GraceFall: Visualizer for Diverse Wear-Out Reliability Degradation Data Spanning Multiple Time Scales

Matthew Tang and Ian Hill

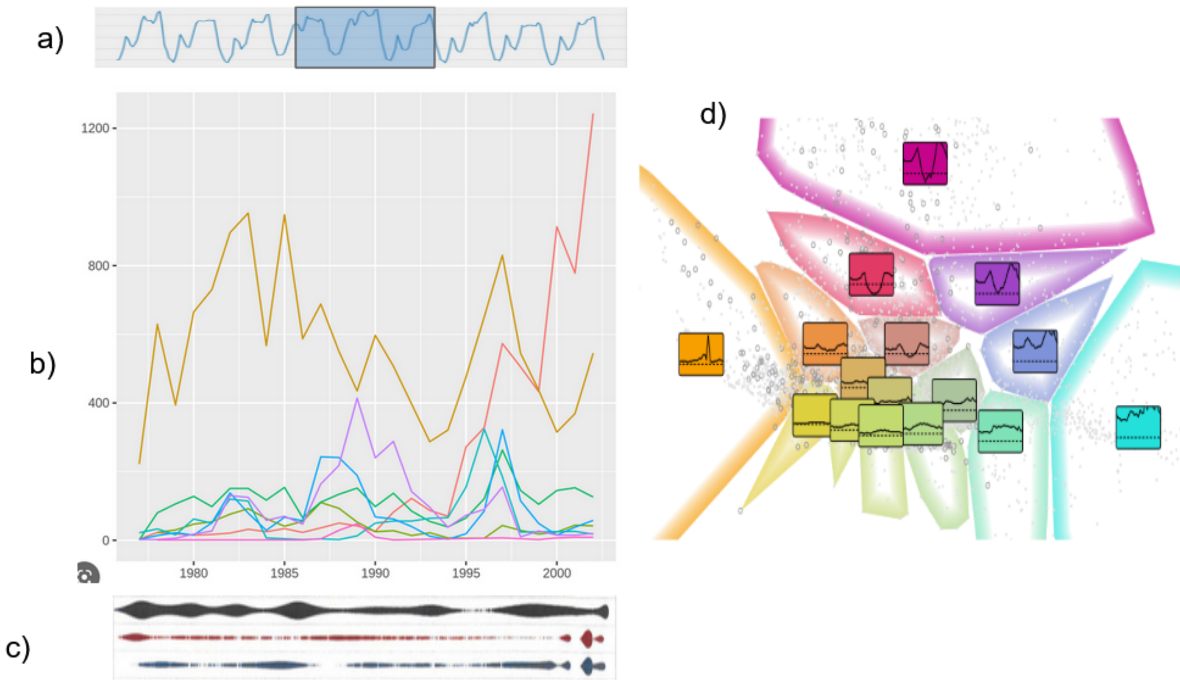

Fig. 1. Moc of proposed visualization. All details can be found in section 4. **a)** The time summary plot for satisfying **R1.b**. **b)** The Group Line Plot. **c)** The three panes satisfying **R2**. **d)** the PCA Distance Plot

Index Terms—Multiple time scale, many series summary, integrated circuit reliability.

1 Introduction

Long-term reliability is a niche but critical aspect of modern electronics design. Testing and planning for degradation and failures is incredibly challenging; nearly all product development cycles are much shorter than the required lifespans and so accelerated stress testing followed by extrapolation/prediction is leveraged in lieu of direct testing. These testing practices are the core focus of one author's (Ian's) doctoral thesis topic, and several shortcomings of existing reliability engineering practices were identified in a survey paper published in IEEE Transactions on Device and Materials Reliability [5]. A simulator has already been developed that enables stochastic temporal simulation of wear-out processes and Bayesian inference on wear-out models to address two of these shortcomings.

