducted by inspecting the best-fitting result according to each individual GoF test. The distributions selected as optimal by each test: AD, KS, AIC, and BIC rarely coincided. In particular, the models selected by AIC and BIC consistently exhibited poorer visual and statistical fit compared to those chosen by the AD and KS tests. This performance gap persisted even in the cases where AIC and BIC selected models with a higher number of components.

E.0.0.5 Comparison with Single-Component Models

For comparison purposes, the lognormal distribution provided the best fit among the single-component models evaluated, as shown in Table 44. Given the strictly positive support and right-skewed nature of the TBF data, the lognormal model represents a natural baseline choice. Importantly, this single-component model is not rejected by goodness-of-fit testing and shows acceptable agreement with the empirical data.

Goodness-of-fit testing confirms this observation. The lognormal distribution yields moderate p-values for both the Kolmogorov–Smirnov (KS = 0.1603, p = 0.2545) and Anderson–Darling (AD = 0.192, p = 0.2914) tests, indicating that it cannot be statistically rejected at conventional significance levels. Visual inspection of the fitted curves further supports this conclusion, as the single model captures the overall trend of the empirical distribution.

However, a direct comparison with the mixture model reveals substantial differences in representational quality. The mixture model achieves considerably lower KS and AD statistics (KS = 0.055; AD = -0.757), along with markedly higher p-values (0.999 and 0.798, respectively),

Table 44 – Mixture vs. Single Distributions: G2Rac_MC-157400_iexplore_c0000374.

GOF Metric	Mixture Model (5-comp)	Lognormal Simple Distribution
KS / p-value	0.055 / 0.999	0.1603 / 0.2545
AD / p-value	-0.757 / 0.798	0.192 / 0.2914
Log-Likelihood	-127.11	-198.53
AIC	292.23	280.75
BIC	323.35	285.66

indicating a much closer agreement with the empirical distribution. In particular, the KS statistic for the single lognormal model is nearly three times larger than that of the mixture model, suggesting systematic deviations that are mitigated by the multi-component representation.

From a model selection perspective, the information criteria favor the simpler lognormal model, which attains lower AIC (280.75 vs. 292.23) and BIC (285.66 vs. 323.35) values due to its reduced complexity. Nevertheless, this advantage in parsimony comes at the expense of diminished descriptive accuracy. In this context, the substantially improved goodness-of-fit achieved by the mixture model provides a strong justification for the additional complexity, particularly given the heterogeneous nature of software failure processes.

Figure 40 further illustrates this trade-off. While the lognormal model provides a reasonable approximation of the overall distribution, the PDF reveals limitations in representing the sharp peak near zero and fails to capture the multimodal features present in the empirical histogram. Deviations are also observed in the CDF, particularly in the lower and upper tail regions.

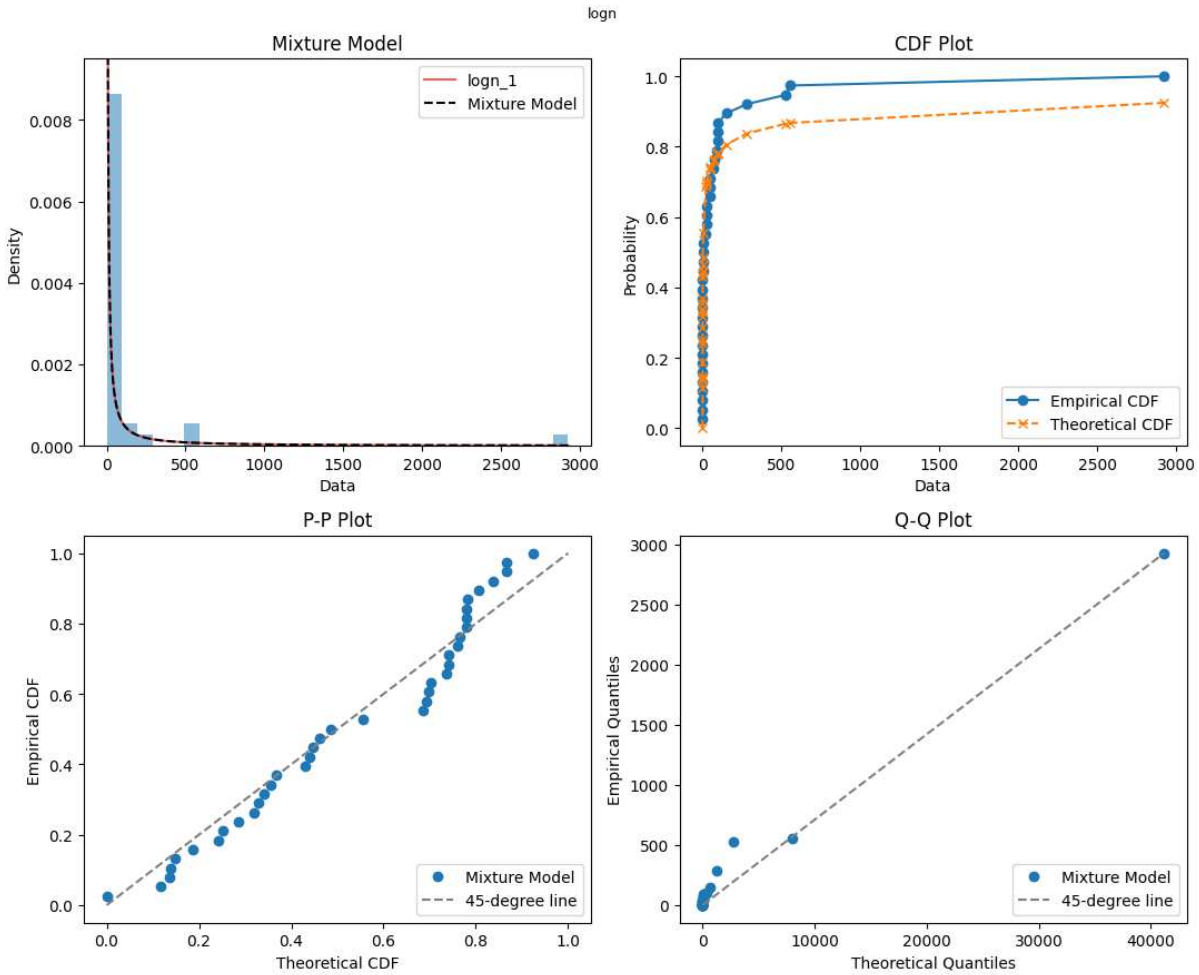

Figure 40 – G2Rac_MC-157400_iexplore_c0000374 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

The P–P and Q–Q plots reinforce this interpretation. Although the points broadly follow the expected trend, systematic departures from linearity, indicating residual structure in the data that is not fully explained by a single-component model.

E.0.0.6 Computational Cost Analysis

Given that mixture models can be computationally intensive, particularly when evaluating multiple distribution types and component numbers, a computational cost analysis was conducted. This analysis examines the trade-offs between model complexity and computational requirements.

Figure 41 and 42 present the computational cost analysis, showing execution time, memory usage, CPU utilization, and operation counts across different distribution types and component numbers.

Figure 41 presents the computational cost analysis for homogeneous mixture models across varying numbers of components. The analysis reveals distinct computational characteristics for different distribution families.

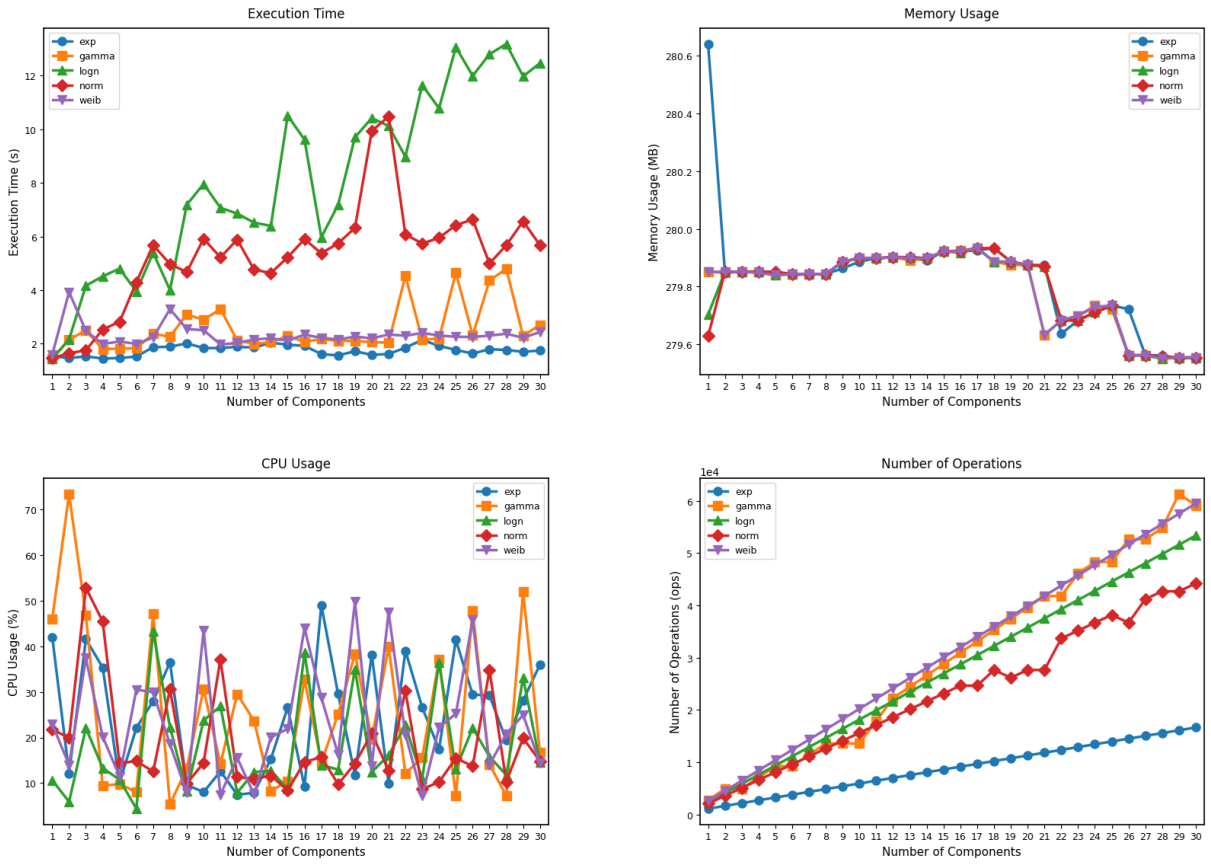

Figure 41 – G2Rac_MC-157400_iexplore_c0000374 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

Figure 42 extends the analysis to heterogeneous mixture models, where components can belong to different distribution families. This analysis is particularly relevant for the selected 5-component model, which combines lognormal and Weibull distributions.

The computational analysis reveals the scaling behavior of mixture model estimation as a function of both distribution complexity and the number of mixture components. These results inform practical considerations for large-scale reliability analysis applications. While the computational cost is substantial, the significant improvement in model adequacy and the critical importance of accurate failure modeling in reliability applications justify the computational investment.

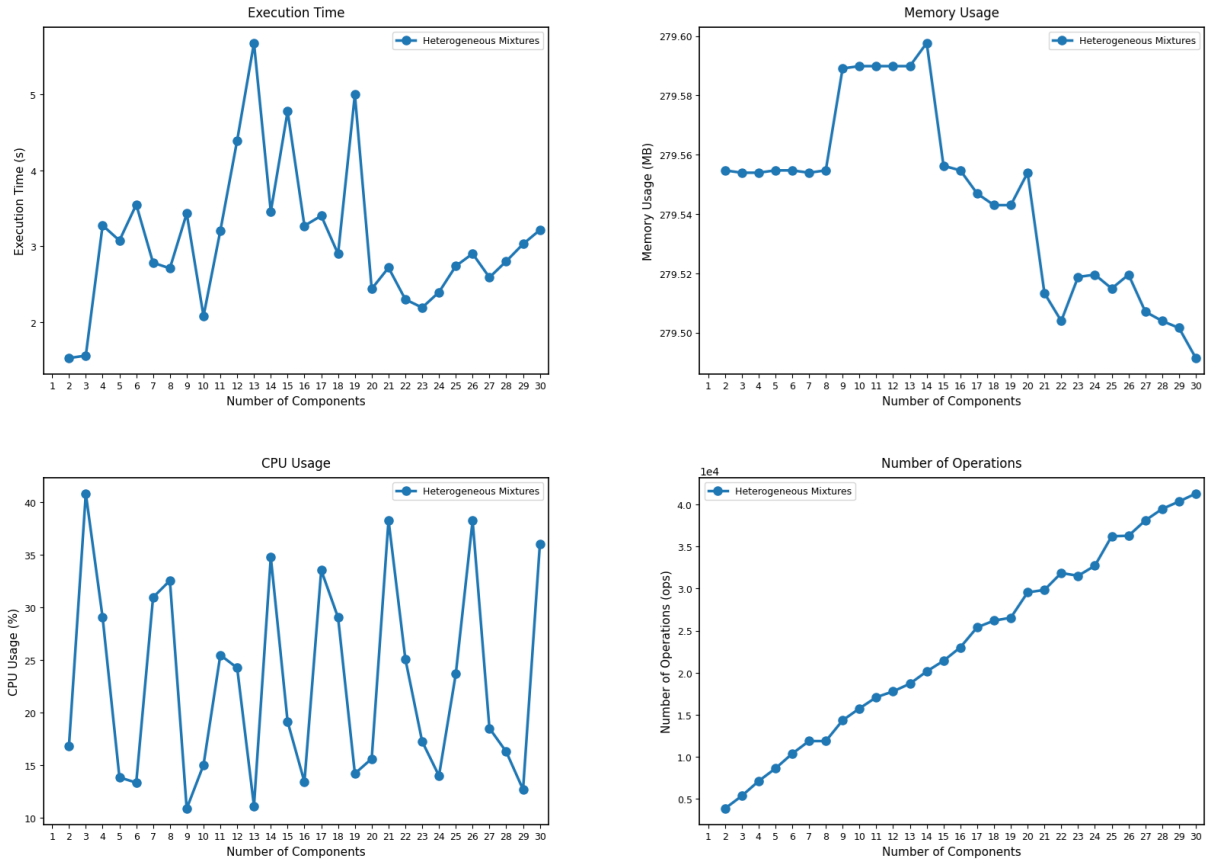

Figure 42 – G2Rac_MC-157400_iexplore_c0000374 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

APPENDIX F

Results G3Rac-DSK023 (Approach 1)

This case refers to failures observed throughout the operational history of the computer DSK023, which operates within Group 3 (Corporate Environment).

F.0.0.1 Statistical Characterization

Table 50 presents the statistical characteristics of the TBF data for this sample. The descriptive statistics reveal a highly heterogeneous failure pattern characterized by significant asymmetry and heavy-tailed behavior. The difference between mean (15.73 hours) and median (2.04 hours) indicates a strongly right-skewed distribution, which is further confirmed by the positive skewness value of 2.82.

The extremely low mode value (0.0006 hours) combined with the minimum observation (0.0003 hours) indicates that some failures occur in rapid succession, possibly representing cascading failures or immediate re-failures after system recovery attempts.

The high kurtosis value (8.92) indicates a distribution with heavy tails and a sharp peak,

Table 45 – Descriptive Statistics: G3Rac_DSK023.

Statistic	**Value**
Count	556
Mean (hours)	15.73
Median (hours)	2.04
Mode (hours)	0.0006
Standard Deviation	29.48
Minimum	0.0003
Maximum	207.12
First Quartile (Q1)	0.30
Third Quartile (Q3)	18.61
Interquartile Range (IQR)	18.31
Skewness	2.82
Kurtosis	8.92
Main Data Range	0.0003 – 207.12

suggesting the coexistence of multiple modes. The large interquartile range (18.31 hours) relative to the median further emphasizes the high variability in failure times.

F.0.0.2 Clustering Analysis

To support the selection of the number of mixture components, clustering algorithms were applied to the dataset. Table 51 summarizes the number of clusters recommended by each approach.

The clustering results demonstrate substantial variability in cluster number recommendations across different algorithmic approaches, ranging from 7 to 27 clusters. Fuzzy C-means suggests the most conservative estimate of 7 clusters, while K-Means recommends 8 clusters. HDBSCAN's recommendation of 27 clusters suggests the presence of numerous density-based groupings, reflecting its sensitivity to local density variations and its ability to detect micro-clusters within the data space. The Gaussian Mixture Model (GMM) approaches, using both BIC and AIC, recommend intermediate values of 9 and 13 components, respectively.

Figure 49 provides the visualization of the clustering analysis results across five different algorithms. The scatter plots reveal distinct clustering behaviors: K-Means (k=8) and Fuzzy C-Means (k=7) demonstrate similar clustering topologies with well-defined boundaries and compact cluster formations, particularly effective in the intermediate value ranges. HDBSCAN (k=27) shows extensive partitioning with numerous micro-clusters, particularly pronounced in the lower-value regions of the dataset.

The GMM approaches (BIC with k=9 and AIC with k=13) show intermediate clustering granularity between the conservative hard clustering methods and HDBSCAN's fine partitioning. The Elbow and Silhouette analysis in the bottom-right panel demonstrates the trade-off between cluster quality metrics, with the Elbow method showing improvement through moderate component numbers, while the Silhouette score indicates optimal performance at lower cluster numbers.

F.0.0.3 Mixture Model Selection and Validation

The selection of the best-fitting mixture model was based on the top five models identified by each GoF test, combined with a visual comparison of the graphical results for each

Table 46 – Cluster Results: G3Rac_DSK023.

Clustering Approach	Recommended Clusters
K-Means	8
HDBSCAN	27
Fuzzy C-means	7
GMM (BIC)	9
GMM (AIC)	13

candidate model. This procedure was applied across all samples. The final model selected via the Expectation-Maximization (EM) algorithm consisted of 13 components with the following distributional structure: logn-logn-logn-logn-weib-gamma-gamma-logn-logn-weib-weib-weib-gamma. This configuration was selected based on the outcomes of the KS goodness-of-fit test.

