UMass | Dartmouth UNIVERSITY OF MASSACHUSETTS DARTMOUTH

Software Failure and Reliability Assessment Tool (SFRAT): An Open Source Application for the Practitioner and Research Community

Vidhyashree Nagaraju[1], Thierry Wandji[2], and Lance Fiondella[1*]

[1]University of Massachusetts, North Dartmouth, MA 02747

[2]Naval Air Systems Command, Patuxent River, MD 20670

Motivation

- Many government organizations and national labs depend on mission and life critical software
  - To assure national security
  - Safety of human operators and communities in which software enabled systems reside

Reliability key to success of software

Motivation (2)

- Department of Defense (DoD) increasingly depends on software intensive systems
  - Mission and life critical
  - Must preserve high reliability and availability
- Urgency to deploy new technologies and military capabilities may result in
  - Inadequate reliability testing
  - Severe economic damage and loss of life

Background

- Recent National Academies report on Enhancing Defense System Reliability recommends
  - Use of reliability growth models to direct contractor design and test activities
- Several tools to
  - Automatically apply reliability models
  - Automate reliability test and evaluation

Existing Tools

- CASRE (Computer-Aided Software Reliability Estimation Tool)
  - Incorporates SMERFS (Statistical Modeling & Estimation of Reliability Functions for Software)
  - Automatically ranks models according to set of goodness of fit measures
- **Caution:** Users strongly advised to study underlying mathematics
  - Can better inform model selection process

Shortcomings of existing tools

- Over 20 years old
  - Not updated in over 15 years
  - Not compatible with current operating systems
- Interface
  - Does not impose intuitive workflow
  - Possible to run models on data that does not exhibit reliability improvement
  - User may fail to recognize lack of model fit
- Not open source (not sustainable)
  - Inhibits dialog between researchers and practitioners

Software Failure and Reliability Assessment Tool (SFRAT)

- SFRAT is an open source application
  - Designed for practitioner and research community
  - Programmed in R and provides functionality through a Shiny graphical user interface
- Reduces the need for knowledge of the underlying statistical techniques
  - Can help contractors quantitatively assess software as part of their data collection and reporting process

Software Failure and Reliability Assessment Tool (SFRAT) (2)

- Allows users to answer following questions about a software system during test
  1. Is the software ready to release (has it achieved a specified reliability goal)?
  2. How much more time and test effort will be required to achieve a specified goal?
  3. What will be the consequences to system's operational reliability if not enough testing resources are available?

SFRAT Output/Deliverables

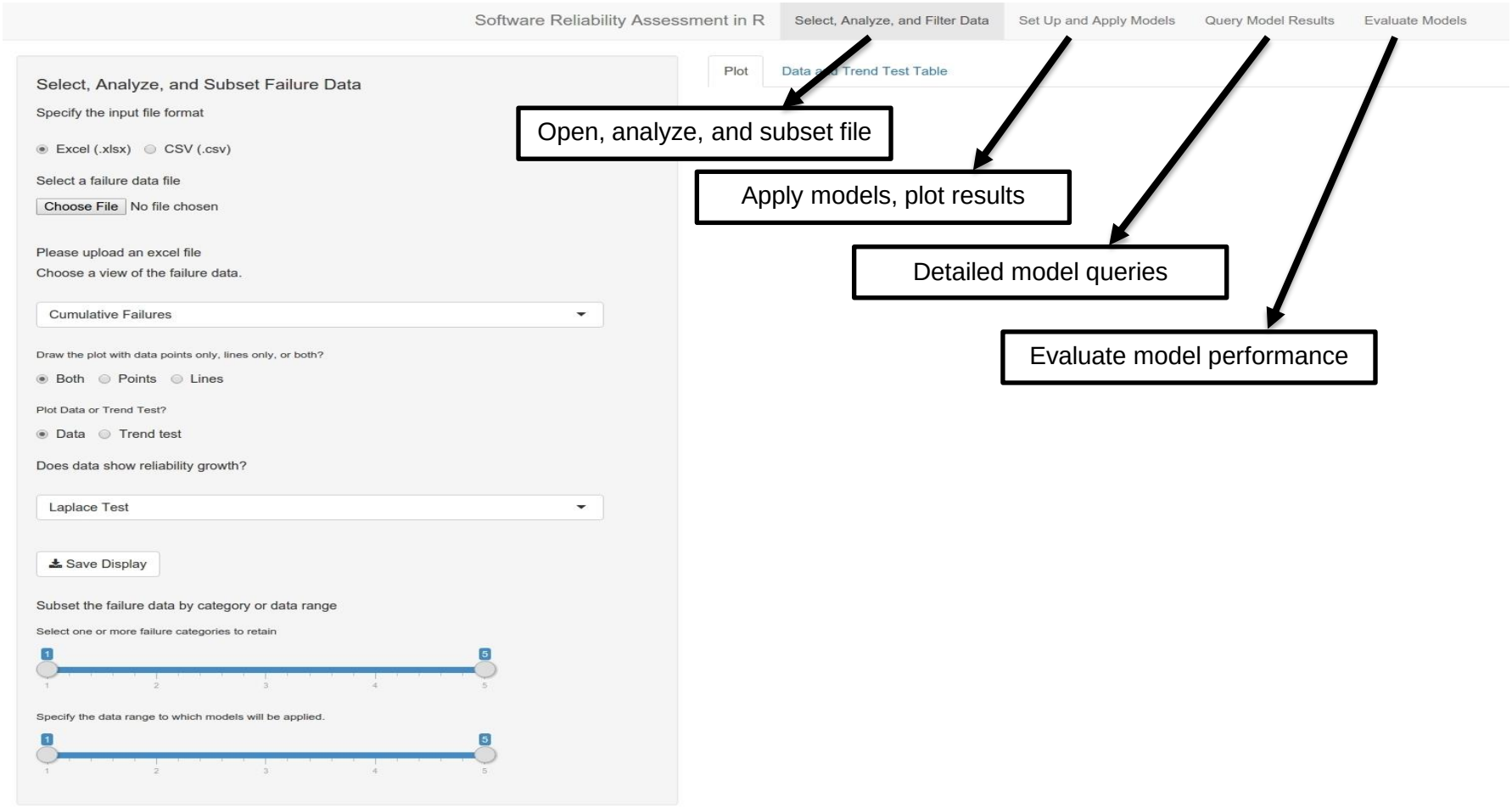
- Trend tests
- Model rankings
- Visualization
  - Cumulative failure plot
  - Time between failure plot
  - Failure intensity plot
  - Reliability growth plot
- Predictions
  - Time to achieve reliability
  - Expected number of faults for next t time units
  - Expected time to next k failures