Due to the model-agnostic implementation and probabilistic focus of the developed simulator, visualizing the simulated stress, measurement data, and specified stochastic models in a way that supports the variety of user tasks enabled by the simulator presents a significant challenge. This project develops a visualization tool to aid reliability engineers carry out multiple user tasks that can be tackled using the simulator in an effort to overcome the conceptual challenges of non-constant stress and probabilistic models that engineers are not likely to be accustomed to. Key tasks include iterative stress test design, test data failure analysis, and model quality comparisons.

Our preliminary investigation found that in all three use cases engineers will need to compare large quantities of scalar data series under multiple stress configurations through time. The proposed solution, GraceFall, handles diverse line graphs with dynamic time scales to aid in these exploratory analysis tasks. The proposed visualization will integrate with the existing simulator and thus be primarily developed with Python.

2 Related Work

There are two types of related works to discuss in order to place this project in context, those within the integrated circuit wear-out reliability space, and then those within the field of info visualization.

2.1 Integrated Circuit Reliability Visualizations

Integrated circuit reliability is not well known for effective visualization tools, as the field is reasonably niche and driven by individual companies within the semiconductor industries. Reliability reporting is mostly opaque to end users of integrated circuit products, thus visual representations of reliability and reliability tests are often neglected. Typically, the visualizations that do get constructed are of poor quality. Common visual representations of reliability test data include Weibull plots of cumulative failures against time 2, or box plots showing degradation distributions for different devices 3.

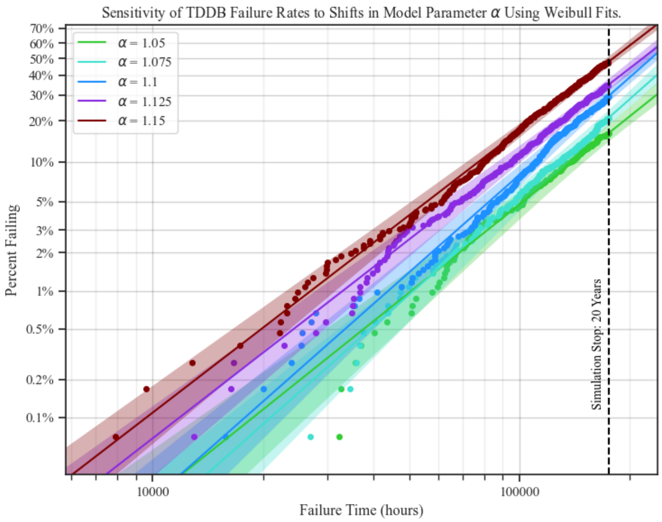

Fig. 2. Sample Weibull plot from Ian's recent conference submission. Do not distribute.

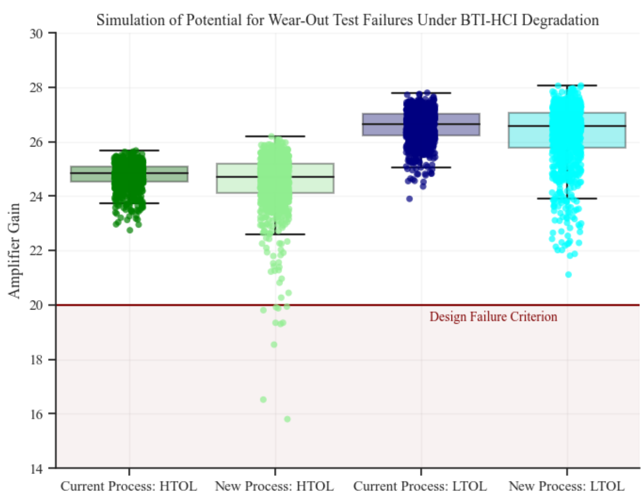

Fig. 3. Sample end-of-test degradation boxplots from Ian's recent conference submission. Do not distribute.

We believe a key missing element towards effective understanding of wear-out processes in existing reliability visualizations is temporal comparison. We hypothesize that there are three main reasons why temporal visualizations of wear-out are so rarely seen: (1) collecting measurements during stress is costly and thus rarely is sufficient data available for meaningful trends to be evident within the data; (2) product reliability requirements are typically based on failure rates at some time instant, thus results are often only displayed for that point; and (3) visualizing the evolution of derived statistical quantities through time is challenging from a design perspective as many channels are needed and large sample sizes are prone to visual clutter. The first two reasons for rarity are not applicable in the case of the developed simulator, which imposes no measurement costs and is intended for developing models and tests, not for evaluating product reliability at arbitrary end-of-supported-life instants.