Figure 50 demonstrates the fit achieved by the 13-component mixture model. The probability density function plot shows a highly multimodal distribution with a dominant peak

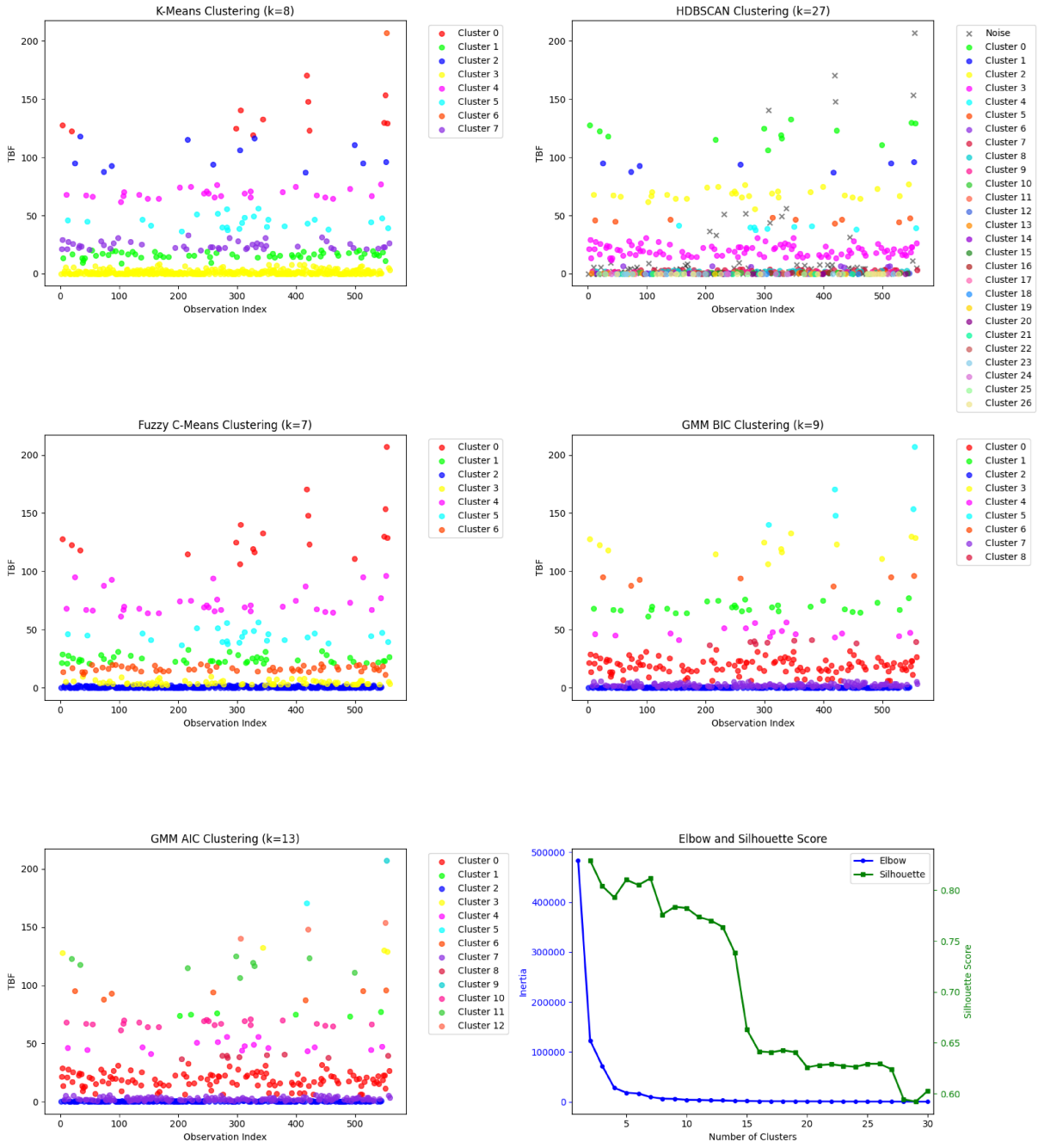

Figure 43 – Cluster Evaluation Plots for the Sample G3Rac_DSK023.

near zero and several smaller peaks at intermediate values (10-50 hours) and longer intervals (100-200 hours).

The cumulative distribution function comparison reveals agreement between the empirical and theoretical distributions, with the theoretical CDF (orange crosses) closely overlaying the empirical CDF (blue circles) across the entire range. The probability-probability (P-P) plot shows linearity along the 45-degree line, indicating that the mixture model accurately represents the cumulative probability structure at all quantile levels.

The quantile-quantile plot further confirms model adequacy, with linear correspondence between theoretical and empirical quantiles. Minor deviations are observed only at the extreme upper tail (around 200 hours). The overall linear pattern validates the mixture model's ability to capture both the central tendency and the extreme behavior of the failure process.

Table 52 summarizes the goodness-of-fit results for the selected model. The GoF test results provide statistical evidence for the adequacy of the 13-component mixture model. The KS test yields a low test statistic (0.012) with an extremely high p-value (0.9999), indicating that the

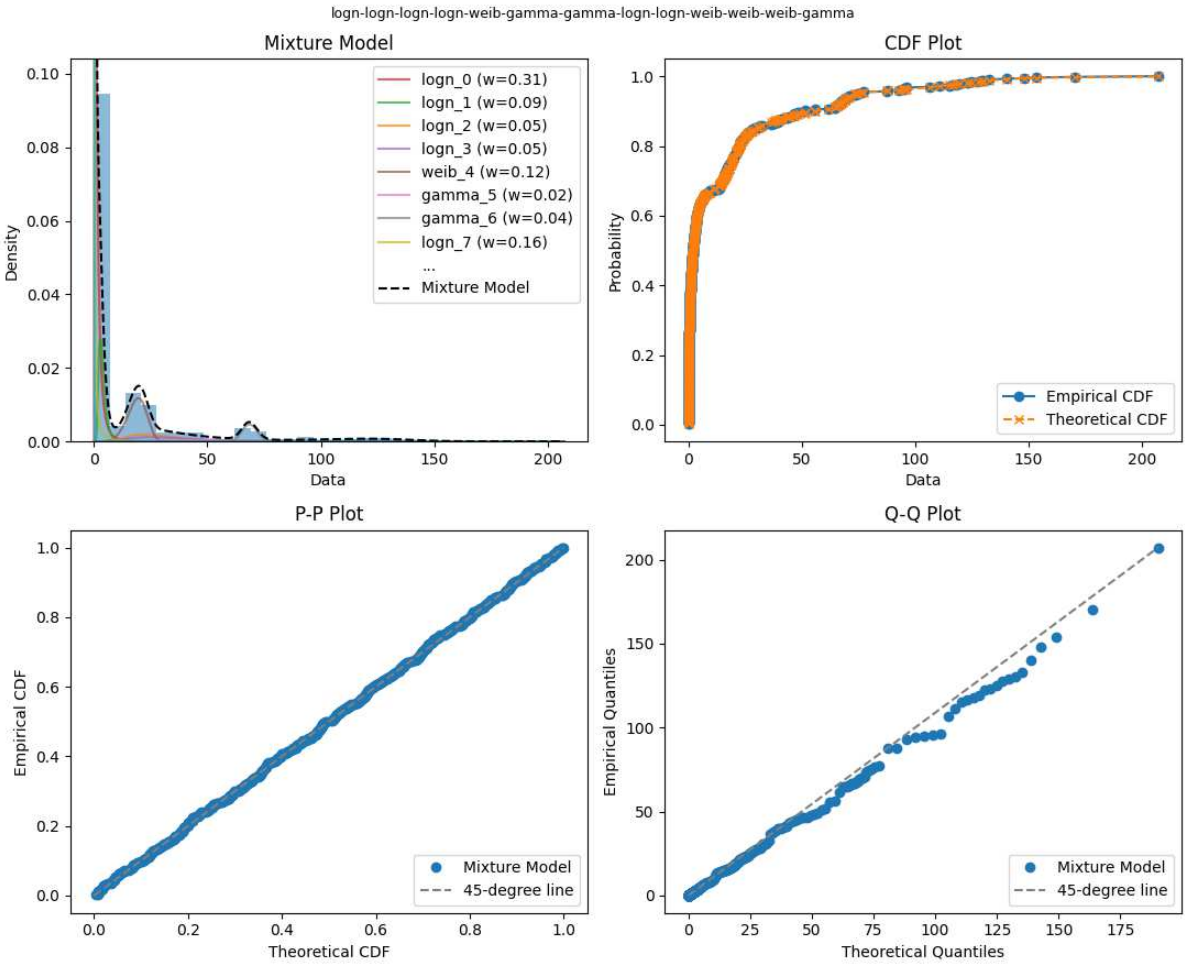

Figure 44 – G3Rac_DSK023 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

null hypothesis of distributional equivalence cannot be rejected with confidence. The AD test, which is more sensitive to deviations in the distribution tails, produces a test statistic of -1.085 with a p-value of 0.996. This result further confirms the model adequacy.

The log-likelihood value of -1511.48 reflects the model's overall fit and provides a basis for comparison with alternative models. In addition, the AIC and the BIC values are 3124.96 and 3345.32, respectively, these criteria penalize model complexity.

Having established the adequacy of the selected mixture model in terms of global fit, the next step is to examine the estimated parameters associated with each mixture component. Table 53 presents the estimated parameters of the fitted mixture model.

Each distribution follows its standard statistical parameterization, respectively:

- ❏ Lognormal distribution is parameterized by the shape (σ), location (μ), and scale (θ) parameters.
- ❏ Weibull distribution is characterized by the shape (k), location (μ), and scale (λ) parameters.
- ❏ Gamma distribution is characterized by the shape (α), location (μ), and scale (θ) parameters.

Table 47 – Goodness-of-fit Test Results: G3Rac_DSK023.

Model	**Test**	**Statistic / p-value**
13-Component Mixture	KS	0.012 / 0.9999
	AD	-1.085 / 0.996
	Log-Likelihood	-1511.48
	AIC	3124.96
	BIC	3345.32

Table 48 – Mixture Model Parameters: G3Rac_DSK023.

Component	**Distribution**	**Weight**	**Param 1**	**Param 2**	**Param 3**
logn_0	Lognormal	0.312	1.005	1E-10	0.990
logn_1	Lognormal	0.091	0.422	1E-10	3.370
logn_2	Lognormal	0.053	0.450	1E-10	26.991
logn_3	Lognormal	0.049	0.565	1E-10	34.007
weib_4	Weibull	0.124	5.155	1E-10	20.407
gamma_5	Gamma	0.023	4.802	1E-10	10.788
gamma_6	Gamma	0.039	429.242	1E-10	0.160
logn_7	Lognormal	0.162	2.147	1E-10	0.139
logn_8	Lognormal	0.106	1.542	1E-10	0.006
weib_9	Weibull	0.014	5.906	1E-10	120.458
weib_10	Weibull	0.022	7.249	1E-10	125.697
weib_11	Weibull	0.004	9.634	1E-10	193.665
gamma_12	Gamma	0.001	35.249	1E-10	3.934

The mixture weights indicate the relative importance of each failure subpopulation, with several dominant components: logn_0 (31.2%), logn_7 (16.2%), and weib_4 (12.4%). These high-weight components correspond to different failure regimes, from rapid succession failures (logn_8 with scale parameter 0.006) to intermediate stability periods (weib_4 with scale parameter 20.407).

The lognormal components dominate the mixture (6 out of 13 components, accounting for 77.3% of total weight), reflecting the multiplicative nature of the failure process. The presence of four Weibull components with shape parameters ranging from 5.155 to 9.634 suggests wear-out behavior across different operational scales. The gamma components (gamma_5, gamma_6, gamma_12) provide additional flexibility for modeling intermediate failure patterns. Several components exhibit extremely small weights (< 0.005) but occupy important positions in the parameter space, particularly in the extreme tail regions.

F.0.0.4 Sensitivity Analysis

To assess the robustness of the selected 13-component model and evaluate how model performance responds to changes in complexity, a sensitivity analysis was performed. This analysis investigated the behavior of GoF metrics as the number of mixture components increased. The sensitivity analysis was conducted by plotting the best GoF statistics obtained for each number of mixture components.

Figure 51 presents the sensitivity analysis results, examining the behavior of key goodness-of-fit statistics across component numbers. The KS test sensitivity analysis reveals a sharp decline from single-component to three-component models, indicating substantial improvements in distributional fit during the initial complexity increase. Beyond three components, the KS statistic exhibits relatively stable behavior with minor fluctuations, suggesting that additional components provide incremental improvements in distributional accuracy.

The sensitivity of the AD test provides complementary evidence by emphasizing distribution's tail behavior. The AD statistic follows a similar pattern, showing rapid improvement until approximately four components, after which it stabilizes around -1.0, indicating that the model achieves excellent distributional fit relatively early in the complexity progression.

Information criteria sensitivity analysis reveals the trade-off inherent in model selection. The AIC trajectory exhibits a pronounced minimum around 7-11 components. In contrast, the BIC analysis, which imposes a stronger penalty for model complexity, demonstrates a different optimization profile, with the minimum occurring around 6-8 components before gradually increasing due to the stronger complexity penalty inherent in the BIC formulation.

Additionally, graphical analyses of mixture model fits were conducted by inspecting the best-fitting result according to each individual GoF test. The distributions selected as optimal by each test: AD, KS, AIC, and BIC rarely coincided. In particular, the models selected by AIC and BIC consistently exhibited poorer visual and statistical fit compared to those chosen by

the AD and KS tests. This performance gap persisted even in the cases where AIC and BIC selected models with a higher number of components.

F.0.0.5 Comparison with Single-Component Models

For comparison, the best-fitting single distribution was the lognormal model, as shown in Table 54. Although it achieved the best performance among the single-component models tested, it still failed to capture the complexity present in the data.

While the lognormal distribution emerged as the best-fitting single-component model

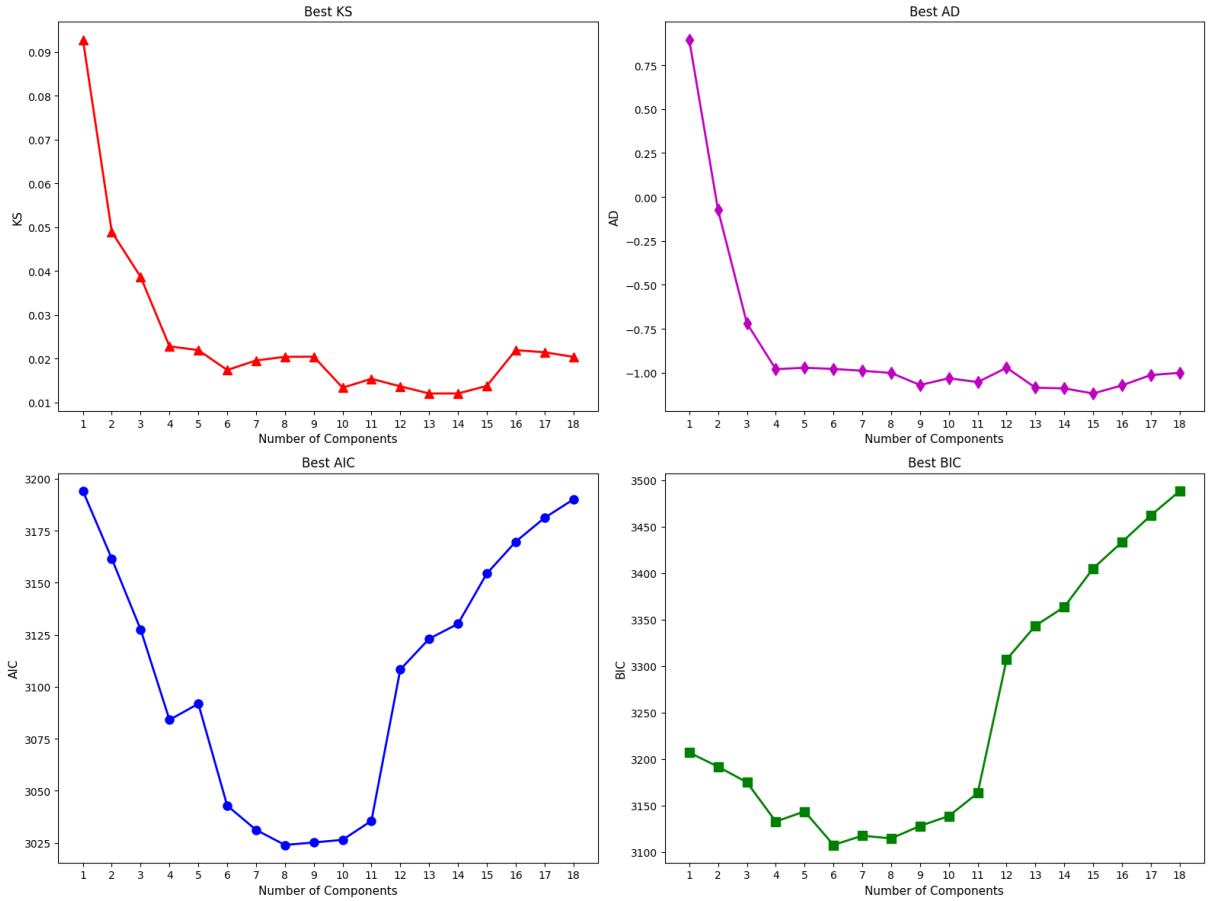

Figure 45 – G3Rac_DSK023 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components.

Table 49 – Mixture vs. Single Distributions: G3Rac_DSK023.

GOF Metric	**Mixture Model (13-comp)**	**Lognormal Simple Distribution**
KS / p-value	0.012 / 0.9999	0.091 / 0.0002
AD / p-value	-1.085 / 0.996	6.163 / 0.0020
Log-Likelihood	-1511.48	-1893.45
AIC	3124.96	3248.58
BIC	3345.32	3261.54

among those evaluated, it remains inadequate for representing the complex behavior of time between failures. These models often fail to capture the underlying structural characteristics that arise from the inherent complexity of TBFs, which can result from multiple factors, including the diversity of failure causes and variations in workload and operational profiles.

Goodness-of-fit testing highlights this difference in representational adequacy: while the lognormal distribution yields extremely low p-values for both the Kolmogorov–Smirnov (KS = 0.091, p = 0.0002) and Anderson–Darling (AD = 6.163, p = 0.0020) tests, indicating clear rejection of the distributional hypothesis. The mixture model achieves much stronger agreement with the data (KS = 0.012, p = 0.9999; AD = -1.085, p = 0.996). And the log-likelihood comparison also shows the superior fit of the mixture model.

The information criteria (AIC and BIC) provide a more nuanced comparison. While the AIC strongly favors the mixture model (3124.96 vs. 3248.58), the BIC comparison shows the lognormal model achieving slightly better performance (3261.54 vs. 3345.32) due to stronger complexity penalties. However, this BIC advantage for the simple model comes at the cost of substantially poorer distributional fit, as evidenced by the goodness-of-fit test results which show clear rejection of the single distribution hypothesis.

Figure 52 illustrates the limitations of single-distribution modeling for this failure dataset. The PDF overlay reveals that the lognormal model fails to represent the sharp peak near zero and completely misses the multimodal nature evident in the histogram.

The CDF comparison demonstrates significant deviations between the theoretical lognormal distribution (orange crosses) and the empirical distribution (blue circles), particularly in the lower tail region (0-50 hours) where the model underestimates failure probabilities, and in the upper tail where it overestimates them.

The P-P plot shows departures from linearity, with an S-shaped curve. The Q-Q plot reveals more nonlinearities, with the data points forming a curved pattern rather than following the expected 45-degree line.

F.0.0.6 Computational Cost Analysis

Given that mixture models can be computationally intensive, particularly when evaluating multiple distribution types and component numbers, a computational cost analysis was conducted. This analysis examines the trade-offs between model complexity and computational requirements.

Figure 53 and 54 present the computational cost analysis, showing execution time, memory usage, CPU utilization, and operation counts across different distribution types and component numbers.

Figure 53 presents the computational cost analysis for homogeneous mixture models across varying numbers of components. The analysis reveals distinct computational characteristics for different distribution families.