SRGM classification

- Based on data formats
  - Failure Rate models
    - Inter-failure times - time between $(i-1)^{st}$ and i^{th} failure, defined as $t_i = (\mathbf{T}_i - \mathbf{T}_{i-1})$
    - Failure times – vector of failure times,
      $$\mathbf{T} =< t_1, t_2, \ldots, t_n >$$
  - Failure Counting models
    - Failure count data - length of the interval and number failures observed within it,

$$< \mathbf{T}, \mathbf{K} >=< (t_1, k_1), (t_2, k_2), \ldots, (t_n, k_n) >$$

SFRAT - Tab view



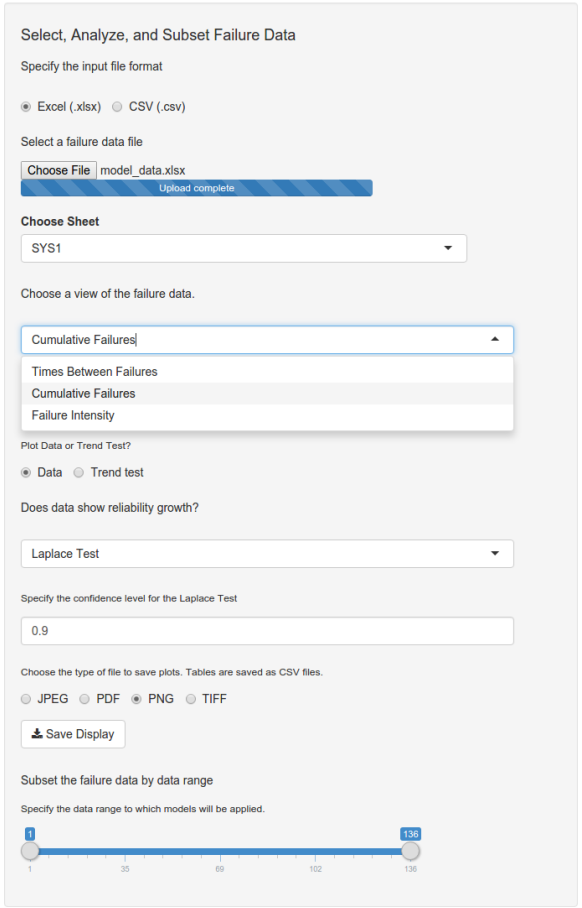
Input File Format

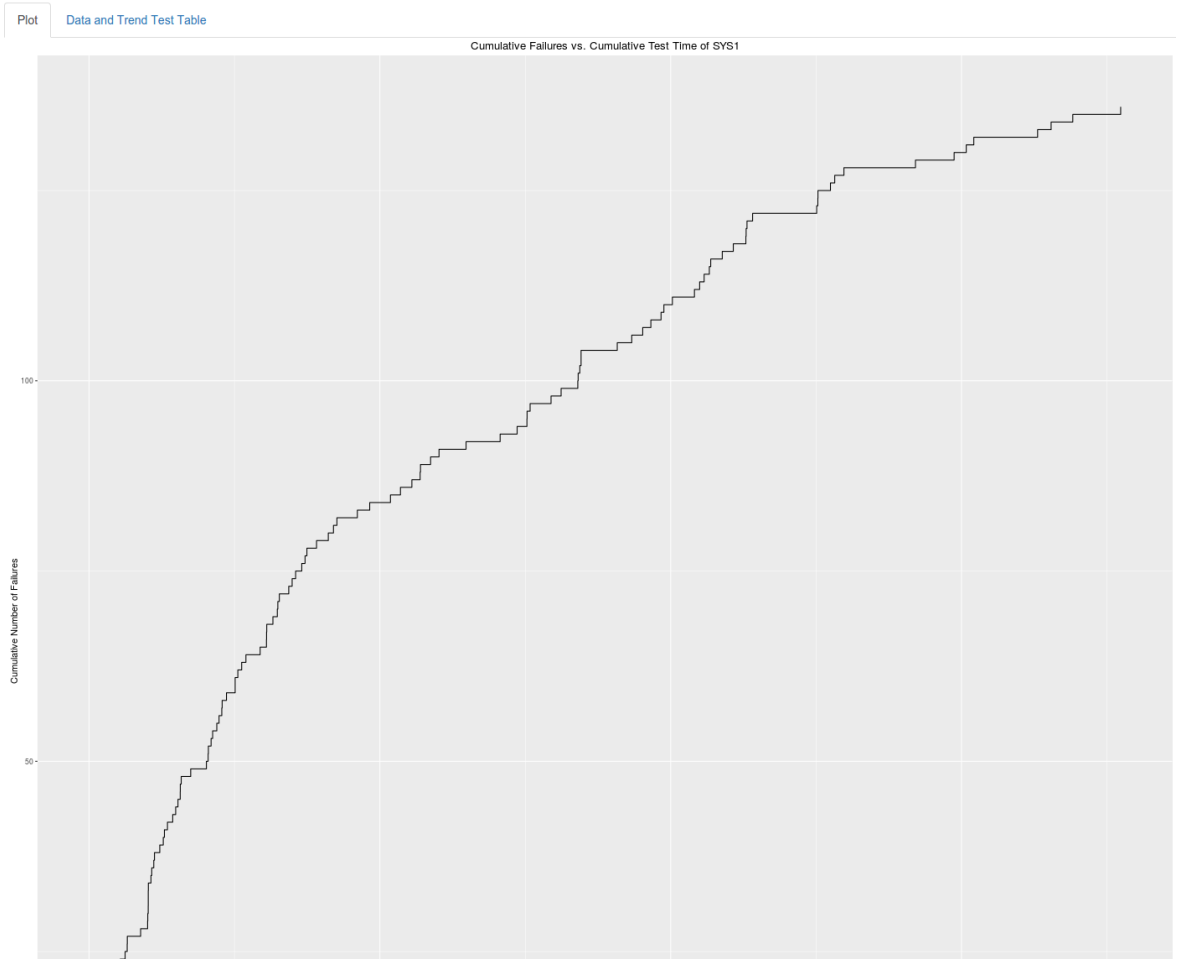
- Excel or csv
- First row indicates type of failure data
  - FN – Failure number
  - IF – Inter failure times
  - FT – Failure times
  - FC – Failure count
- Regardless of input format
  - Tool converts data to other two formats