Some additional existing wear-out visualizations published by one of the most recognizable researchers in the field are shown in Figure 4, highlighting the prevalence of visual clutter, overuse of contextual labelling to frame data, and minimal consideration of how degradation and/or failures evolve over time [13]. Interested readers can find even more sample visualizations in [14, 18]. For Ian's research, visual representations are needed to eventually help engineers explore test design spaces and compare different probabilistic models. The existing visualizations such as those shown are not effective for design space exploration tasks within this domain as they do not provide little to no information on the stress resulting in wear-out or the models used to produce or fit degradation and/or failure data.

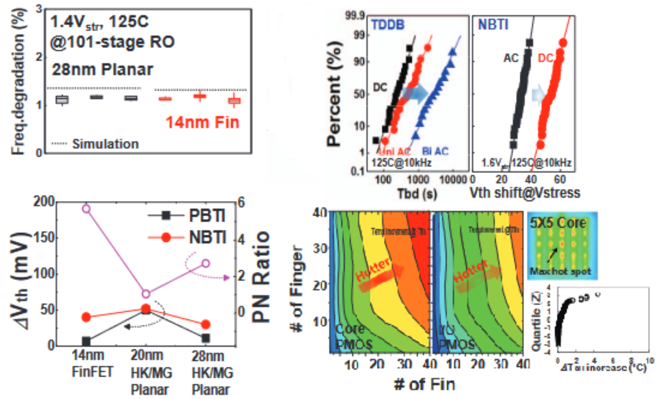

Fig. 4. Examples of accepted visualizations within the IC reliability research community from a well-regarded paper [13]

2.2 Relevant Information Visualization Design Studies and Techniques

Moving on to how this project fits within data visualization research, we are primarily focused on the application of visualization techniques as opposed to the development of novel ones. With this in mind, our project is suitably described as a design study, and can be positioned relative to existing similar design projects of larger scope, namely KD-Box [20] and Vismon [3]. KD-Box is a method that addresses line cluttering of plotting many time-series by finding a representative line using density. It is useful as a comparative study because it addresses similar problems of visualizing time-series data from sensors where many independent data series need to be summarized or reduced to extract useful information about the collective. Vismon is a interactive visualization tool allowing fishery scientist to explore hypothesis about their models through sensitivity and constraint-based analysis. While Vismon does not consider time series plots, its treatment of model comparison across multiple views and incorporation of information uncertainty is comparable to the design components encountered further into the proposed project.

Additionally, specific design challenges within our project have been explored by previous works. Although our project allocates significant time for more detailed literature review, we have already looked at some initial solutions for overcoming anticipated design challenges:

1. Multiple time scale plots introduce challenges when visualization users are interested in both long-term summaries of temporal data and in the more detailed structure of the same data when considered over a short interval. Multiple techniques have been introduced to in addressing this problem. Many tackle this problem by showing a coarse to fine time scale approach. Time calendar [17] representation provides a month scale summary is provided by coloring the days on the calendar with a day-scale view on the side. Time-Series-Path [2] provide a squeezed colorized bar summarizing the the property of the time series with color while the detailed color representation is directly plotted for viewing. "Chronolensing" [19] allows interactivity enabling users to choose desired their time scale view.

2. There are much work on plotting many time series. We will focus on only ones that are relevant here. A general summary can be found in [1]. Many works attempts to directly plot the time series and reduce the cluttering afterwards. This class of methods include the use of density [12], grouping [10], and ordered plotting [16] to make the plots easier to view. However, by doing so, the visualization loses the fine resolution of individual time series and, hence, difficult to detect anomalies from them. Other works that maps each time series to points 2D space [9] [15] can see the difference clearly, however, the dimension loses all semantic meaning.