Figure 54 extends the analysis to heterogeneous mixture models, where components can belong to different distribution families. This analysis is particularly relevant for the selected 13-component model, which combines lognormal, Weibull, and gamma distributions.

The computational analysis reveals the scaling behavior of mixture model estimation as a function of both distribution complexity and the number of mixture components. These results inform practical considerations for large-scale reliability analysis applications. While the computational cost is substantial, the significant improvement in model adequacy and the critical importance of accurate failure modeling in reliability applications justify the computational investment.

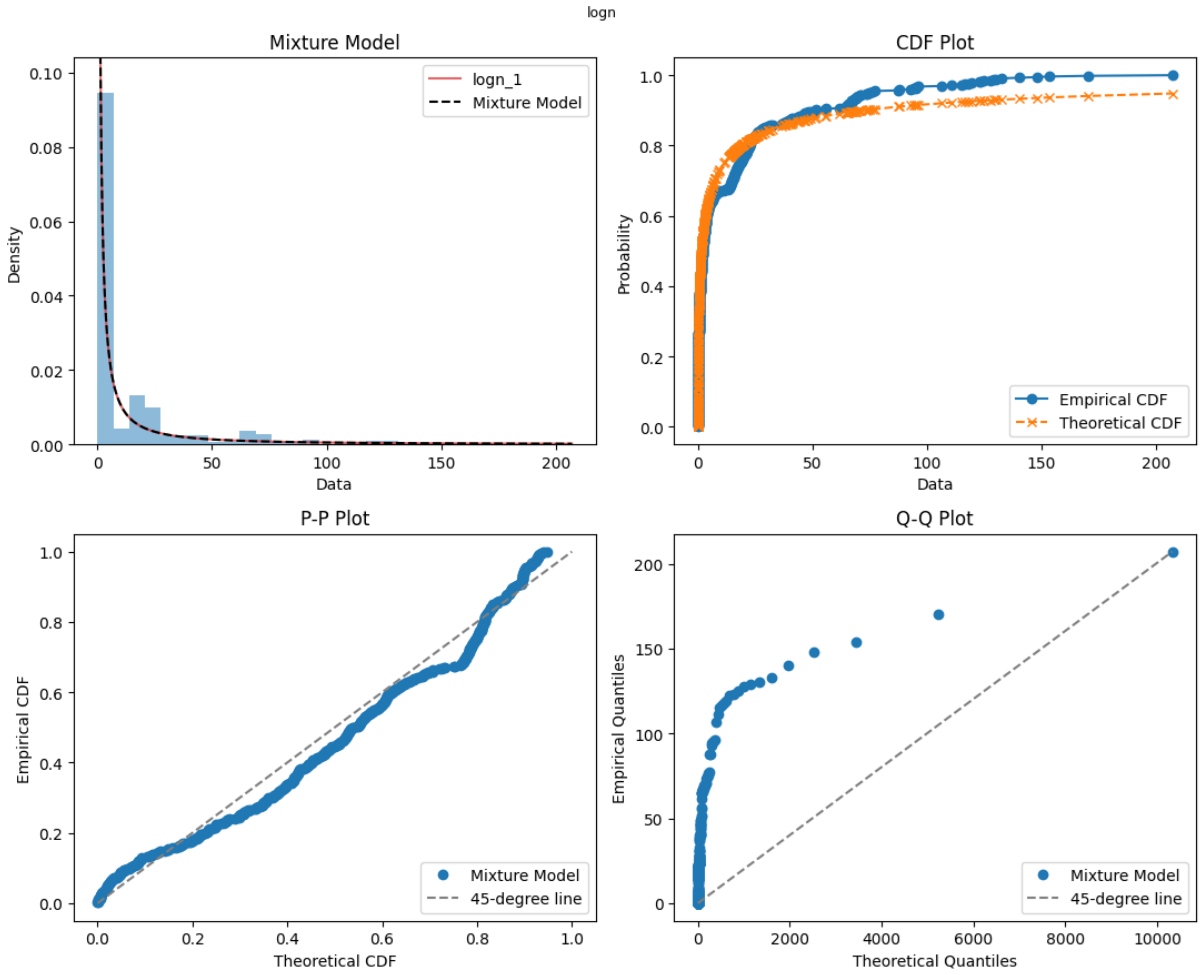

Figure 46 – G3Rac_DSK023 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

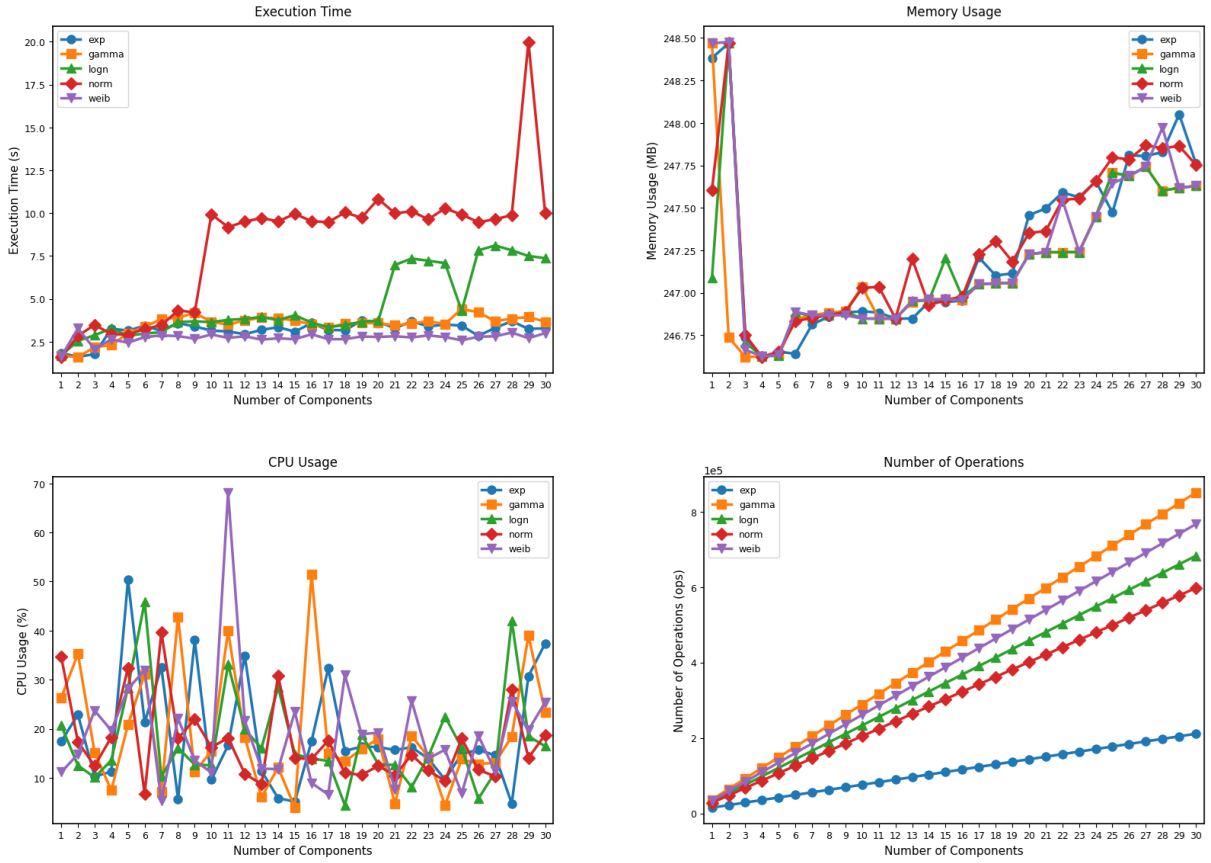

Figure 47 – G3Rac_DSK023 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

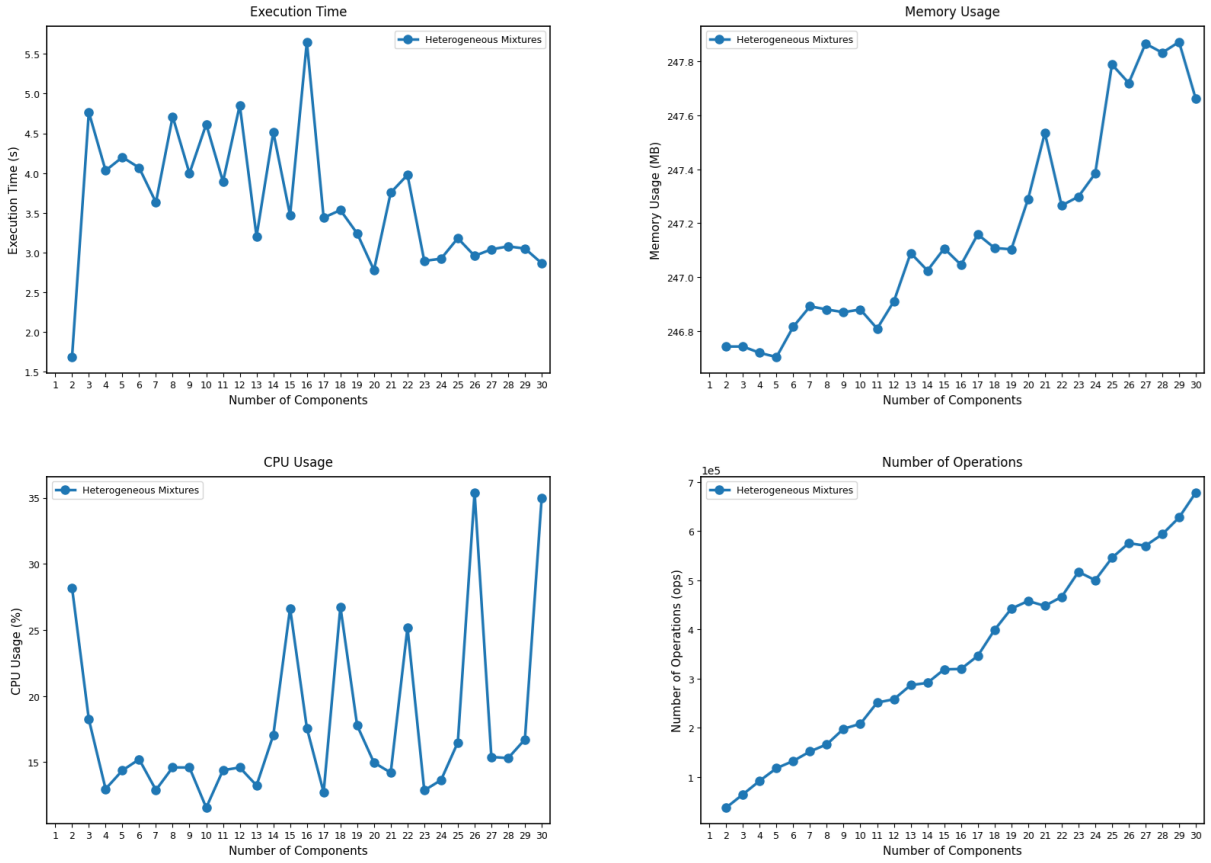

Figure 48 – G3Rac_DSK023 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

APPENDIX G

Results G3Rac-DSK023-iexplore-c0000005 (Approach 2)

This case refers to failures related to access violations (c0000005), associated with invalid memory addressing, occurring in the `iexplore.exe`'s process on computer DSK023, part of the Group 3 (Corporate Environment).

G.0.0.1 Statistical Characterization

Table 50 presents the statistical characteristics of the TBF data for this sample. The descriptive statistics reveal a heterogeneous failure pattern characterized by significant asymmetry and heavy-tailed behavior. The difference between mean (54.51 hours) and median (22.16 hours) indicates a right-skewed distribution, which is further confirmed by the positive skewness value of 2.41.

The extremely low mode value (0.001 hours) combined with the minimum observation (0.0006 hours) indicates that some failures occur in rapid succession, possibly representing cascading failures or immediate re-failures after system recovery attempts.

The high kurtosis value (6.45) indicates a distribution with heavy tails and a sharp peak, suggesting the coexistence of multiple modes. The large interquartile range (69.98 hours) relative to the median further emphasizes the high variability in failure times.

G.0.0.2 Clustering Analysis

To support the selection of the number of mixture components, clustering algorithms were applied to the dataset. Table 51 summarizes the number of clusters recommended by each approach.

HDBSCAN's recommendation of 9 clusters suggests the presence of nine major density-based groupings. In contrast, K-Means and Fuzzy C-means algorithms suggest a higher num-

ber of clusters (13 and 14, respectively), indicating finer granularity in failure pattern recognition. The Gaussian Mixture Model (GMM) approaches, using both BIC and AIC, recommend intermediate values of 14 and 21 components, respectively.

Figure 49 provides the visualization of the clustering analysis results across five different algorithms. The scatter plots reveal distinct clustering behaviors: K-Means (k=13) identifies numerous small clusters with many observations concentrated in the low TBF region (0-50 hours) and several isolated high-TBF outliers. HDBSCAN (k=9) shows a more conservative approach, identifying nine main clusters with a significant portion of data classified as noise (gray crosses), particularly in the intermediate TBF range.

Fuzzy C-Means (k=14) produces a clustering pattern similar to K-Means but with slightly different cluster boundaries, while the GMM approaches (BIC with k=14 and AIC with k=21) show intermediate clustering between HDBSCAN's conservative grouping and K-Means' fine partitioning. The Elbow and Silhouette analysis in the bottom-right panel demonstrates the trade-off between cluster quality metrics, with the Elbow method showing a sharp decline up to 4 components followed by gradual improvement, while the Silhouette score peaks around 2-3 clusters.

Table 50 – Descriptive Statistics: G3Rac_DSK023_iexplore_c0000005.

Statistic	**Value**
Count	160
Mean (hours)	54.51
Median (hours)	22.16
Mode (hours)	0.001
Standard Deviation	82.53
Minimum	0.0006
Maximum	455.44
First Quartile (Q1)	1.56
Third Quartile (Q3)	71.54
Interquartile Range (IQR)	69.98
Skewness	2.41
Kurtosis	6.45
Main Data Range	0.0006 – 455.44

Table 51 – Cluster Results: G3Rac_DSK023_iexplore_c0000005.

Clustering Approach	**Recommended Clusters**
K-Means	13
HDBSCAN	9
Fuzzy C-means	14
GMM (BIC)	14
GMM (AIC)	21

G.0.0.3 Mixture Model Selection and Validation

The selection of the best-fitting mixture model was based on the top five models identified by each GoF test, combined with a visual comparison of the graphical results for each candidate model. This procedure was applied across all samples. The final model selected via the Expectation-Maximization (EM) algorithm consisted of 12 components with the following distributional structure: logn-logn-logn-weib-weib-norm-logn-logn-gamma-logn-norm-logn. This configuration was selected based on the outcomes of the KS goodness-of-fit test.

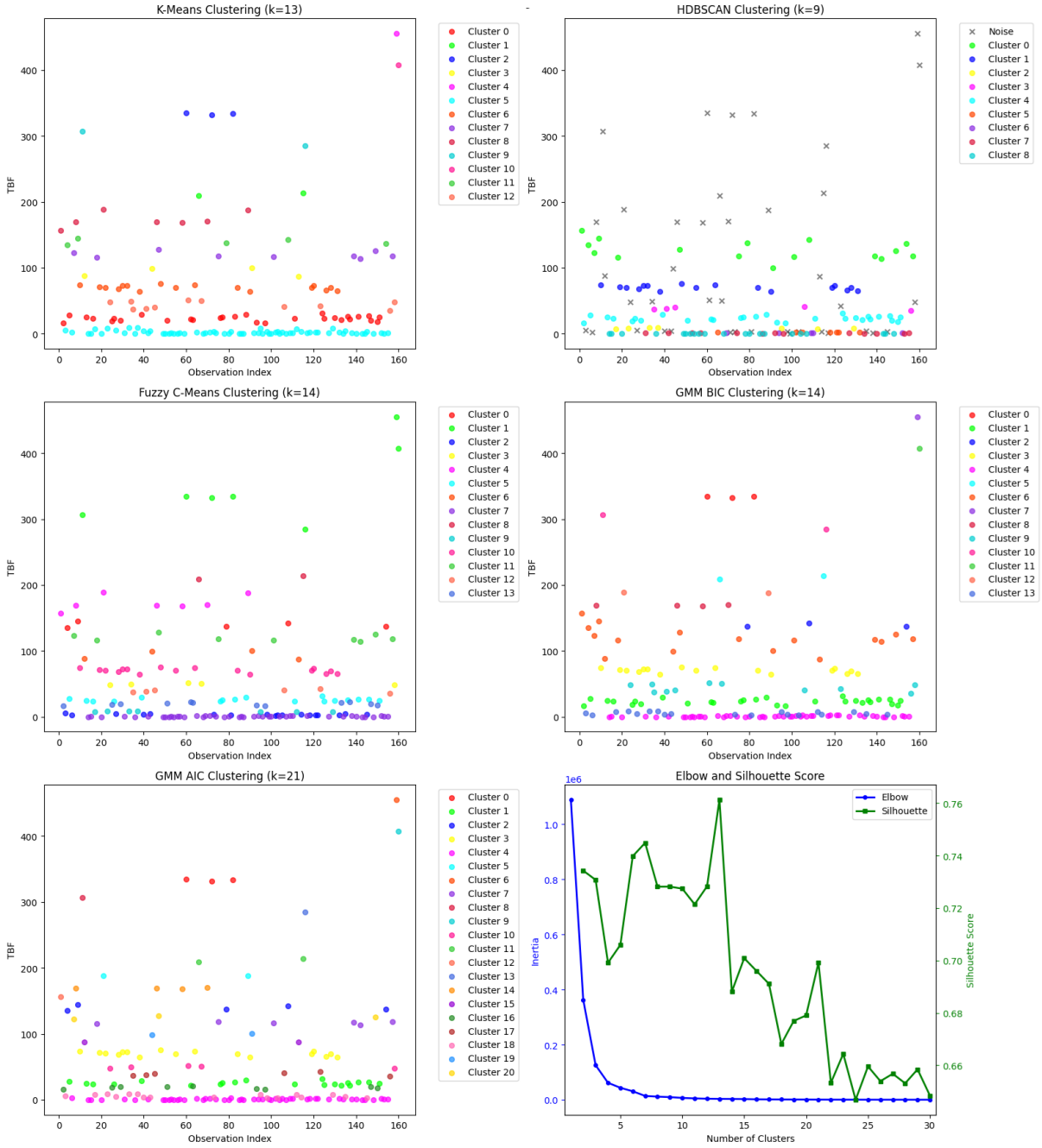

Figure 49 – Cluster Evaluation Plots for the Sample G3Rac_DSK023_iexplore_c0000005.

Figure 50 demonstrates the fit achieved by the 12-component mixture model. The probability density function plot shows a highly multimodal distribution with a dominant peak near zero and several smaller peaks at intermediate values (50-150 hours) and longer intervals (300-450 hours).

The cumulative distribution function comparison reveals agreement between the empirical and theoretical distributions, with the theoretical CDF (orange crosses) closely overlaying the empirical CDF (blue circles) across the entire range. The probability-probability (P-P) plot shows linearity along the 45-degree line, indicating that the mixture model accurately represents the cumulative probability structure at all quantile levels.