1	FN	IF	FT
2	1	3	3
3	2	30	33
4	3	113	146
5	4	81	227
6	5	115	342
7	…	…	…

Tab 1
Select, Analyze, and Filter data

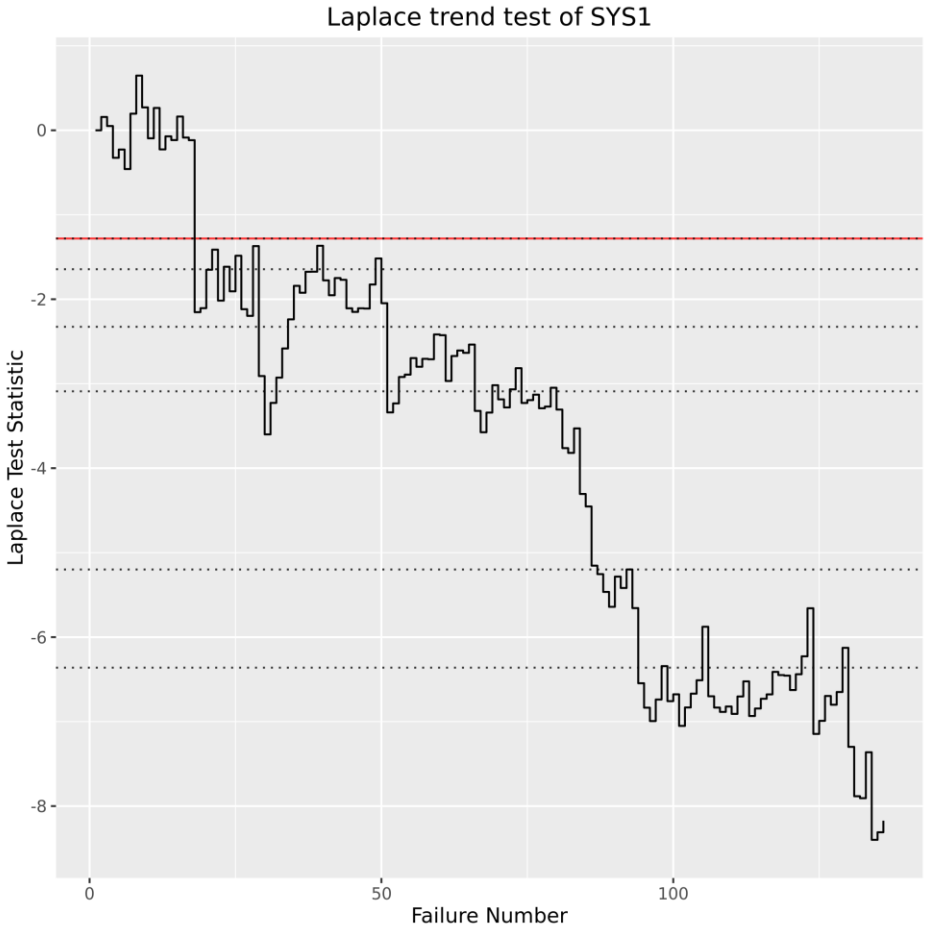
Tab 1 – After data upload





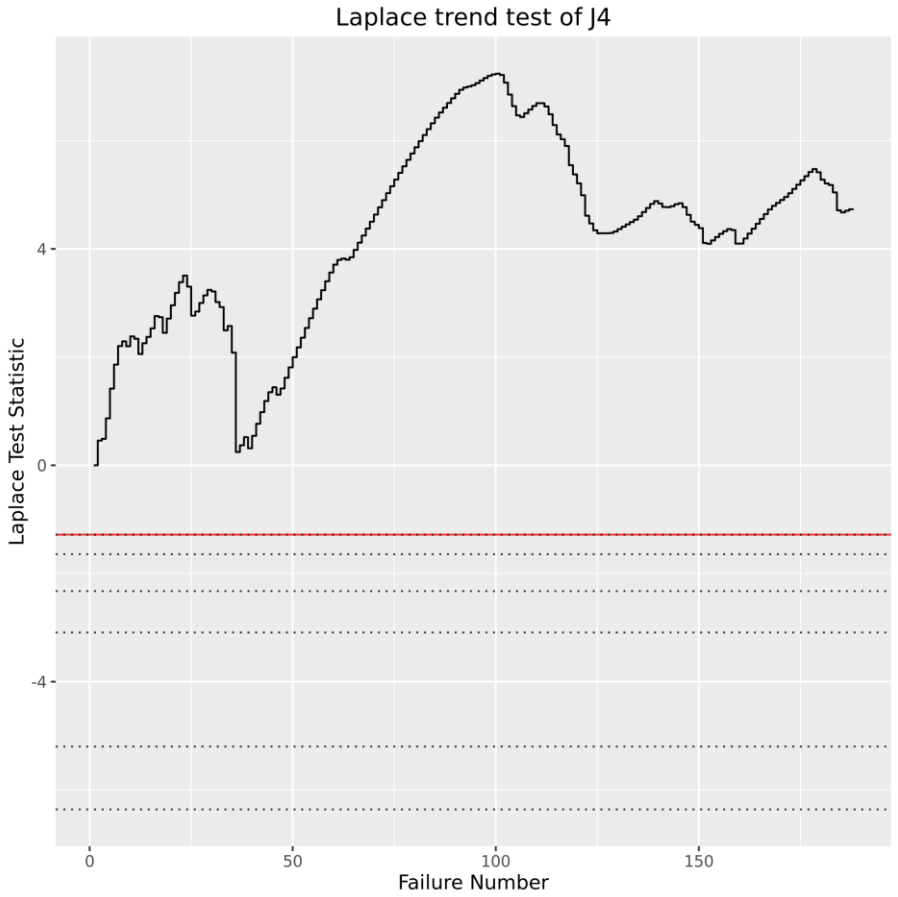
Cumulative failure data view