3. Users of the visualization are likely to want to compare different tests or data sets in relation to one another, introducing a significant challenge to effectively lay out multiple test plots within a single view frame and navigate them. [3] can inform some initial design decisions for this project aspect.

4. Visualizing mathematical models is an interesting challenge as they don't fit as cleanly within standard visualization conceptual frameworks. The data object being shown is the output of an expression that is variable, making the data dynamic when dependent on other visualized data. A 2008 paper on plotting models based on parameter selections is considered as a useful starting point for this component of the project [11].

3 Data and Task Abstraction

A core problem reliability engineers encounter when analyzing a set of test results is a poor understanding of how observed component degradation depends on multiple induced stressors, especially as the models used to explain physical degradation are incomplete and rely on numerous physical assumptions. To understand how products degrade and fail as a result of stress or to evaluate how well a model captures complex physical phenomenon, engineers need to view the effects of wear-out processes as a function of time. Without existing visualization tools, engineers frequently need to compare multiple test results in the form of pure numerical tables, a notoriously difficult data form to analyze from. We introduce GraceFall to aid engineers in exploratory analysis of temporal information available in wear-out test/measured data, both simulated and real.

3.1 Data Set

Prior to dissecting the overarching visualization objective, it is necessary to first present the specific data available. There are three major data sets relevant to our visualization which we describe at an abstracted level.

1. Tabular data listing values associated with several attributes. This is long-form data with multi-attribute identification needed to determine the value "source", along with an independent time attribute. Specifically, each quantitative value is a measured quantity associated with some quantitative time instant and categorical sample source. A sample source is uniquely identified by the combination of four attributes: type, circuit number, chip number, and lot number. Each of these attributes are individually non-unique, and sample sources sharing the same chip number and/or lot number field values will have potentially correlated values. If the tabular data is separated by unique sample source, each separable set of values will represent a single data series of measured values at different instants in time. The quantity of sample sources, samples per source, and value ranges are fully unbounded, however will be restricted within the scope of the GraceFall design requirements.

2. Tabular data listing values associated with a unique time interval. Each row entry has an attribute for the quantitative value of each applied stressor along with three quantitative attributes for start time, end time, and duration of the unique time interval. A final categorical attribute indicates the name of the stress interval but is out of the scope of the GraceFall tool. The quantity of stressor attributes, stressed time intervals, and value ranges are once again unbounded but restricted within the scope of GraceFall.

3. Hypothesized mathematical models that can be fit to the measurement data. These are mathematical expressions attempting to explain the test results as fitted functions of the stress data. To keep the project in a manageable scope, this project will restrict the data to deterministic models as opposed to the eventual desire to additionally support probabilistic ones. As this data type does not fit into the framework introduced in the CPSC 547 course, the visualized data for a fitted model will be simply represented as sets of quantitative points.

3.2 Task Specification

To better analyze the visualization design problem of GraceFall we use a generalized framing, thus the expected user tasks have been abstracted and are summarized in Table 1.

Based on these tasks, design requirements have been developed that will be used to inform the design decisions made as part of the development of GraceFall. Early mock-ups identified several design challenges that require the evaluation and selection of sophisticated techniques to overcome.

1. **C1** - Handling of clutter caused by many lines per graph due to large quantities of data series on multiple time scales of interest.

2. **C2** Users of the visualization are likely to want to compare different tests or data sets in relation to one another, it is challenging to effectively lay out multiple test plots with potentially different features and navigate them.

3. **C3** - Visualizing mathematical models is an interesting challenge as they don't fit as cleanly within standard visualization conceptual frameworks.

4. **C4** - Multiple time scale plots introduce challenges when visualization users are interested in both long-term summaries of temporal data and in the more detailed structure of the same data when considered over a short interval.

4 Proposed Solution

The developed requirements for GraceFall used to implement the identified tasks and tackle the anticipated design challenges are shown in Table 2.