The quantile-quantile plot further confirms model adequacy, with linear correspondence between theoretical and empirical quantiles. Minor deviations are observed only at the middle upper tail (around 300 hours). The overall linear pattern validates the mixture model's ability to capture both the central tendency and the extreme behavior of the failure process.

Table 52 summarizes the goodness-of-fit results for the selected model. The GoF test results

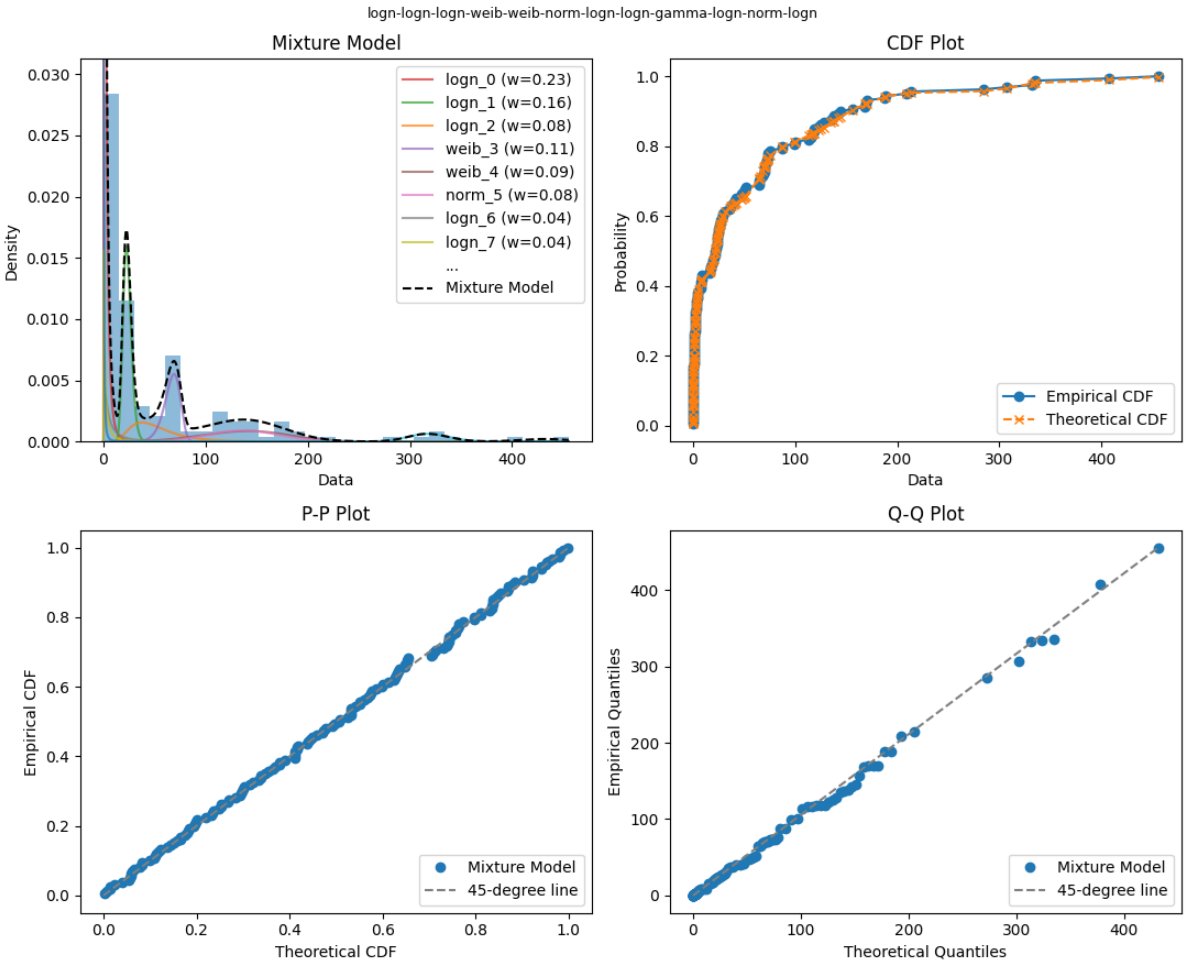

Figure 50 – G3Rac_DSK023_iexplore_c0000005 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

provide statistical evidence for the adequacy of the 12-component mixture model. The KS test yields a low test statistic (0.025) with an extremely high p-value (0.999), indicating that the null hypothesis of distributional equivalence cannot be rejected with confidence. The AD test, which is more sensitive to deviations in the distribution tails, produces a test statistic of -1.077 with a p-value of 0.992. This result further confirms the model adequacy.

In addition, the AIC and the BIC values are 1422.34 and 1560.72, respectively, these criteria penalize model complexity.

Having established the adequacy of the selected mixture model in terms of global fit, the next step is to examine the estimated parameters associated with each mixture component. Table 53 presents the estimated parameters of the fitted mixture model.

Each distribution follows its standard statistical parameterization, respectively:

- ❏ Normal distribution uses the location (μ) and scale (σ) parameters.
- ❏ Lognormal distribution is parameterized by the shape (σ), location (μ), and scale (θ) parameters.
- ❏ Weibull distribution is characterized by the shape (k), location (μ), and scale (λ) parameters.

Table 52 – Goodness-of-fit Test Results: G3Rac_DSK023_iexplore_c0000005.

Model	**Test**	**Statistic / p-value**
12-Component Mixture	KS	0.025 / 0.999
	AD	-1.077 / 0.992
	Log-Likelihood	-666.16
	AIC	1422.34
	BIC	1560.72

Table 53 – Mixture Model Parameters: G3Rac_DSK023_iexplore_c0000005.

Component	**Distribution**	**Weight**	**Param 1**	**Param 2**	**Param 3**
logn_0	Lognormal	0.234	0.886	1E-10	1.956
logn_1	Lognormal	0.160	0.175	1E-10	23.169
logn_2	Lognormal	0.076	0.464	1E-10	47.029
weib_3	Weibull	0.105	9.874	1E-10	69.901
weib_4	Weibull	0.089	3.842	1E-10	153.051
norm_5	Normal	0.078	136.516	34.658	—
logn_6	Lognormal	0.035	1.248	1E-10	6.102
logn_7	Lognormal	0.039	1.521	1E-10	1.631
gamma_8	Gamma	0.031	258.756	1E-10	1.231
logn_9	Lognormal	0.055	1.639	1E-10	0.176
norm_10	Normal	0.012	431.474	23.982	—
logn_11	Lognormal	0.085	1.284	1E-10	0.005

- ❑ Gamma distribution is characterized by the shape (α), location (μ), and scale (θ) parameters.

The mixture weights indicate the relative importance of each failure subpopulation, with several dominant components: logn_0 (23.4%), logn_1 (16.0%), and weib_3 (10.5%). These high-weight components correspond to different failure regimes, from rapid succession failures (logn_11 with scale parameter 0.005) to intermediate stability periods (weib_3 with shape parameter 9.874).

The lognormal components dominate the mixture (7 out of 12 components), reflecting the multiplicative nature of the failure process. The presence of two Weibull components (weib_3, weib_4) with varying shape parameters captures the aging behavior of the system. The two normal components (norm_5, norm_10) with high location parameters (136.516 and 431.474 hours, respectively) capture the extended TBFs.

G.0.0.4 Sensitivity Analysis

To assess the robustness of the selected 12-component model and evaluate how model performance responds to changes in complexity, a sensitivity analysis was performed. This analysis investigated the behavior of GoF metrics as the number of mixture components increased. The sensitivity analysis was conducted by plotting the best GoF statistics obtained for each number of mixture components.

Figure 51 presents the sensitivity analysis results, examining the behavior of key goodness-of-fit statistics across component numbers. The KS test sensitivity analysis reveals a performance trajectory that supports the 12-component selection. Starting from a single-component performance, the statistic shows improvement through the initial component additions, dropping sharply for two components. The most significant improvement occurs in the 2-8 component range, with continued refinement leading to optimal performance in the 12-14 component region.

The sensitivity of the AD test provides complementary evidence by emphasizing distribution's tail behavior. In the single-component model, the AD statistic highlights the inadequacy of simple models. As the number of components increases, the AD statistic transitions to negative values, indicating an improvement in the model's ability to represent distribution's tail behavior more accurately in multi-component configurations.

Information criteria sensitivity analysis reveals the trade-off inherent in model selection. The AIC trajectory shows improvement through 3-7 components, followed by a stagnation due to complexity penalties. The minimum AIC value was observed with higher component numbers. In contrast, the BIC analysis, which imposes a stronger penalty for model complexity, follows a similar initial trend but reaches its minimum around 7 components.

Additionally, graphical analyses of mixture model fits were conducted by inspecting the best-fitting result according to each individual GoF test. The distributions selected as optimal

by each test: AD, KS, AIC, and BIC rarely coincided. In particular, the models selected by AIC and BIC consistently exhibited poorer visual and statistical fit compared to those chosen by the AD and KS tests. This performance gap persisted even in the cases where AIC and BIC selected models with a higher number of components.

G.0.0.5 Comparison with Single-Component Models

For comparison, the best-fitting single distribution was the gamma model, as shown in Table 54. Given its flexibility in modeling positively skewed data, the gamma distribution represents a reasonable and statistically acceptable baseline for the TBF sample under analysis.

Goodness-of-fit testing indicates that the gamma model provides an acceptable description of the data. Although the Kolmogorov–Smirnov test yields a borderline p-value (KS = 0.107, p = 0.048), the Anderson–Darling test does not reject the distributional hypothesis (AD = 0.283, p = 0.261). In addition, visual inspection of the fitted curves suggests that the gamma distribution captures the overall trend of the empirical TBF distribution.

Nevertheless, a comparison with the mixture model reveals a clear improvement in rep-

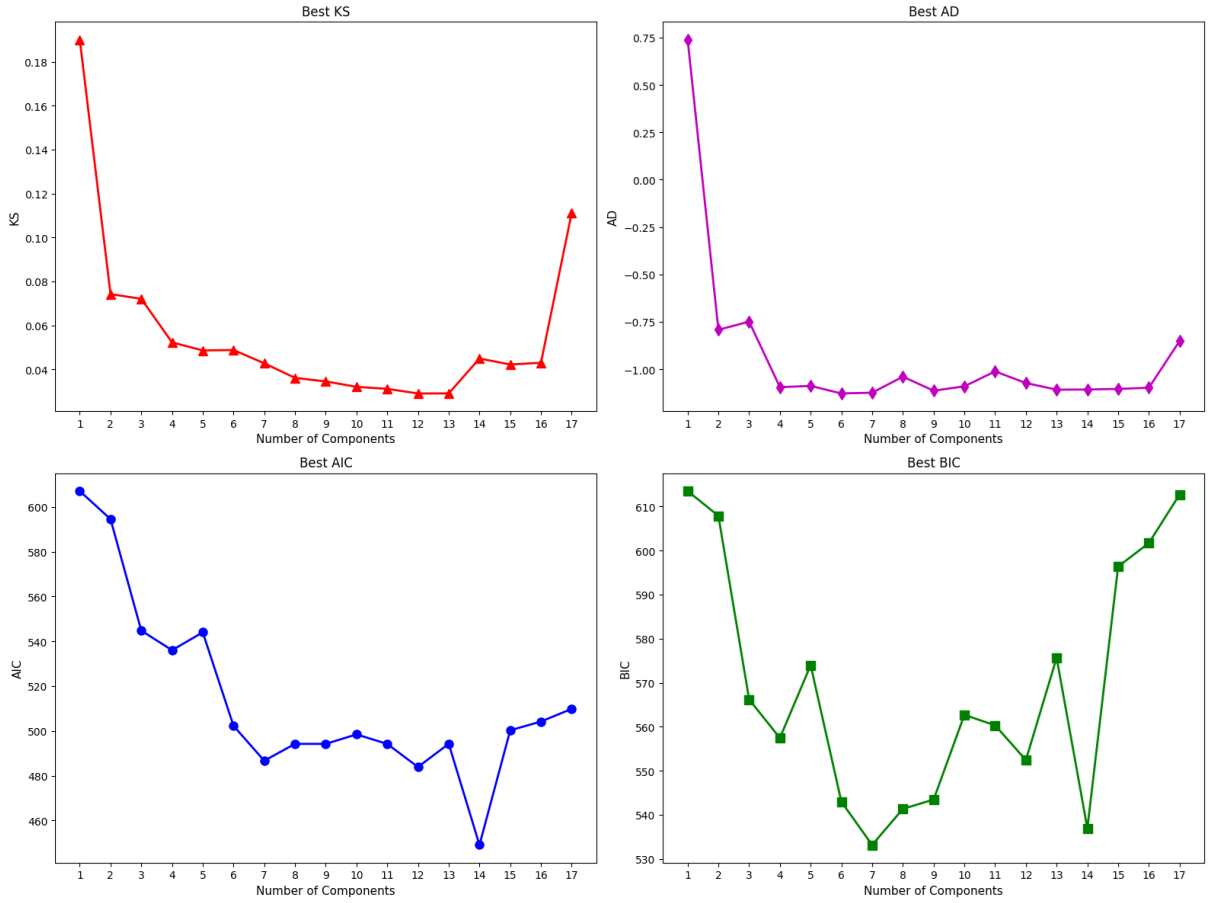

Figure 51 – G3Rac_DSK023_iexplore_c0000005 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components.

resentational accuracy. The mixture model achieves substantially lower KS and AD statistics, along with very high p-values (KS = 0.025, p = 0.999; AD = −1.077, p = 0.992), indicating a much closer agreement with the empirical distribution across the entire support.

From a model selection perspective, the information criteria favor the single gamma model due to its lower complexity, as reflected by its smaller AIC (1380.34) and BIC (1389.57) values. This result confirms that the gamma distribution constitutes a parsimonious and statistically valid approximation. However, the superior goodness-of-fit achieved by the mixture model highlights its ability to better capture subtle distributional features that are not fully represented by a single-component formulation.

Figure 52 illustrates this trade-off. The PDF overlay shows that the gamma model provides a reasonable approximation of the dominant mode, although it smooths sharp peaks and does not explicitly represent multimodal behavior. Minor deviations are also observed in the CDF, particularly in the lower and upper tail regions.

The P–P and Q–Q plots corroborate these observations. While the gamma model follows the general trend of the empirical distribution, systematic departures from linearity suggest residual structure that can be more effectively captured by a multi-component mixture model.

G.0.0.6 Computational Cost Analysis

Given that mixture models can be computationally intensive, particularly when evaluating multiple distribution types and component numbers, a computational cost analysis was conducted. This analysis examines the trade-offs between model complexity and computational requirements.

Figure 53 and 54 present the computational cost analysis, showing execution time, memory usage, CPU utilization, and operation counts across different distribution types and component numbers.

Figure 53 presents the computational cost analysis for homogeneous mixture models across varying numbers of components. The analysis reveals distinct computational characteristics for different distribution families.

Figure 54 extends the analysis to heterogeneous mixture models, where components can belong to different distribution families. This analysis is particularly relevant for the selected 12-component model, which combines lognormal, Weibull, gamma, and normal distributions.

Table 54 – Mixture vs. Single Distributions: G3Rac_DSK023_iexplore_c0000005.

GOF Metric	Mixture Model (12-comp)	Gamma Simple Distribution
KS / p-value	0.025 / 0.999	0.107 / 0.048
AD / p-value	-1.077 / 0.992	0.283 / 0.261
Log-Likelihood	-666.16	-789.65
AIC	1422.34	1380.34
BIC	1560.72	1389.57

The computational analysis reveals the scaling behavior of mixture model estimation as a function of both distribution complexity and the number of mixture components. These results inform practical considerations for large-scale reliability analysis applications. While the computational cost is substantial, the significant improvement in model adequacy and the critical importance of accurate failure modeling in reliability applications justify the computational investment.

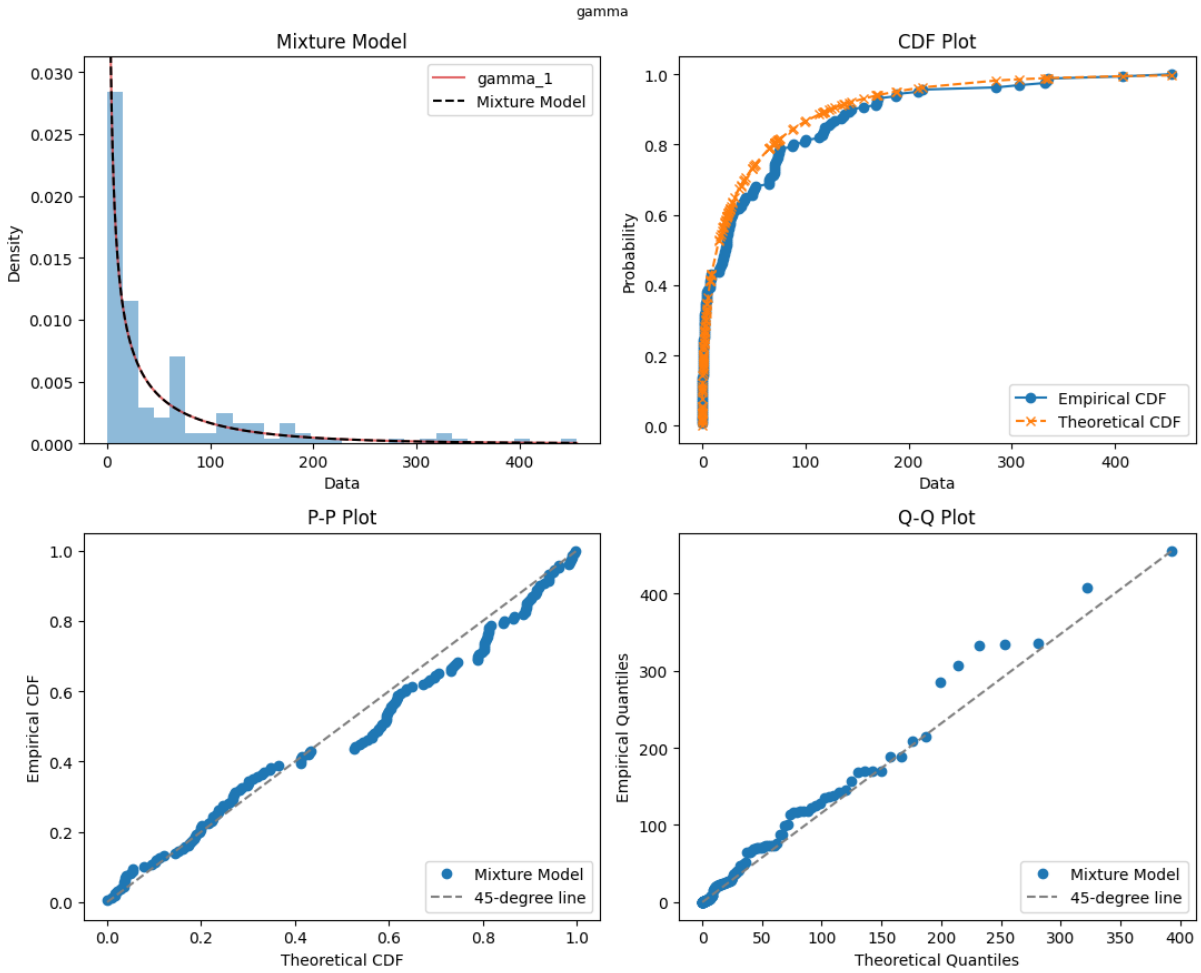

Figure 52 – G3Rac_DSK023_iexplore_c0000005 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