Laplace trend test – SYS1 data



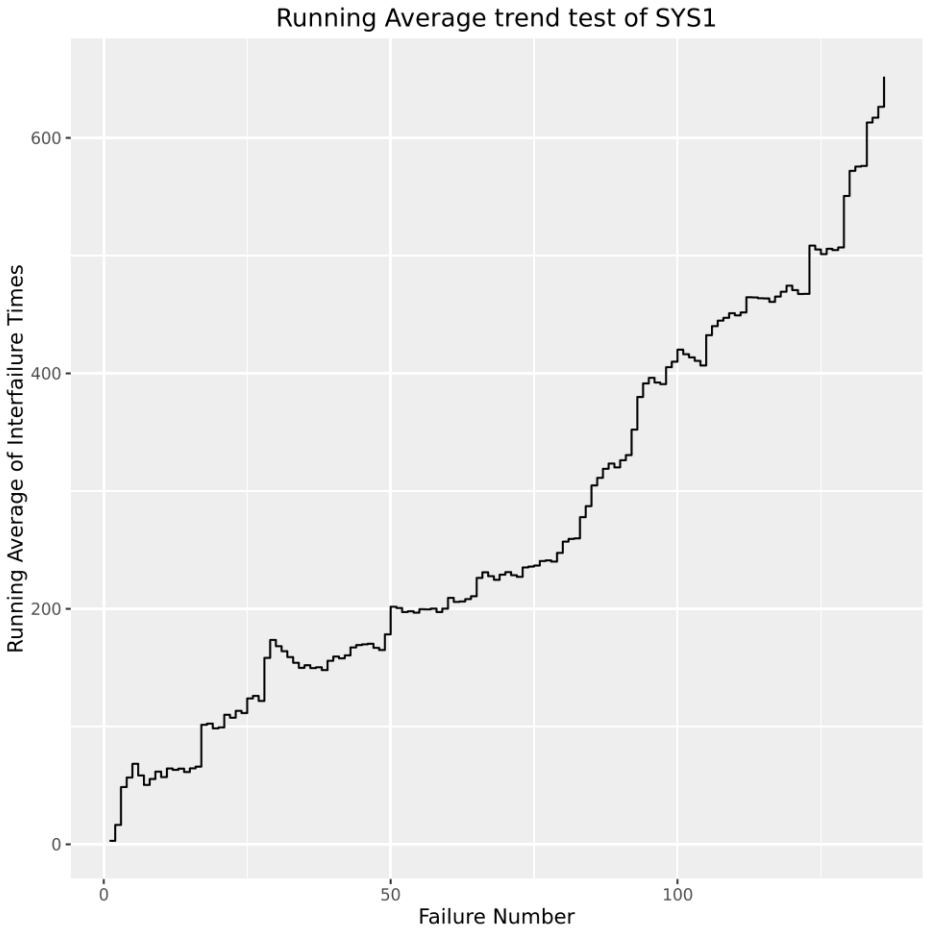
Decreasing trend indicates reliability growth
(application of SRGM appropriate)

Laplace trend test – J4 data



Does not exhibit reliability growth (additional testing required)