4.1 Design Technique Decision Process

Design decisions are in progress; those that have been made will be discussed here. To contextualize the following discussion, a mock-up with the selected design features in shown in Figure 1. First, to address **R1.c** in a way that mitigates the clutter risk discussed in challenge **C1**, two view panes will be used for a given data set: a group line plot and a PCA distance plot.

Group Line Plot The primary choice of visualizing the time-series will be standard direct line plots as required in **R1**. Other visualization techniques that conduct major data transform on the time-series data, such as those utilizing PCA will not work, because it is of critical importance for the user to see the detailed evolution of slope and aggregate behaviour as time progresses. To allow users to easily identify trends, k-means clustering will be used to meet **R1.e** and **R1.a**, and will partially overcome **C1**, cluster-colouring time series with similar properties as in [4, 7].

PCA Distance Plot Clustering by itself is insufficient as direct line plot reduces the much needed resolution of the time series to find points of interest. To address this, PCA reduction will be used to map the time series down to a 2D point. Those points will be assigned colours as in the k-means approach. These points can better identify outliers as demonstrated by [15]. Further, when users pan over a region with the same color, the mean time series will appear above the region to give the user a sense of the underlying data. Additionally, within

Table 1. Abstracted task definitions for GraceFall visualization tool.

Task ID	Priority	Task Definition
T1	Mandatory	Examine the statistical distribution evolution of multiple data series as time progresses
T1.a	Important	Find identifiable groups that show similar behaviour within the larger set
T1.b	Mandatory	Roughly track sample variance and likelihood region evolution, identify asymmetries
T1.c	Important	Catch outlier samples and propose probable causes for the outliers using background/meta data
T2	Mandatory	Search for probable values in background/meta data that result in large changes to data series trends in time
T3	Important	Evaluate the success/performance of proposed models in explaining the observed data series
T4	Nice to Have	Compare multiple different sets of background/meta data and their corresponding sets of data series in terms of their influence on the outcomes of the previous three tasks

Table 2. GraceFall design requirements.

ID	Priority	Requirement Definition
R1	Mandatory	Tool must display multiple data series against time
R1.a	Important	Up to five sets of data series can be visualized on the same axes
R1.b	Mandatory	Can change focus between multiple time scales of interest on demand
R1.c	Mandatory	Up to 1000 data series must be individually viewable simultaneously
R1.d	Nice to Have	Aggregate properties of sets of data series can be overlaid or substituted for raw data on demand
R1.e	Nice to Have	Provide automated classification of data series into groupings by similarity
R2	Mandatory	Background/meta data is displayed alongside data series to maximize spatial ”closeness” / minimize cognitive effort required to jump between viewing data series and meta data values
R2.a	Mandatory	Up to three meta data fields must be displayable simultaneously
R3	Important	Proposal model outputs, dependent on background/meta data values, are displayable on the data series axes on demand
R3.a	Important	Must be immediately visually distinguishable from displayed data series or statistical properties of the data series
R4	Nice to Have	Up to four data series axes views can be displayed simultaneously, arranged to maximize spatial ”closeness” / minimize cognitive effort required to jump between viewing different axes
R4.a	Nice to Have	Significant differences in background/meta data values between views can be highlighted on demand

this view, the user can select and filter only the time series they are of interest to show on the group line plot, allowing them to explore the data they are only interested in.

To address **R1.b**, a way of navigating multiple time scale is required. Although multiple techniques in the literature allow for this functionality, lensing similar to that in [19] was selected for one key reason. The other approaches mentioned in the Related Work section are discrete, limiting the scales at which the user can explore the data. For wear-out test data, the expected time scales of interest are not predefined and will have to be fine-tuned depending on the test conducted and user task. Allowing for continuously-defined time scales to be selected is thus valuable to prioritize. To ease lensing navigation, a squeeze overview of the full time interval will be provided. The user can select an area that they wish to explore. The selected sequence will appear in the Group Line Plot for exploration.