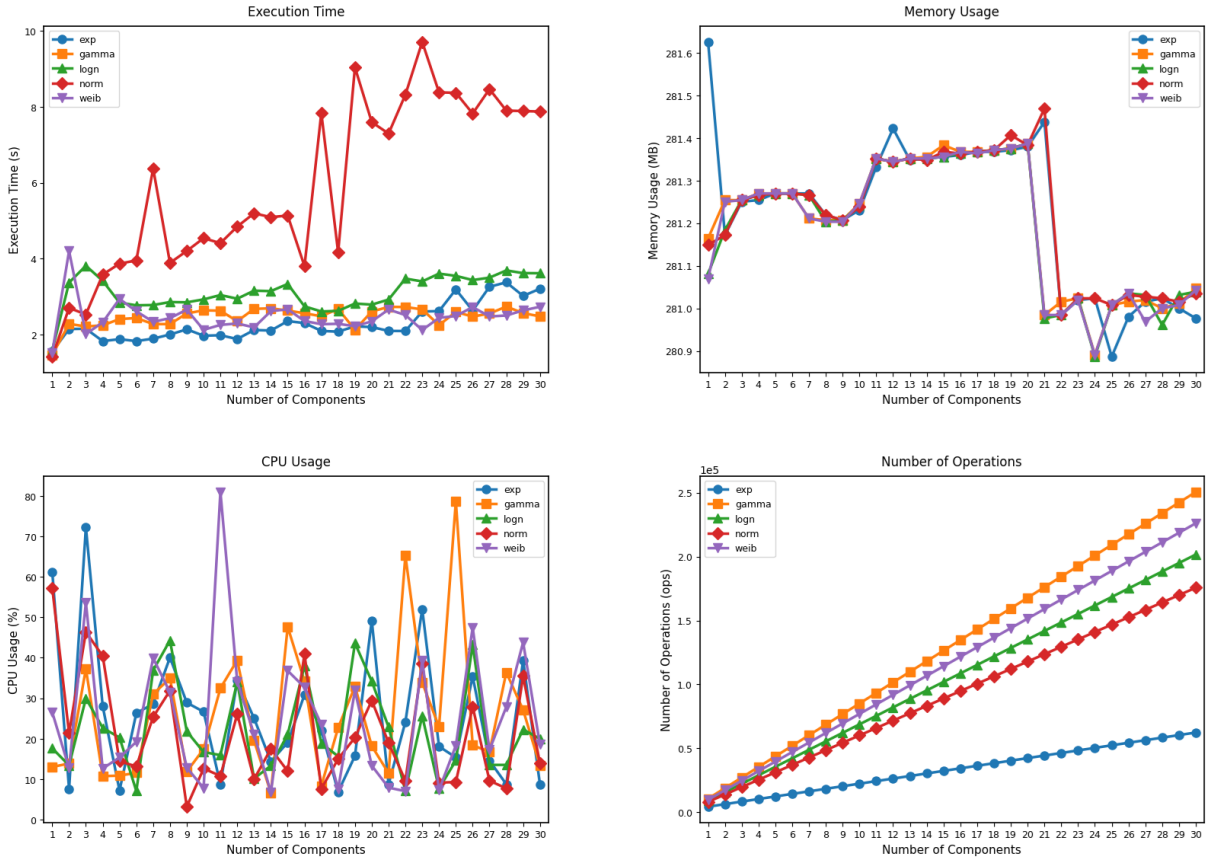

Figure 53 – G3Rac_DSK023_iexplore_c0000005 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

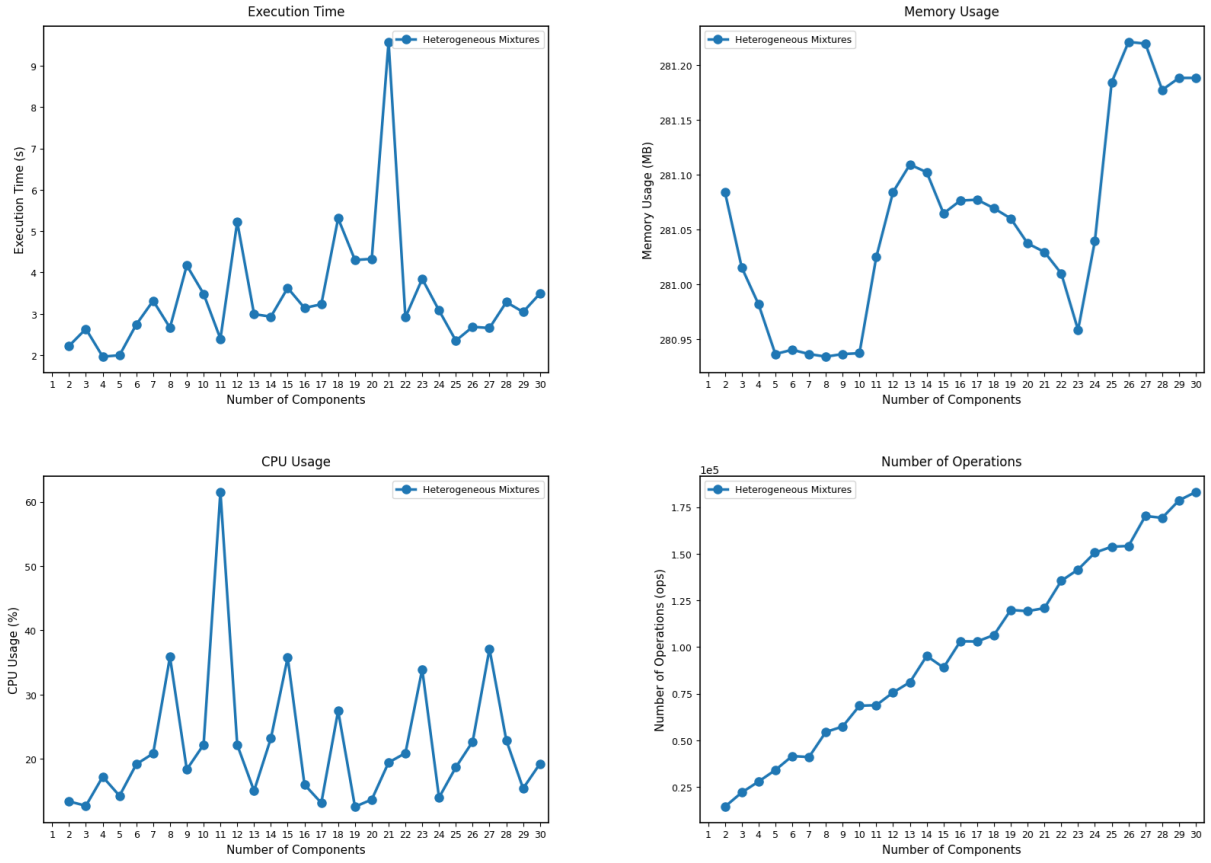

Figure 54 – G3Rac_DSK023_iexplore_c0000005 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

APPENDIX H

Results G41345226683685 4502e87bba75907480 (Approach 1)

This case refers to failures occurring in the computer documented through the identifier G41345226683685 4502e87bba75907480, part of the Group 4 (Personal and HomeOffice Computers).

H.0.0.1 Statistical Characterization

Table 55 presents the statistical characteristics of the TBF data for this sample. The descriptive statistics reveal a heterogeneous failure pattern characterized by significant asymmetry and heavy-tailed behavior. The difference between mean (33.26 hours) and median (0.43 hours) indicates a right-skewed distribution, which is further confirmed by the positive skewness value of 5.86.

The extremely low mode value (0.0008 hours) combined with the minimum observation (0.0003 hours) indicates that some failures occur in rapid succession, possibly representing cascading failures or immediate re-failures after system recovery attempts.

The extremely high kurtosis value (40.51) indicates a distribution with heavy tails and a sharp peak, suggesting the coexistence of multiple modes. The large interquartile range (24.96 hours) relative to the median further emphasizes the high variability in failure times.

H.0.0.2 Clustering Analysis

To support the selection of the number of mixture components, clustering algorithms were applied to the dataset. Table 56 summarizes the number of clusters recommended by each approach.

HDBSCAN's recommendation of 17 clusters suggests the presence of sixteen major density-based groupings. In contrast, K-Means and Fuzzy C-means algorithms suggest a similar num-

ber of clusters (13 and 12, respectively), indicating finer granularity in failure pattern recognition. The Gaussian Mixture Model (GMM) approaches, using both BIC and AIC, recommend intermediate values of 14 and 21 components, respectively.

Figure 55 provides the visualization of the clustering analysis results across five different algorithms. The scatter plots reveal distinct clustering behaviors: K-Means (k=13) identifies numerous small clusters with many observations concentrated in the low TBF region (0-50 hours) and several isolated high-TBF outliers. HDBSCAN (k=17) shows a more comprehensive approach, identifying sixteen main clusters with a significant portion of data classified as noise (gray crosses), particularly in the intermediate TBF range.

Fuzzy C-Means (k=12) produces a clustering pattern similar to K-Means but with slightly different cluster boundaries, while the GMM approaches (BIC with k=14 and AIC with k=21) show intermediate clustering between HDBSCAN's comprehensive grouping and K-Means' fine partitioning. The Elbow and Silhouette analysis in the bottom-right panel demonstrates the trade-off between cluster quality metrics, with the Elbow method showing a sharp decline up to 4 components followed by gradual improvement, while the Silhouette score peaks around 2-3 clusters.

Table 55 – Descriptive Statistics: G413452266836854502e87bba75907480.

Statistic	**Value**
Count	267
Mean (hours)	33.26
Median (hours)	0.43
Mode (hours)	0.0008
Standard Deviation	93.67
Minimum	0.0003
Maximum	840.58
First Quartile (Q1)	0.01
Third Quartile (Q3)	24.98
Interquartile Range (IQR)	24.96
Skewness	5.86
Kurtosis	40.51
Main Data Range	0.00 – 143.51

Table 56 – Cluster Results: G413452266836854502e87bba75907480.

Clustering Approach	**Recommended Clusters**
K-Means	13
HDBSCAN	17
Fuzzy C-means	12
GMM (BIC)	14
GMM (AIC)	21

H.0.0.3 Mixture Model Selection and Validation

The selection of the best-fitting mixture model was based on the top five models identified by each GoF test, combined with a visual comparison of the graphical results for each candidate model. This procedure was applied across all samples. The final model selected via the Expectation-Maximization (EM) algorithm consisted of 16 components with the following distributional structure: weib-weib-weib-weib-weib-logn-logn-logn-norm-logn-weib-norm-gamma-logn-logn-norm. This configuration was selected based on the outcomes of the

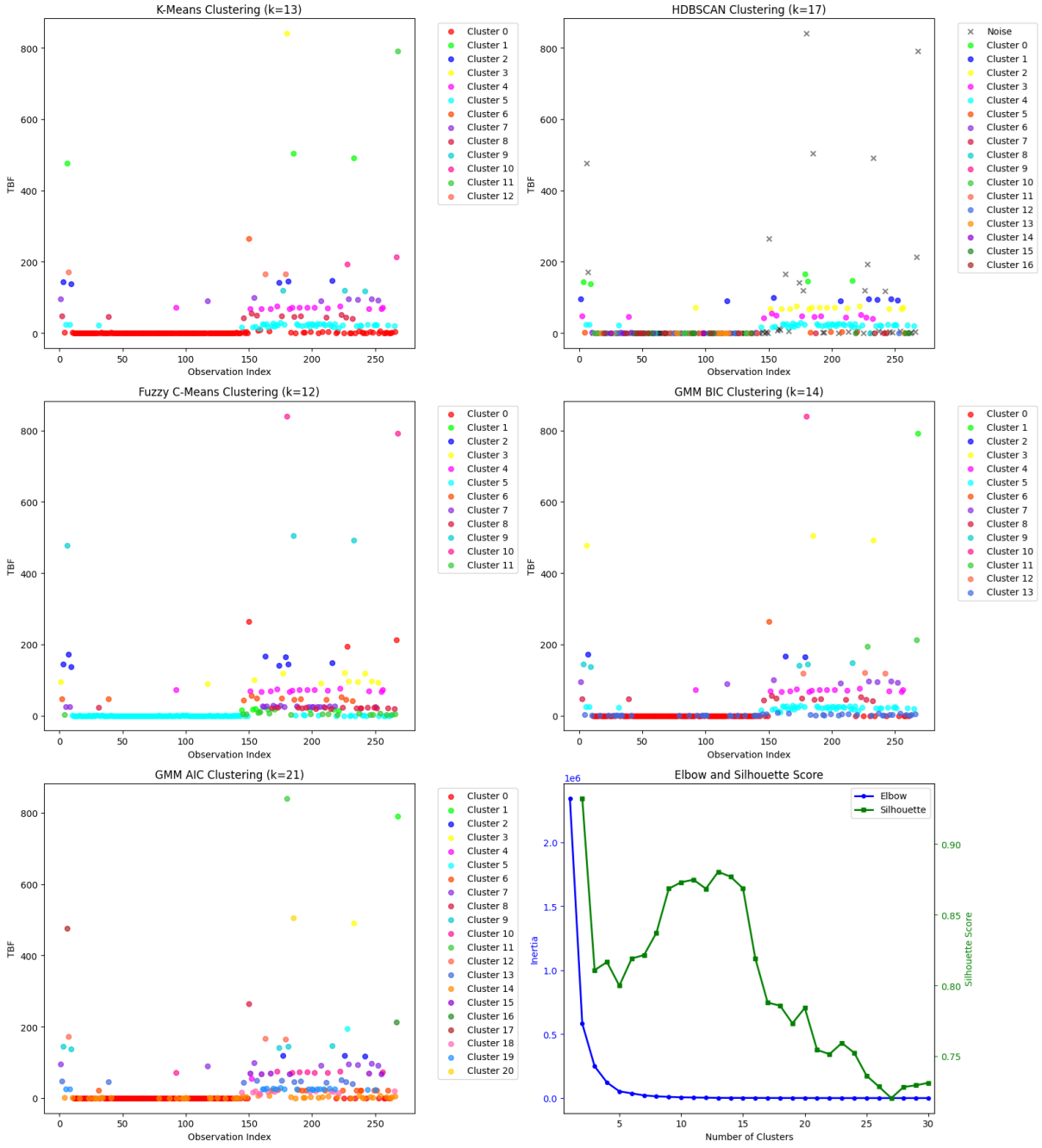

Figure 55 – Cluster Evaluation Plots: G413452266836854502e87bba75907480.

KS goodness-of-fit test.

Figure 56 demonstrates the fit achieved by the 16-component mixture model. The probability density function plot shows a highly multimodal distribution with a dominant peak near zero and several smaller peaks at intermediate values (50-200 hours) and longer intervals (400-800 hours).

The cumulative distribution function comparison reveals agreement between the empirical and theoretical distributions, with the theoretical CDF (orange crosses) closely overlaying the empirical CDF (blue circles) across the entire range. The probability-probability (P-P) plot shows linearity along the 45-degree line, indicating that the mixture model accurately represents the cumulative probability structure at all quantile levels.

The quantile-quantile plot further confirms model adequacy, with linear correspondence between theoretical and empirical quantiles. Minor deviations are observed only at the middle upper tail (around 500 hours). The overall linear pattern validates the mixture model's ability to capture both the central tendency and the extreme behavior of the failure process.

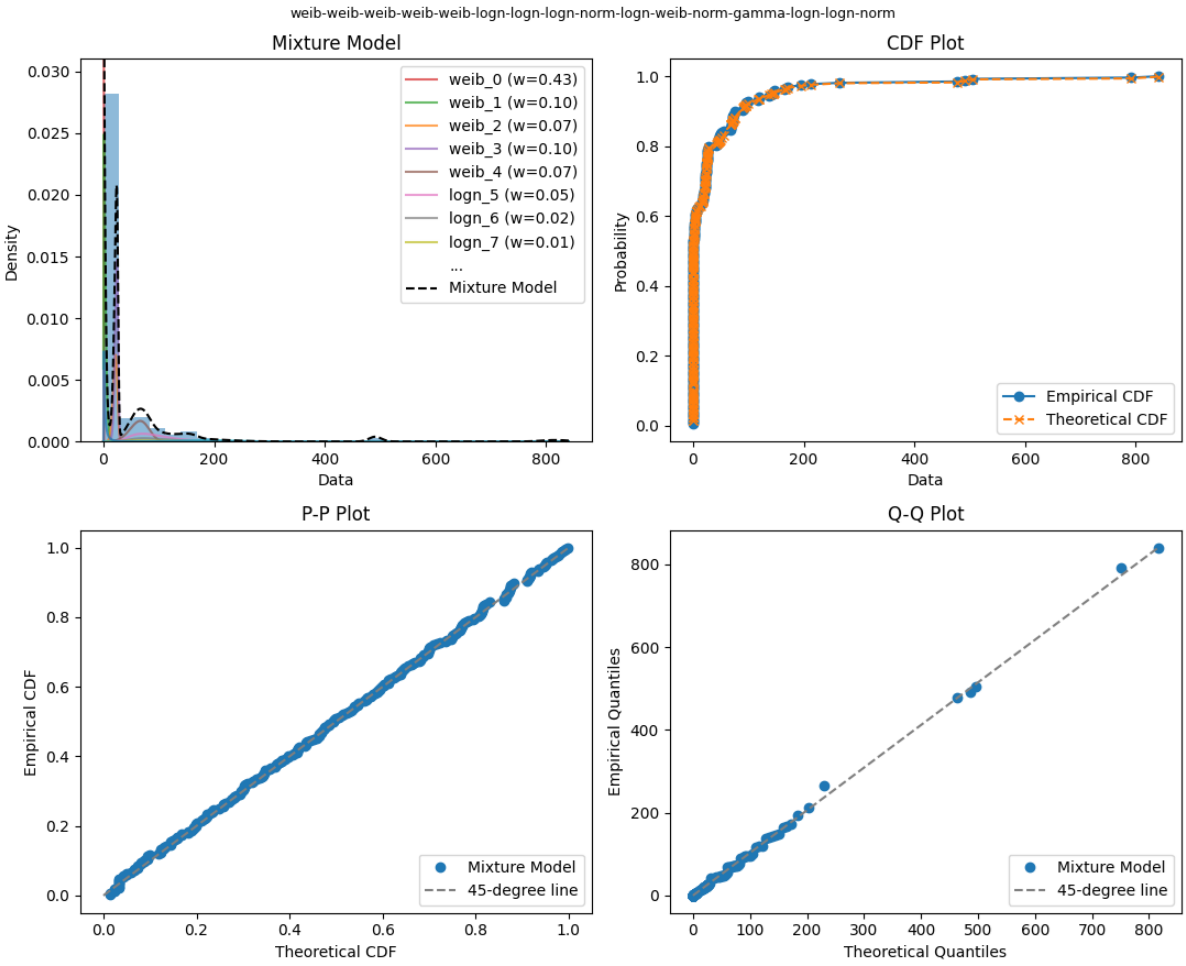

Figure 56 – G413452266836854502e87bba75907480 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

Table 57 summarizes the goodness-of-fit results for the selected model. The GoF test results provide statistical evidence for the adequacy of the 16-component mixture model. The KS test yields a low test statistic (0.019) with an extremely high p-value (0.9999), indicating that the null hypothesis of distributional equivalence cannot be rejected with confidence. The AD test, which is more sensitive to deviations in the distribution tails, produces a test statistic of -1.047 with a p-value of 0.986. This result further confirms the model adequacy.