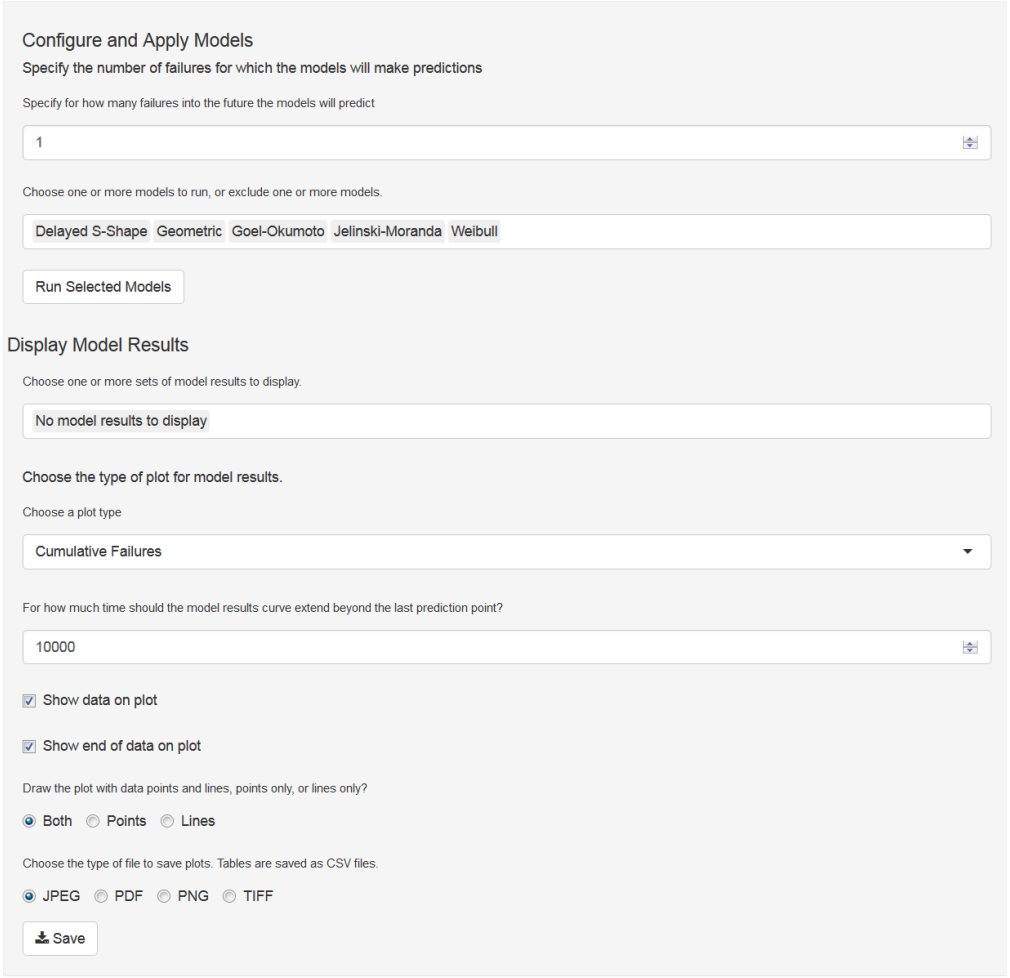
Running Arithmetic Average – SYS1 data



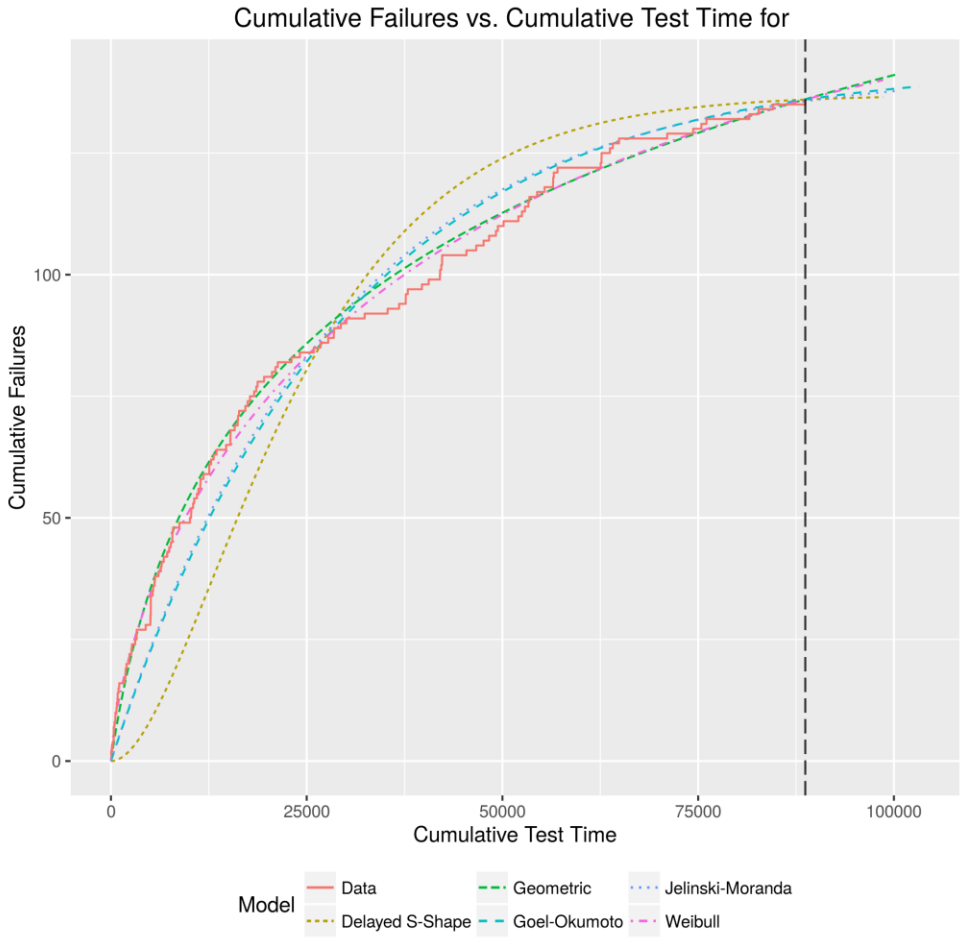
Increasing trend indicates reliability growth

Tab 2
Set Up and Apply Models

Tab 2 view

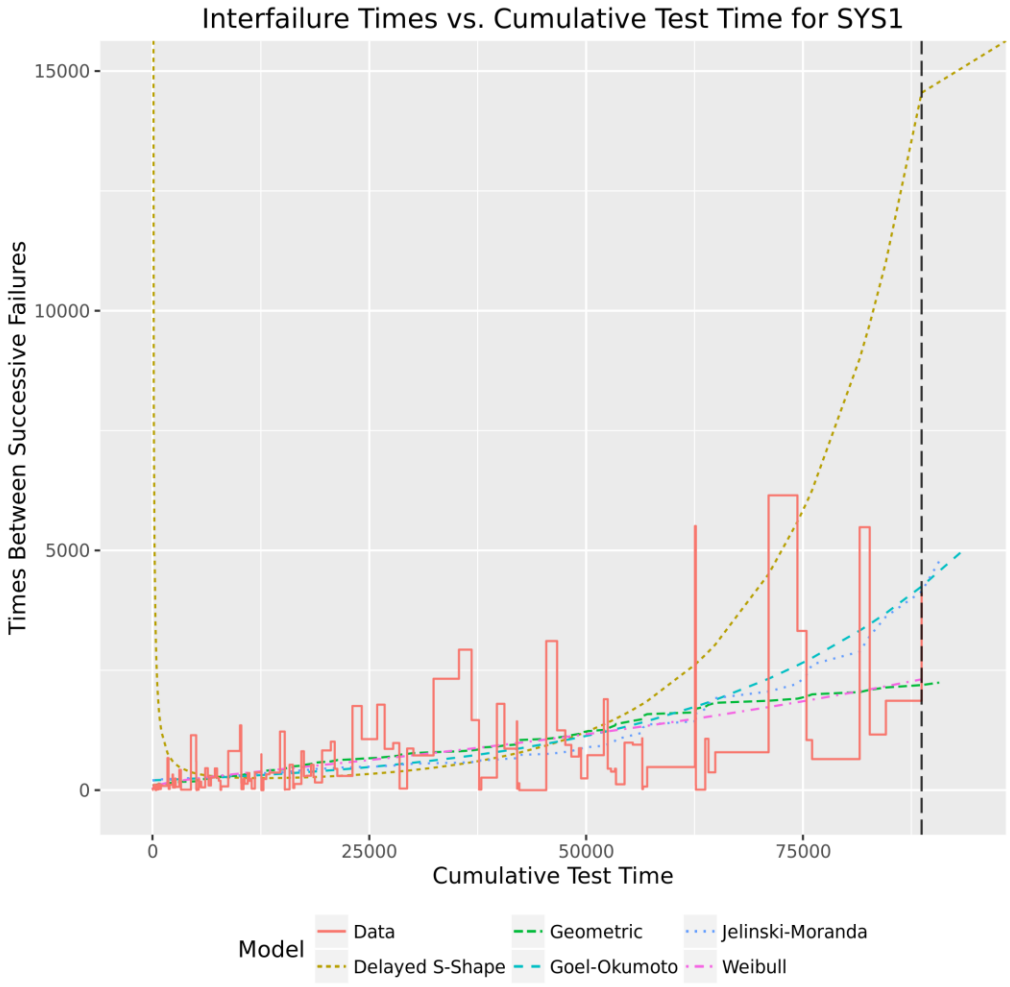

Configure and Apply Models
Specify the number of failures for which the models will make predictions
Specify for how many failures into the future the models will predict
1
Choose one or more models to run, or exclude one or more models.
Delayed S-Shape
Geometric
Goel-Okumoto
Jelinski-Moranda
Weibull
Run Selected Models
Display Model Results
Choose one or more sets of model results to display.
No model results to display
Choose the type of plot for model results.
Choose a plot type
Cumulative Failures
For how much time should the model results curve extend beyond the last prediction point?
10000
Show data on plot
Show end of data on plot
Draw the plot with data points and lines, points only, or lines only?
Both
Points
Lines
Choose the type of file to save plots. Tables are saved as CSV files.
JPEG
PDF
PNG
TIFF
Save

Cumulative failures



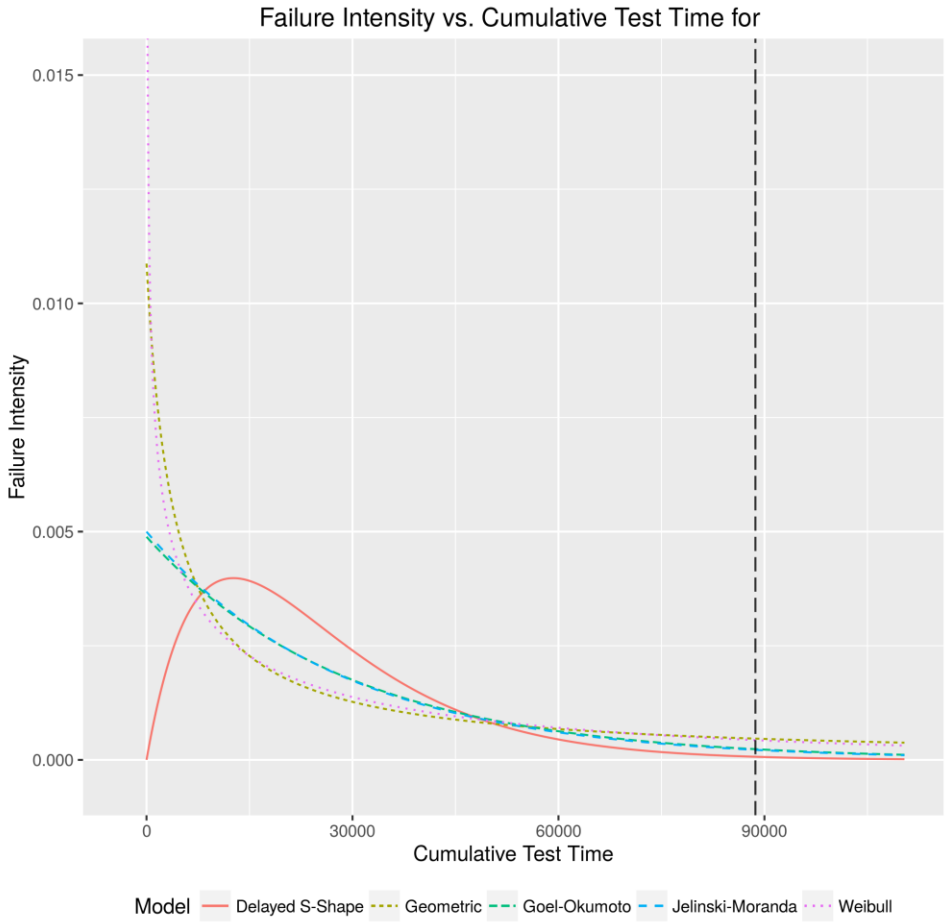
Plot enables comparison of data and model fits