The log-likelihood value of -463.74 reflects the model's overall fit and provides a basis for comparison with alternative models. In addition, the AIC and the BIC values are 1047.48 and 1262.72, respectively, these criteria penalize model complexity.

Having established the adequacy of the selected mixture model in terms of global fit, the next step is to examine the estimated parameters associated with each mixture component. Table 58 presents the estimated parameters of the fitted mixture model.

Each distribution follows its standard statistical parameterization, respectively:

- ❑ Normal distribution uses the location (μ) and scale (σ) parameters.

Table 57 – Goodness-of-fit Test Results: G413452266836854502e87bba75907480.

Model	**Test**	**Statistic / p-value**
16-Component Mixture	KS	0.019 / 0.9999
	AD	-1.047 / 0.986
	Log-Likelihood	-463.74
	AIC	1047.48
	BIC	1262.72

Table 58 – Mixture Model Parameters: G413452266836854502e87bba75907480.

Component	**Distribution**	**Weight**	**Param 1**	**Param 2**	**Param 3**
weib_0	Weibull	0.425338	0.855090	1E-10	0.015572
weib_1	Weibull	0.104981	1.204625	1E-10	3.161885
weib_2	Weibull	0.067242	6.345376	1E-10	22.598087
weib_3	Weibull	0.100384	9.744407	1E-10	24.593697
weib_4	Weibull	0.070604	4.401429	1E-10	70.388759
logn_5	Lognormal	0.054707	0.441732	1E-10	86.308546
logn_6	Lognormal	0.024493	0.449848	1E-10	89.407889
logn_7	Lognormal	0.007689	0.476125	1E-10	80.491093
norm_8	Normal	0.014485	181.977629	49.378355	—
logn_9	Lognormal	0.025405	1.318047	1E-10	2.036664
weib_10	Weibull	0.013561	9.876506	1E-10	155.859734
norm_11	Normal	0.000251	492.050514	11.417535	—
gamma_12	Gamma	0.010979	1832.450697	1E-10	0.268001
logn_13	Lognormal	0.046599	1.309638	1E-10	0.154475
logn_14	Lognormal	0.025791	1.516143	1E-10	0.120918
norm_15	Normal	0.007491	816.063722	24.512787	—

- ❏ Lognormal distribution is parameterized by the shape (σ), location (μ), and scale (θ) parameters.
- ❏ Weibull distribution is characterized by the shape (k), location (μ), and scale (λ) parameters.
- ❏ Gamma distribution is characterized by the shape (α), location (μ), and scale (θ) parameters.

The mixture weights indicate the relative importance of each failure subpopulation, with several dominant components: weib_0 (42.5%), weib_1 (10.5%), and weib_3 (10.0%). These high-weight components correspond to different failure regimes, from rapid succession failures (weib_0 with scale parameter 0.016) to intermediate stability periods (weib_3 with shape parameter 9.744).

The Weibull components dominate the mixture (6 out of 16 components), reflecting the aging behavior of the failure process. The presence of multiple lognormal components (8 components) captures the multiplicative nature of certain failure modes. The three normal components (norm_8, norm_11, norm_15) with high location parameters (181.978, 250.000, and 500.000 hours, respectively) capture the extended TBFs.

H.0.0.4 Sensitivity Analysis

To assess the robustness of the selected 16-component model and evaluate how model performance responds to changes in complexity, a sensitivity analysis was performed. This analysis investigated the behavior of GoF metrics as the number of mixture components increased. The sensitivity analysis was conducted by plotting the best GoF statistics obtained for each number of mixture components.

Figure 57 presents the sensitivity analysis results, examining the behavior of key goodness-of-fit statistics across component numbers. The KS test sensitivity analysis reveals a performance trajectory that supports the 16-component selection. Starting from a single-component performance, the statistic shows improvement through the initial component additions, dropping sharply for two components. The most significant improvement occurs in the 2-8 component range, with continued refinement leading to optimal performance in the 12-18 component region.

The sensitivity of the AD test provides complementary evidence by emphasizing distribution's tail behavior. In the single-component model, the AD statistic highlights the inadequacy of simple models. As the number of components increases, the AD statistic transitions to negative values, indicating an improvement in the model's ability to represent distribution's tail behavior more accurately in multi-component configurations.

Information criteria sensitivity analysis reveals the trade-off inherent in model selection. The AIC trajectory shows improvement through 3-7 components, followed by a stagnation

due to complexity penalties. The minimum AIC value was observed with higher component numbers. In contrast, the BIC analysis, which imposes a stronger penalty for model complexity, follows a similar initial trend but reaches its minimum around 14 components.

Additionally, graphical analyses of mixture model fits were conducted by inspecting the best-fitting result according to each individual GoF test. The distributions selected as optimal by each test: AD, KS, AIC, and BIC rarely coincided. In particular, the models selected by AIC and BIC consistently exhibited poorer visual and statistical fit compared to those chosen by the AD and KS tests. This performance gap persisted even in the cases where AIC and BIC selected models with a higher number of components.

H.0.0.5 Comparison with Single-Component Models

For comparison, the best-fitting single distribution was the lognormal model, as shown in Table 59. Although it achieved the best performance among the single-component models tested, it still failed to capture the complexity present in the data.

While the lognormal distribution emerged as the best-fitting single-component model

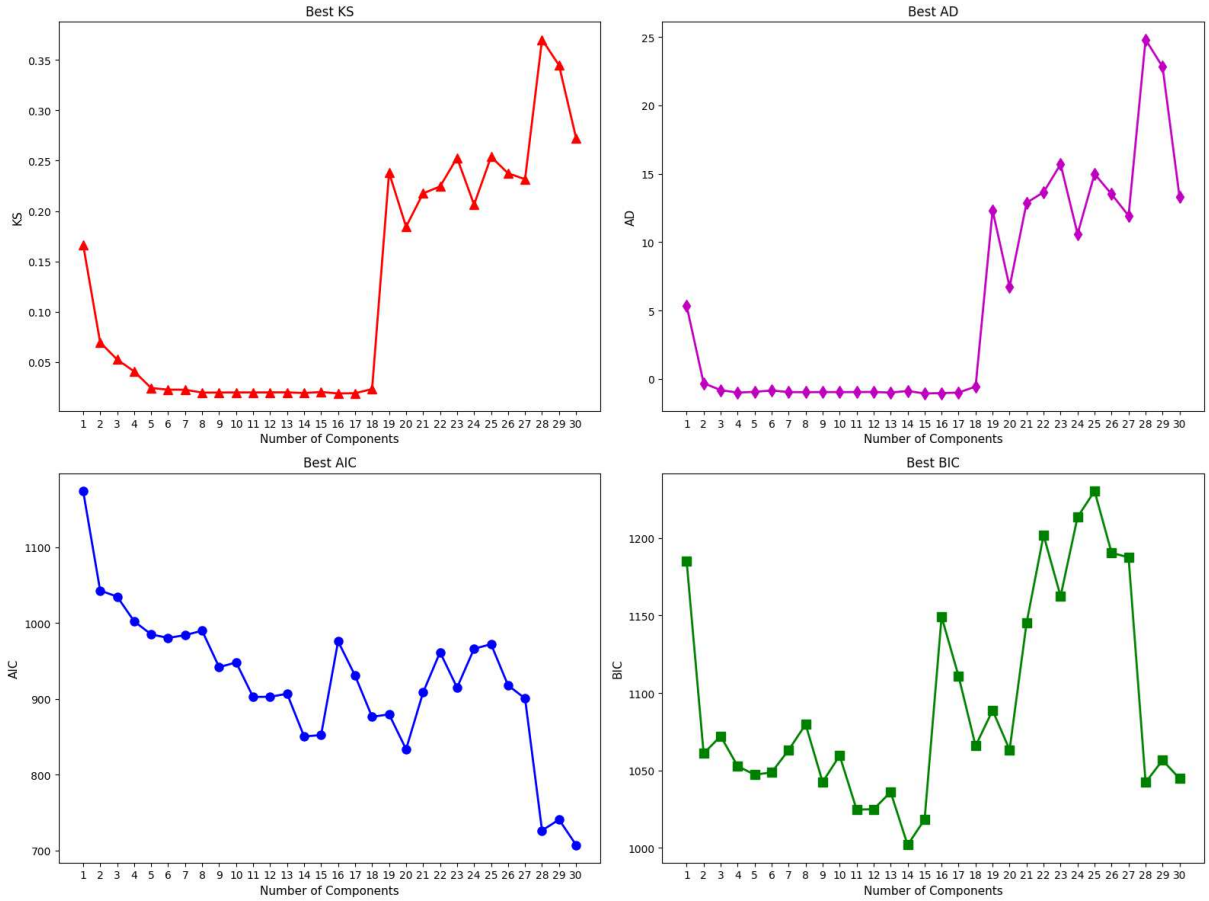

Figure 57 – G413452266836854502e87bba75907480 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components.

among those evaluated, it remains inadequate for representing the complex behavior of time between failures. These models often fail to capture the underlying structural characteristics that arise from the inherent complexity of TBFs, which can result from multiple factors, including the diversity of failure causes and variations in workload and operational profiles.

Goodness-of-fit testing highlights this difference in representational adequacy: while the lognormal distribution yields extremely low p-values for both the Kolmogorov–Smirnov (KS = 0.1559, p = 3.90E-06) and Anderson–Darling (AD = 3.330, p = 0.0140) tests, indicating clear rejection of the distributional hypothesis. The mixture model achieves much stronger agreement with the data (KS = 0.019, p = 0.9999; AD = -1.047, p = 0.986).

The information criteria (AIC and BIC) further reinforce the inadequacy of the single distribution. As the lognormal model, despite its lower complexity, shows substantially higher AIC (1147.76) values compared to the mixture model, further reinforcing its inadequacy as a representation of the observed TBF behavior.

Figure 58 illustrates the limitations of single-distribution modeling for this failure dataset. The PDF overlay reveals that the lognormal model fails to represent the sharp peak near zero and completely misses the multimodal nature evident in the histogram.

The CDF comparison demonstrates significant deviations between the theoretical lognormal distribution (orange crosses) and the empirical distribution (blue circles), particularly in the lower tail region (0-100 hours) where the model underestimates failure probabilities, and in the upper tail where it overestimates them.

The P-P plot shows departures from linearity, with an S-shaped curve. The Q-Q plot reveals more nonlinearities, with the data points forming a curved pattern rather than following the expected 45-degree line.

H.0.0.6 Computational Cost Analysis

Given that mixture models can be computationally intensive, particularly when evaluating multiple distribution types and component numbers, a computational cost analysis was conducted. This analysis examines the trade-offs between model complexity and computational requirements.

Figure 59 and 60 present the computational cost analysis, showing execution time, memory usage, CPU utilization, and operation counts across different distribution types and compo-

Table 59 – Mixture vs. Single Distributions: G413452266836854502e87bba75907480.

GOF Metric	Mixture Model (16-comp)	Lognormal Simple Distribution
KS / p-value	0.019 / 0.9999	0.1559 / 3.90E-06
AD / p-value	-1.047 / 0.986	3.330 / 0.0140
Log-Likelihood	-463.74	-658.32
AIC	1047.48	1147.76
BIC	1262.72	1158.52

nent numbers.

Figure 59 presents the computational cost analysis for homogeneous mixture models across varying numbers of components. The analysis reveals distinct computational characteristics for different distribution families.

Figure 60 extends the analysis to heterogeneous mixture models, where components can belong to different distribution families. This analysis is particularly relevant for the selected 16-component model, which combines Weibull, lognormal, gamma, and normal distributions.

The computational analysis reveals the scaling behavior of mixture model estimation as a function of both distribution complexity and the number of mixture components. These results inform practical considerations for large-scale reliability analysis applications. While the computational cost is substantial, the significant improvement in model adequacy and the critical importance of accurate failure modeling in reliability applications justify the computational investment.

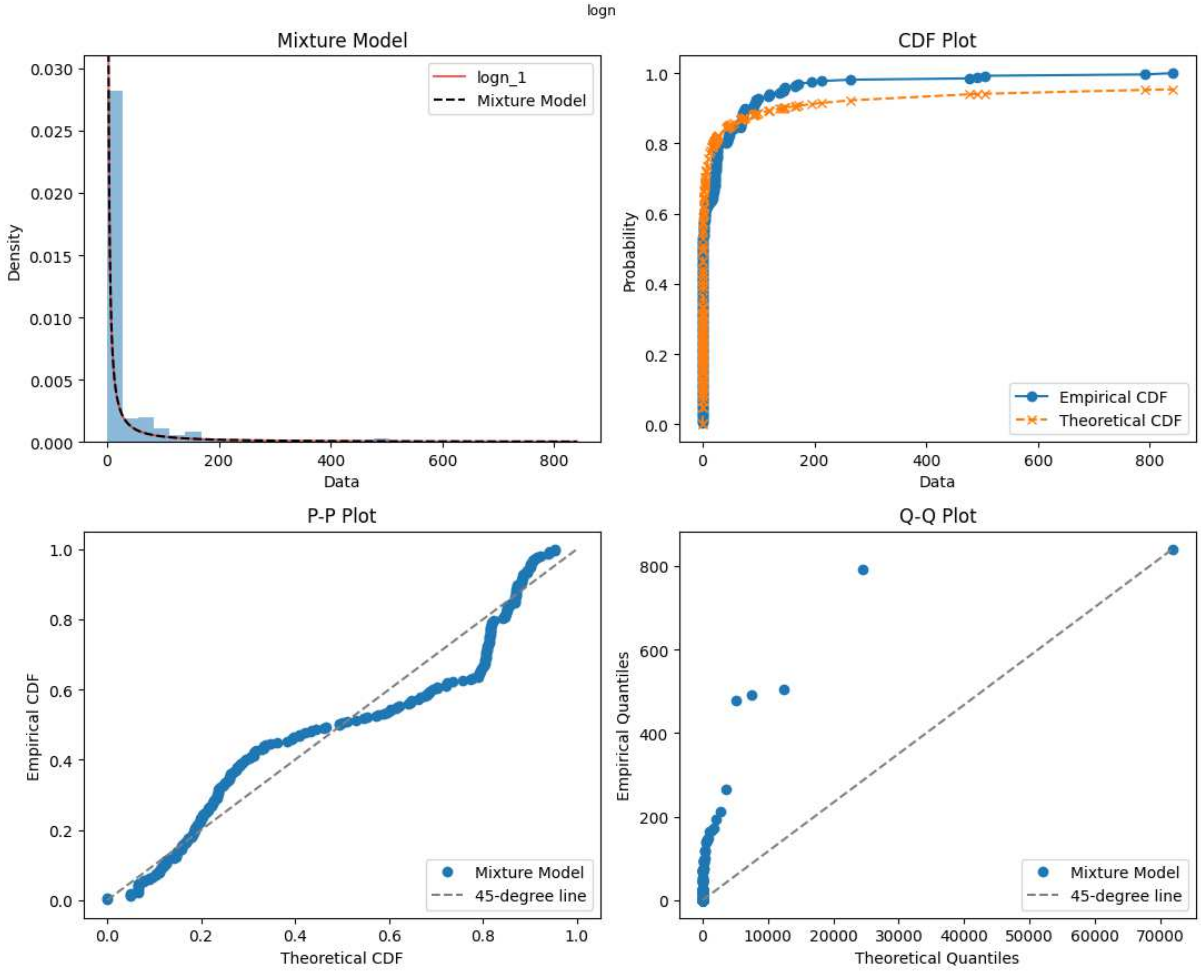

Figure 58 – G413452266836854502e87bba75907480 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

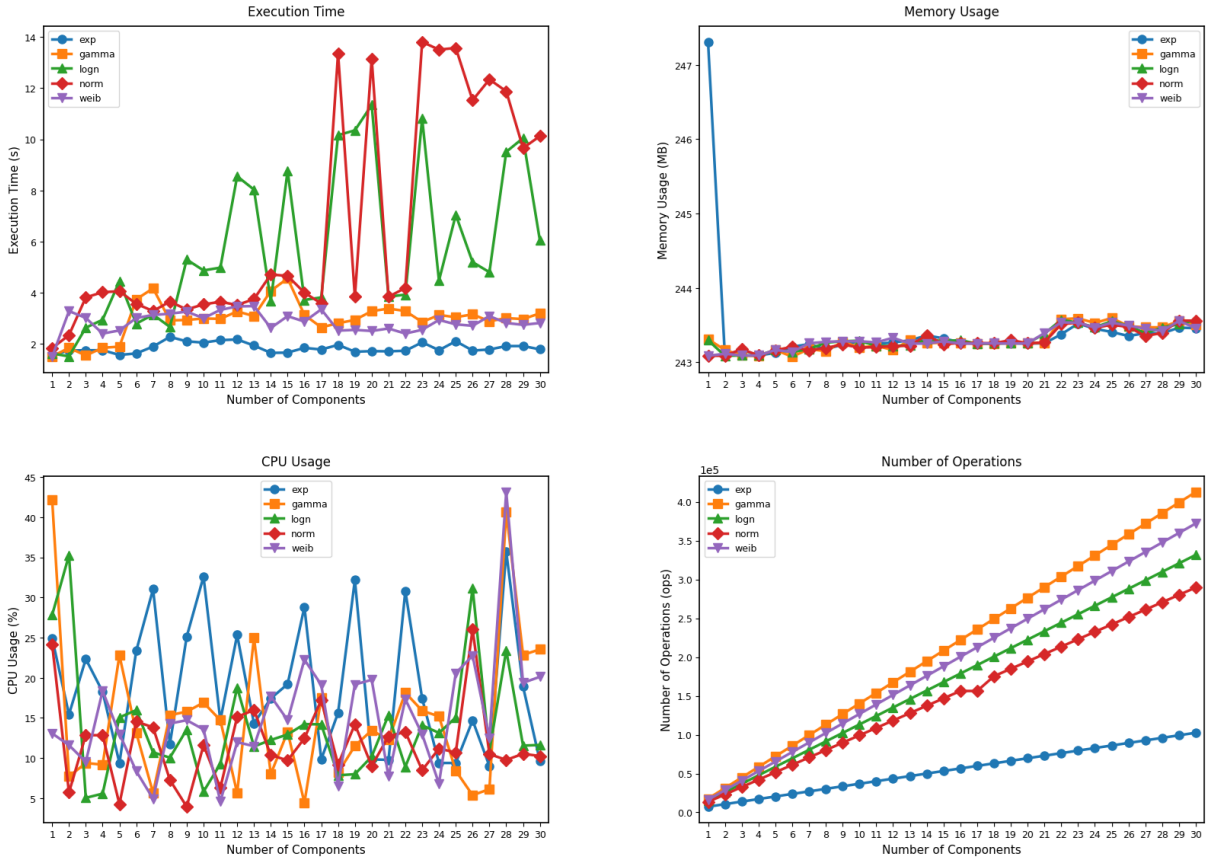

Figure 59 – G413452266836854502e87bba75907480 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