Time between failures



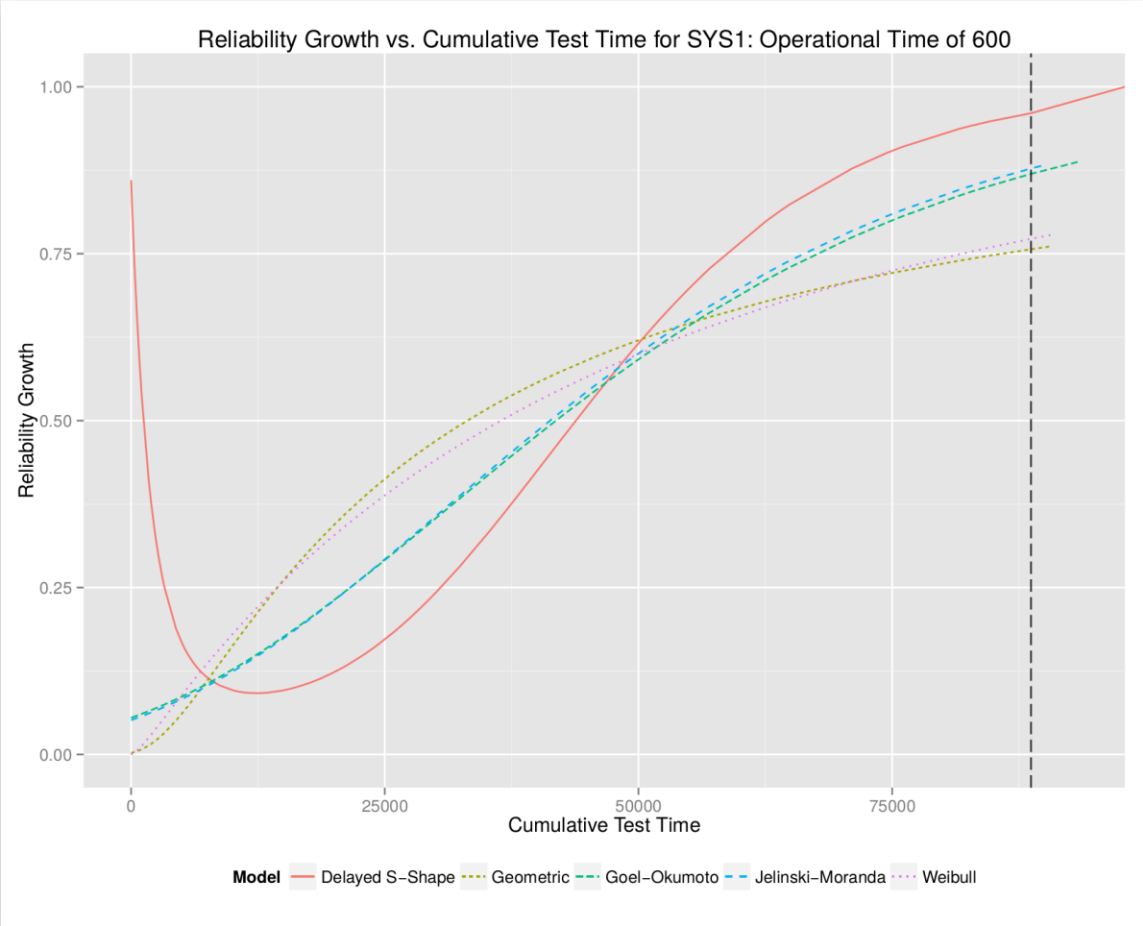
Times between failures should increase (indicates reliability growth)

Failure intensity



Failure intensity should decrease (indicates reliability growth)

Reliability growth curve



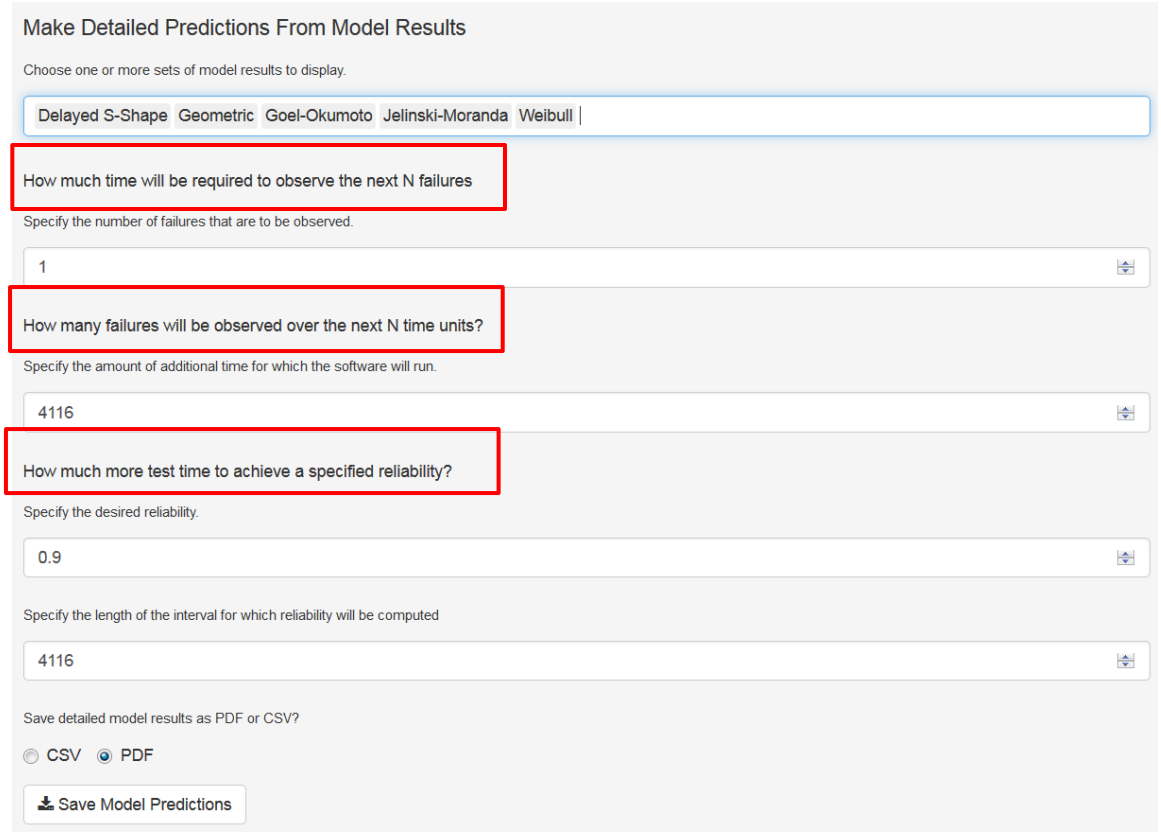
Can determine time to achieve target reliability