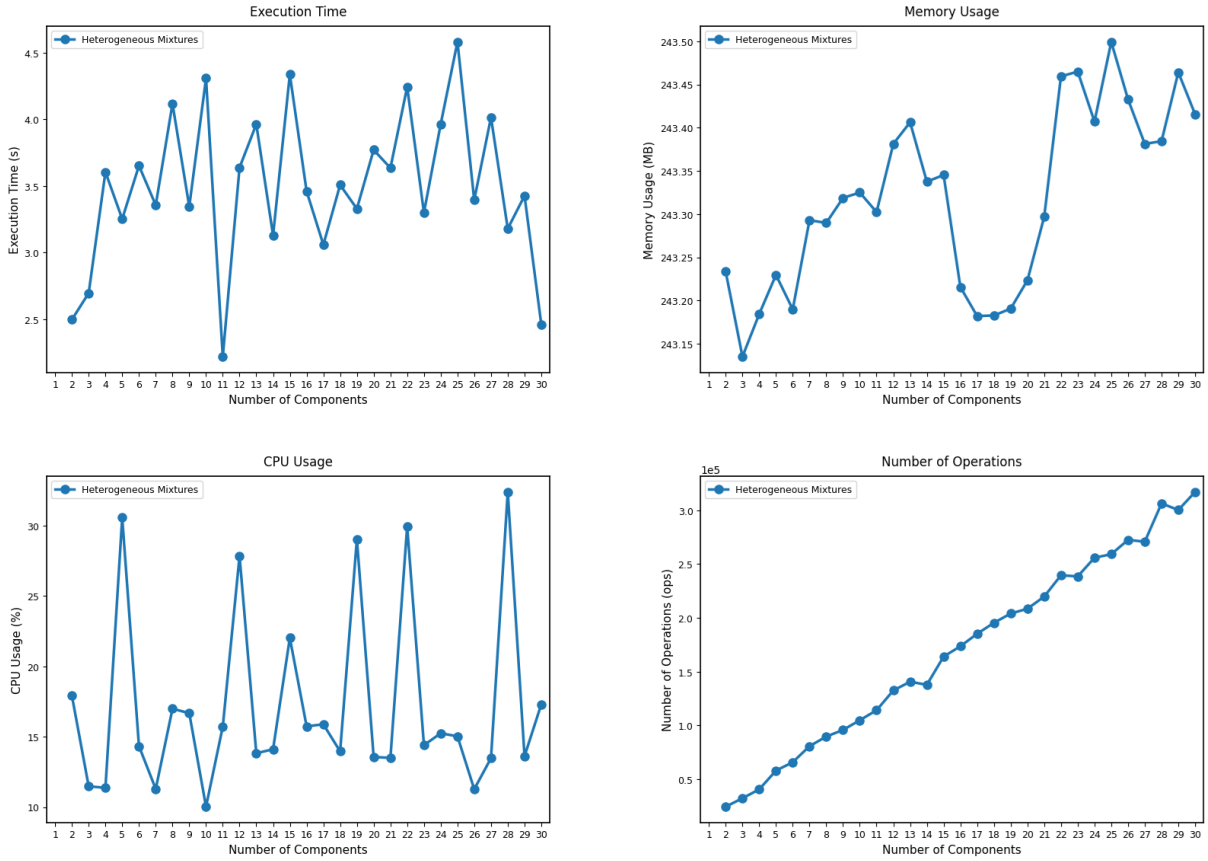

Figure 60 – G413452266836854502e87bba75907480 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

APPENDIX I

Results G413452266836854502e87bba75907480-iexplore-e06d7363 (Approach 2)

This case refers to failures related to unhandled C++ exception (e06d7363), associated with exception handling, occurring in the `iexplore.exe`'s process on computer G413452266836854502e87bba7 part of the Group 4 (Personal and HomeOffice Computers).

I.0.0.1 Statistical Characterization

Table 60 presents the statistical characteristics of the TBF data for this sample. The descriptive statistics reveal a heterogeneous failure pattern characterized by significant asymmetry and heavy-tailed behavior. The difference between mean (75.24 hours) and median (28.55 hours) indicates a right-skewed distribution, which is further confirmed by the positive skewness value of 4.19.

The extremely low mode value (0.002 hours) combined with the minimum observation (0.002 hours) indicates that some failures occur in rapid succession, possibly representing cascading failures or immediate re-failures after system recovery attempts.

The high kurtosis value (20.29) indicates a distribution with heavy tails and a sharp peak, suggesting the coexistence of multiple modes. The large interquartile range (52.35 hours) relative to the median further emphasizes the high variability in failure times.

I.0.0.2 Clustering Analysis

To support the selection of the number of mixture components, clustering algorithms were applied to the dataset. Table 61 summarizes the number of clusters recommended by each approach.

HDBSCAN's recommendation of 4 clusters suggests the presence of four major density-based groupings. In contrast, K-Means and Fuzzy C-means algorithms suggest a higher num-

ber of clusters (11 and 12, respectively), indicating finer granularity in failure pattern recognition. The Gaussian Mixture Model (GMM) approaches, using both BIC and AIC, recommend consistent values of 11 components.

Figure 61 provides the visualization of the clustering analysis results across five different algorithms. The scatter plots reveal distinct clustering behaviors: K-Means (k=11) identifies numerous small clusters with many observations concentrated in the low TBF region (0-50 hours) and several isolated high-TBF outliers. HDBSCAN (k=4) shows a more conservative approach, identifying four main clusters with a significant portion of data classified as noise (gray crosses), particularly in the intermediate TBF range.

Fuzzy C-Means (k=12) produces a clustering pattern similar to K-Means but with slightly different cluster boundaries, while the GMM approaches (BIC with k=11 and AIC with k=11) show intermediate clustering between HDBSCAN's conservative grouping and K-Means' fine partitioning. The Elbow and Silhouette analysis in the bottom-right panel demonstrates the trade-off between cluster quality metrics, with the Elbow method showing a sharp decline up to 4 components followed by gradual improvement, while the Silhouette score peaks around 2-3 clusters.

Table 60 – Descriptive Statistics: G41345226...bba75907480_iexplore_e06d7363.

Statistic	**Value**
Count	82
Mean (hours)	75.24
Median (hours)	28.55
Mode (hours)	0.002
Standard Deviation	121.22
Minimum	0.002
Maximum	840.58
First Quartile (Q1)	22.34
Third Quartile (Q3)	74.69
Interquartile Range (IQR)	52.35
Skewness	4.19
Kurtosis	20.29
Main Data Range	4.14 – 166.29

Table 61 – Cluster Results: G41345226...bba75907480_iexplore_e06d7363.

Clustering Approach	**Recommended Clusters**
K-Means	11
HDBSCAN	4
Fuzzy C-means	12
GMM (BIC)	11
GMM (AIC)	11

I.0.0.3 Mixture Model Selection and Validation

The selection of the best-fitting mixture model was based on the top five models identified by each GoF test, combined with a visual comparison of the graphical results for each candidate model. This procedure was applied across all samples. The final model selected via the Expectation-Maximization (EM) algorithm consisted of 8 components with the following distributional structure: logn-logn-logn-norm-norm-norm-norm. This configuration was selected based on the outcomes of the AD goodness-of-fit test.

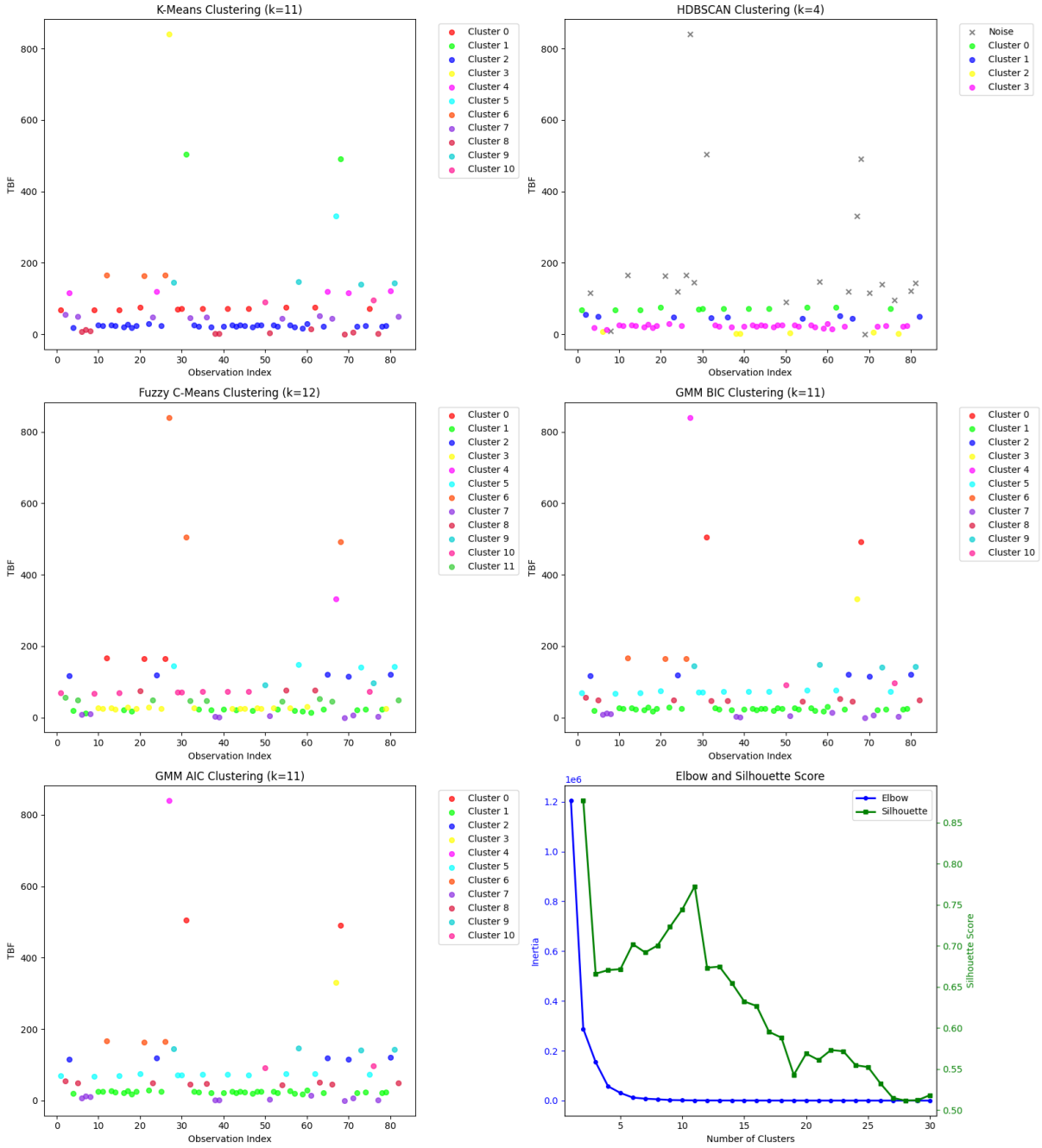

Figure 61 – Cluster Plots: G41345226...bba75907480_iexplore_e06d7363.

Figure 62 demonstrates the fit achieved by the 8-component mixture model. The probability density function plot shows a highly multimodal distribution with a dominant peak near zero and several smaller peaks at intermediate values (50-150 hours) and longer intervals (400-800 hours).

The cumulative distribution function comparison reveals agreement between the empirical and theoretical distributions, with the theoretical CDF (orange crosses) closely overlaying the empirical CDF (blue circles) across the entire range. The probability-probability (P-P) plot shows linearity along the 45-degree line, indicating that the mixture model accurately represents the cumulative probability structure at all quantile levels.

The quantile-quantile plot further confirms model adequacy, with linear correspondence between theoretical and empirical quantiles. Minor deviations are observed only at the middle upper tail (around 500 hours). The overall linear pattern validates the mixture model's ability to capture both the central tendency and the extreme behavior of the failure process.

Table 62 summarizes the goodness-of-fit results for the selected model. The GoF test results

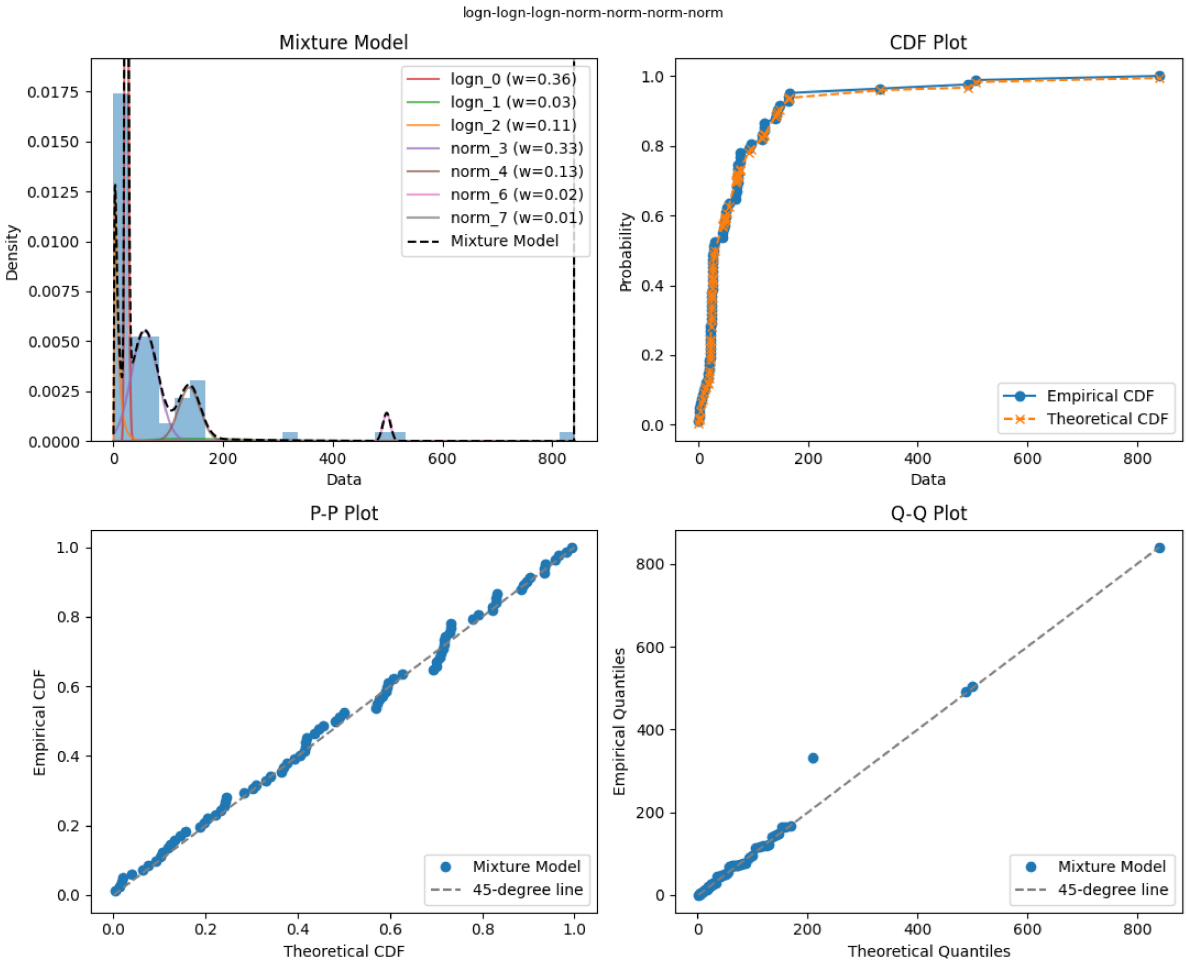

Figure 62 – G41345226...bba75907480_iexplore_e06d7363 Mixture Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

provide statistical evidence for the adequacy of the 8-component mixture model. The KS test yields a low test statistic (0.051) with an extremely high p-value (0.974), indicating that the null hypothesis of distributional equivalence cannot be rejected with confidence. The AD test, which is more sensitive to deviations in the distribution tails, produces a test statistic of -0.931 with a p-value of 0.940. This result further confirms the model adequacy.

The log-likelihood value of -367.89 reflects the model's overall fit and provides a basis for comparison with alternative models. In addition, the AIC and the BIC values are 783.78 and 841.54, respectively, these criteria penalize model complexity.

Having established the adequacy of the selected mixture model in terms of global fit, the next step is to examine the estimated parameters associated with each mixture component. Table 63 presents the estimated parameters of the fitted mixture model.

Each distribution follows its standard statistical parameterization, respectively:

- ❏ Normal distribution uses the location (μ) and scale (σ) parameters.
- ❏ Lognormal distribution is parameterized by the shape (σ), location (μ), and scale (θ) parameters.

The mixture weights indicate the relative importance of each failure subpopulation, with several dominant components: logn_0 (35.7%) and norm_3 (33.3%). These high-weight components correspond to different failure regimes, from rapid succession failures (logn_2 with scale parameter 6.009) to intermediate stability periods (norm_3 with location parameter 56.907).

The lognormal components (3 out of 8 components) reflect the multiplicative nature of the failure process. The presence of four normal components (norm_3, norm_4, norm_5, norm_6)

Table 62 – Goodness-of-fit Test Results: G41345226...bba75907480_iexplore_e06d7363.

Model	Test	Statistic / p-value
8-Component Mixture	KS	0.051 / 0.974
	AD	-0.931 / 0.940
	Log-Likelihood	-367.89
	AIC	783.78
	BIC	841.54

Table 63 – Mixture Model Parameters: G41345226...bba75907480_iexplore_e06d7363.

Component	Distribution	Weight	Param 1	Param 2	Param 3
logn_0	Lognormal	0.357	0.115	1E-10	23.673
logn_1	Lognormal	0.029	0.582	1E-10	184.591
logn_2	Lognormal	0.111	0.842	1E-10	6.009
norm_3	Normal	0.333	56.907	24.226	—
norm_4	Normal	0.134	138.317	19.846	—
norm_5	Normal	0.024	498.209	6.838	—
norm_6	Normal	0.012	840.576	1E-10	—

with varying location parameters (56.907, 138.317, 498.209, and 840.576 hours, respectively) captures the extended TBFs.

I.0.0.4 Sensitivity Analysis

To assess the robustness of the selected 8-component model and evaluate how model performance responds to changes in complexity, a sensitivity analysis was performed. This analysis investigated the behavior of GoF metrics as the number of mixture components increased. The sensitivity analysis was conducted by plotting the best GoF statistics obtained for each number of mixture components.