Tab 3
Query Model Results

Tab 3: Model predictions

- Allows users to answer the following questions
  1. How much time will be required to observe the next N failures
  2. How many failures will be observed over the next N time units?
  3. How much more test time to achieve a specified reliability?

Tab 3 Options



Failure Predictions

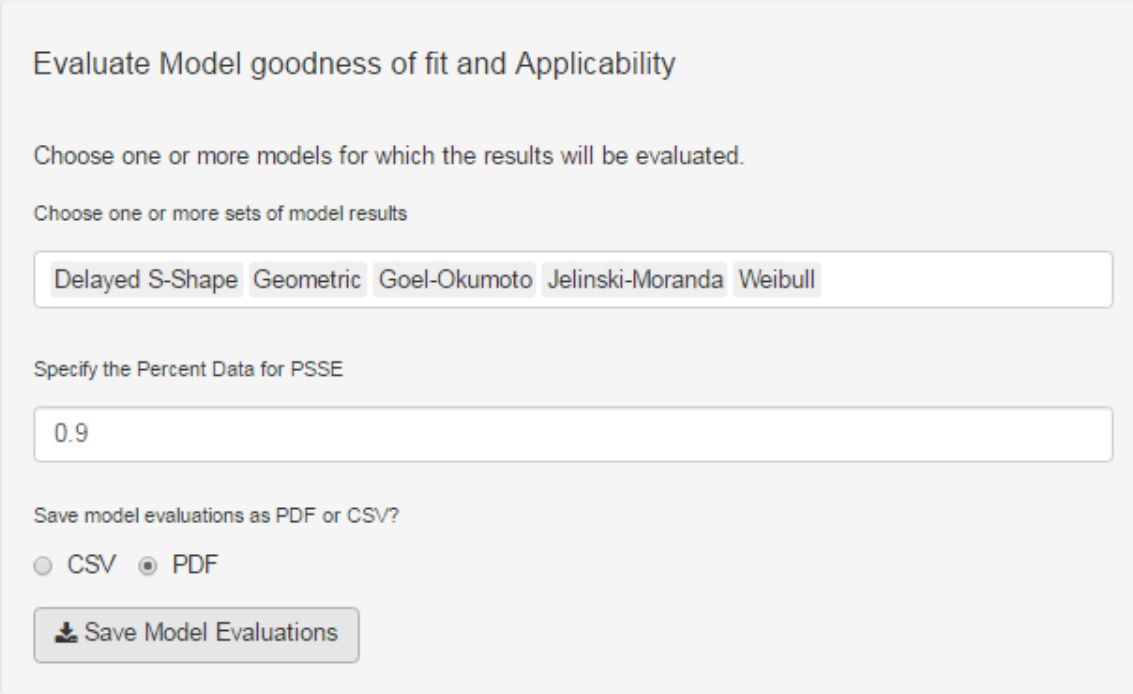
	Model	Time to achieve R = 0.9 for mission of length 4116	Expected # of failures for next 4116 time units	Nth failure	Expected times to next 1 failures
	All	All	All	All	All
1					
2	Delayed S-Shape	12401.1541529981	0.246856262199799	1	NA
3					
4	Goel-Okumoto	62829.7672027733	0.903615409906593	1	4591.28466949961
5					
6	Jelinski-Moranda	59915.2917457156	0.85612548252314	1	4869.80650205625
7					
8	Weibull	259865.770847692	1.72595369956707	1	2353.05254648438
9					
10	Geometric	1592716.45936287	1.87747308675807	1	2170.03088926781

Showing 1 to 10 of 10 entries

Previous 1 Next

Tab 4
Evaluate Models

Tab 4 Options



Model assessment based on AIC and PSSE

AIC and PSSE

	Model	AIC	PSSE
	All	All	All
1	Delayed S-Shape	2075.14631533222	296.349252292955
2	Geometric	1937.03417425106	84.3270812346017
3	Goel-Okumoto	1953.61306630984	23.0712869112105
4	Jelinski-Moranda	1950.53413167956	19.6003726994455
5	Weibull	1938.16066975807	74.9449562450499

Showing 1 to 5 of 5 entries

Previous 1 Next

Lower values preferred

Conclusions and Future Research

- Presented open source application to promote collaboration among
  - Members of software reliability research community
  - Users from industry and government organizations
- Application architecture enables integration of models from research literature
- Future research will expand architecture to enable models for other stages of SDLC

Software Reliability Tool

Available online

http://sasdlc.org/lab

Acknowledgement

- This work was partially supported by (i) the Naval Air Warfare Center (NAVAIR) under Award Number N00421-16-P-0521 and (ii) the National Science Foundation (NSF) (#1526128).