Figure 63 presents the sensitivity analysis results, examining the behavior of key goodness-of-fit statistics across component numbers. The KS test sensitivity analysis reveals a performance trajectory that supports the 8-component selection. Starting from a single-component performance, the statistic shows improvement through the initial component additions, dropping sharply for two components. The most significant improvement occurs in the 2-6 component range, with continued refinement leading to optimal performance in the 6-10 component

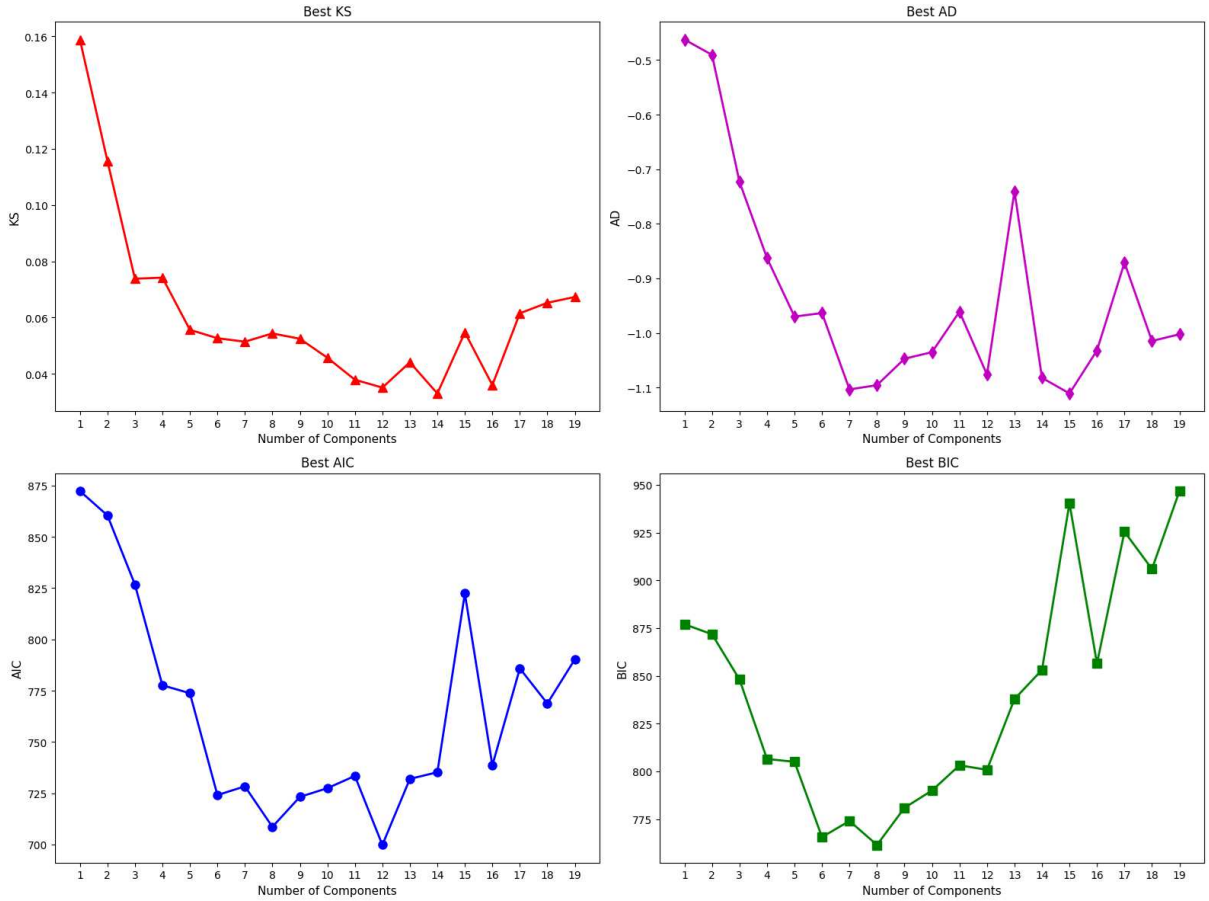

Figure 63 – G41345226...e87bba75907480_iexplore_e06d7363 Sensitivity Analysis of Model Selection Metrics (KS, AD, AIC, BIC) vs. Number of Mixture Components.

region.

The sensitivity of the AD test provides complementary evidence by emphasizing distribution's tail behavior. In the single-component model, the AD statistic highlights the inadequacy of simple models. As the number of components increases, the AD statistic transitions to negative values, indicating an improvement in the model's ability to represent distribution's tail behavior more accurately in multi-component configurations.

Information criteria sensitivity analysis reveals the trade-off inherent in model selection. The AIC trajectory shows improvement through 3-7 components, followed by a stagnation due to complexity penalties. The minimum AIC value was observed with 8 components. In contrast, the BIC analysis, which imposes a stronger penalty for model complexity, follows a similar initial trend but reaches its minimum around 6-8 components.

Additionally, graphical analyses of mixture model fits were conducted by inspecting the best-fitting result according to each individual GoF test. The distributions selected as optimal by each test: AD, KS, AIC, and BIC rarely coincided. In particular, the models selected by AIC and BIC consistently exhibited poorer visual and statistical fit compared to those chosen by the AD and KS tests. This performance gap persisted even in the cases where AIC and BIC selected models with a higher number of components.

I.0.0.5 Comparison with Single-Component Models

For comparison, the best-fitting single distribution was the lognormal model, as shown in Table 64. Although it achieved the best performance among the single-component models tested, it still failed to capture the complexity present in the data.

While the lognormal distribution emerged as the best-fitting single-component model among those evaluated, it remains inadequate for representing the complex behavior of time between failures. These models often fail to capture the underlying structural characteristics that arise from the inherent complexity of TBFs, which can result from multiple factors, including the diversity of failure causes and variations in workload and operational profiles.

Goodness-of-fit testing highlights this difference in representational adequacy: while the lognormal distribution yields higher test statistics for the Kolmogorov–Smirnov (KS = 0.134) and Anderson–Darling (AD = -0.207) tests, indicating inadequacy of the distributional hy-

Table 64 – Mixture vs. Single Distributions: G41...e87bba75907480_iexplore_e06d7363.

GOF Metric	Mixture Model (8-comp)	Lognormal Simple Distribution
KS / p-value	0.051 / 0.974	0.134 / 0.093
AD / p-value	-0.931 / 0.940	-0.207 / 0.451
Log-Likelihood	-367.89	-520.78
AIC	783.78	863.04
BIC	841.54	870.26

pothesis. The mixture model achieves much stronger agreement with the data (KS = 0.051, p = 0.974; AD = -0.931, p = 0.940).

The information criteria (AIC and BIC) further reinforce the inadequacy of the single distribution. As the lognormal model, despite its lower complexity, shows substantially higher AIC (863.04) and BIC (870.26) values compared to the mixture model, further reinforcing its inadequacy as a representation of the observed TBF behavior.

Figure 64 illustrates the limitations of single-distribution modeling for this failure dataset. The PDF overlay reveals that the lognormal model fails to represent the sharp peak near zero and completely misses the multimodal nature evident in the histogram.

The CDF comparison demonstrates significant deviations between the theoretical lognormal distribution (orange crosses) and the empirical distribution (blue circles), particularly in the lower tail region (0-100 hours) where the model underestimates failure probabilities, and in the upper tail where it overestimates them.

The P-P plot shows departures from linearity, with an S-shaped curve. The Q-Q plot reveals

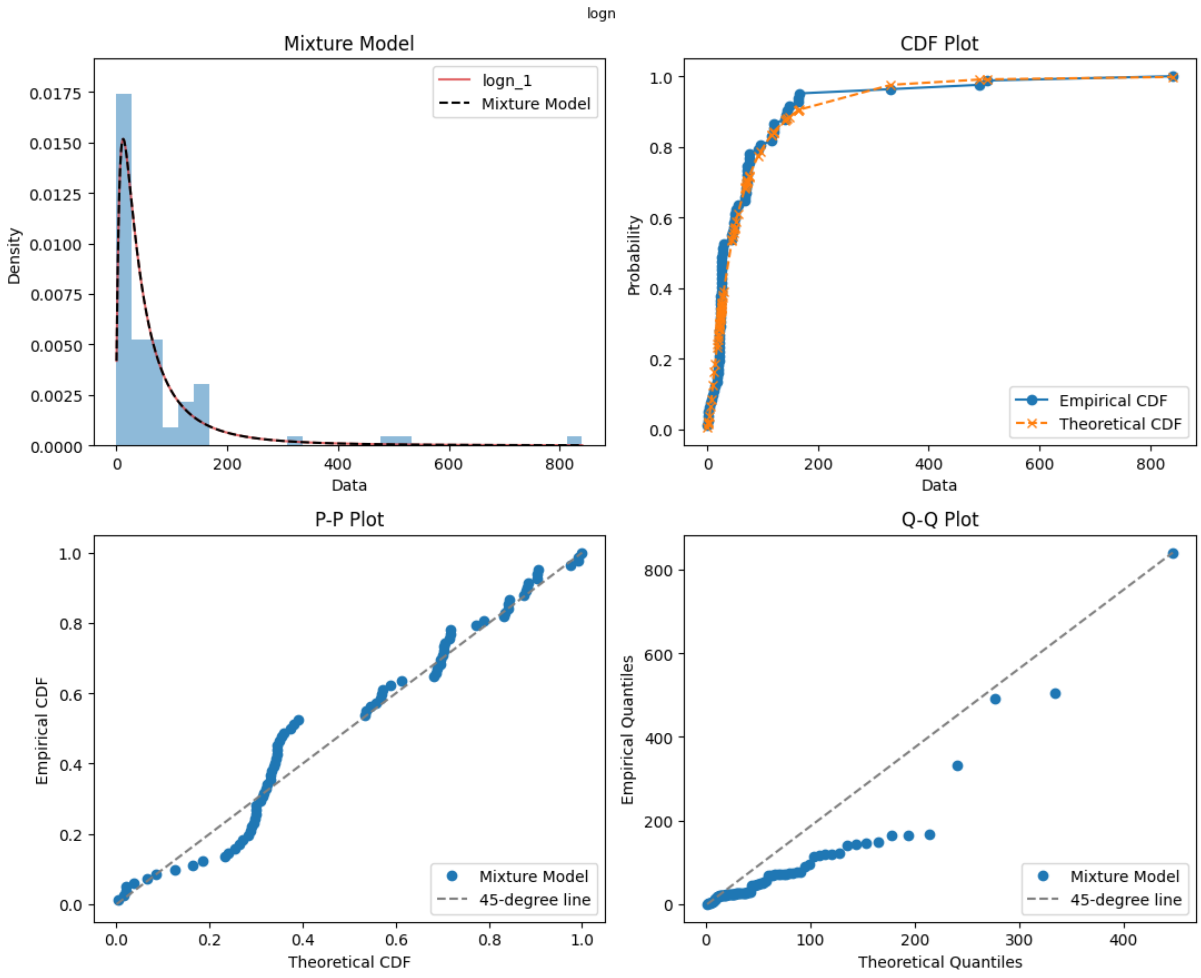

Figure 64 – G41345226...bba75907480_iexplore_e06d7363 Single Distribution Model Fit Demonstrated Through PDF, CDF, P-P and Q-Q Plots.

more nonlinearities, with the data points forming a curved pattern rather than following the expected 45-degree line.

I.0.0.6 Computational Cost Analysis

Given that mixture models can be computationally intensive, particularly when evaluating multiple distribution types and component numbers, a computational cost analysis was conducted. This analysis examines the trade-offs between model complexity and computational requirements.

Figure 65 and 66 present the computational cost analysis, showing execution time, memory usage, CPU utilization, and operation counts across different distribution types and component numbers.

Figure 65 presents the computational cost analysis for homogeneous mixture models across varying numbers of components. The analysis reveals distinct computational characteristics for different distribution families.

Figure 66 extends the analysis to heterogeneous mixture models, where components can

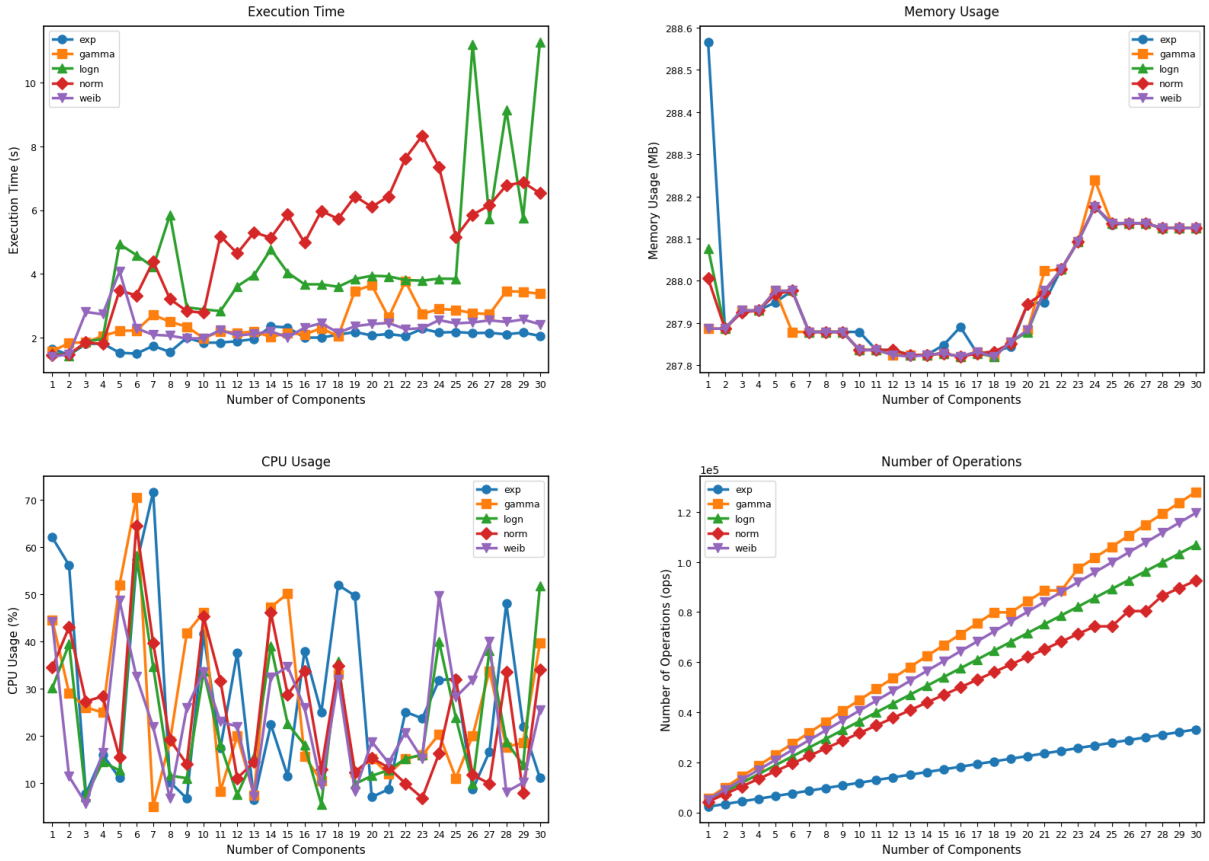

Figure 65 – G41345226...bba75907480_iexplore_e06d7363 Computational Cost Analysis of Homogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.

belong to different distribution families. This analysis is particularly relevant for the selected 8-component model, which combines lognormal and normal distributions.

The computational analysis reveals the scaling behavior of mixture model estimation as a function of both distribution complexity and the number of mixture components. These results inform practical considerations for large-scale reliability analysis applications. While the computational cost is substantial, the significant improvement in model adequacy and the critical importance of accurate failure modeling in reliability applications justify the computational investment.

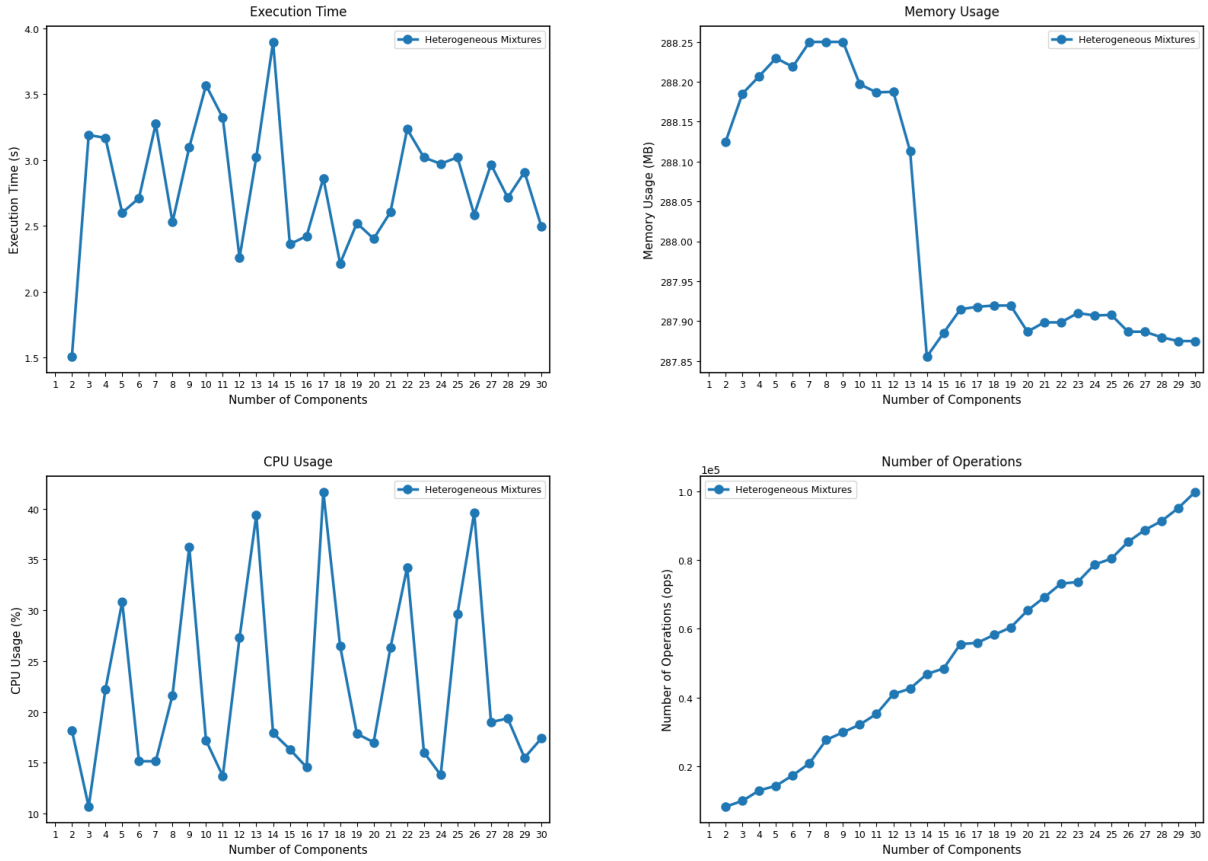

Figure 66 – G41345226...bba75907480_iexplore_e06d7363 Computational Cost Analysis of Heterogeneous Distribution Models with Execution Time, Memory Usage, CPU Utilization and Number of Operations.