For requirement **R2**, the user needs to see stress data values with ease to identify causes of trends in the primary data series. To satisfy **R2**, a three long panes similar to [8] with small balls corresponding to the magnitude of stress parameters will be shown on the bottom of the Group Line Plot. This allows the user to compare a maximum of three stress parameters at a time, satisfying **R2.a**. Compared to alternative channels such as color, the eye can see the magnitude difference between size of balls with improved accuracy and precision, especially as the vertical positions are aligned, effectively allowing for length comparisons. This approach was compared to directly showing bar charts when hovering over the points similar to the box plots in [2], however that approach introduced occlusion and expected comparison difficulties when displaying the values of three stressors simultaneously.

Requirements **R3** and **R4** are stretch goals, and so the design that meets them are not yet finalized. The current plan of record is to reserve black as the color representing a proposed model. Users can provide a candidate model that will be graphed onto the Group Line Plot for comparison.

4.2 Implementation

At time of writing, the design of GraceFall is still in some flux and so technique implementation has not yet begun. To allow for rapid design implementation, however, the selection of programming framework, test data set construction, and boilerplate code implementation are well underway. Initial data sets were generated through the existing wear-out reliability simulator and attempt to encapsulate a variety of different data sets with interesting features (e.g., subgroupings within the data, unique stress tests, different and/or multiple types of degraded values). Approximately five data sets will be used for the verification and validation phase of GraceFall's development, and the data sets will be tuned as needed to introduce further interesting data features to discover as part of the visualization tool tasks.

For the software implementation of GraceFall, the project was constrained to a Python-based toolset for ease of compatibility with the existing wear-out reliability simulator and based on our prior familiarity with the language. Five interactive visualization libraries/frameworks were considered: Plotly, Bokeh, Mpld3, Gleam, and Vega-Altair. After feature comparison among other criteria, Vega-Altair was selected for implementing GraceFall. Although Plotly offered slightly more customized design and additional features, it was determined to be much more challenging to use, and Vega-Altair was a close second in terms of feature-rich visualization design. An additional benefit of using Vega-Altair is that it follows the Vega-Lite specification which follows and extremely similar design space specification to that presented in the CPSC 547 course, providing a valuable opportunity to work with a tool that helps reinforce the learned concepts from classes.

With development data and the visualization framework available, boilerplate to structure the GraceFall code base and outline the expected execution flow has been written, and is available on GitHub [6]. As of now, the GraceFall program can read in the data structures from a standard format for use, and has a 'wishful programming' functional interface for the expected key steps in building a specific visualization view (e.g., constructing data series view, overlay of stress summary view, PCA construction, multiple test view arrangement). A quick demonstration visualization was developed, shown in Figure 5. Note that this visualization does not implement the planned visualization design, merely a quick mockup for code testing purposes. In the figure, simultaneous plotting of 1000 individual data series can be seen, and the temperature stress during each stress summary interval forms the red backdrop, contextualizing the data series. The figure demonstrates that Vega-Altair features will at least be mostly sufficient for implementing the designed visualization, even if interactivity features are not yet tested. Additionally, the performance in constructing this visualization is remarkably fast despite the large number of data series, and so it is not expected that performance will become a limiting issue for the initial version of GraceFall.

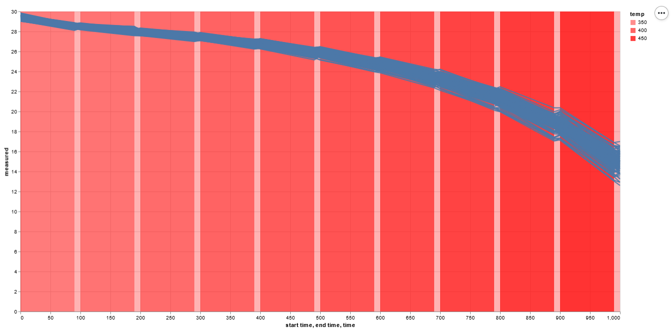

Fig. 5. Time-series visualization of wear-out data overlaid on stress interval data using GraceFall boilerplate code.

5 Milestones

Our visualization can conveniently be divided into a few conceptual components that are reasonably separate in terms of design, easing the process of breaking down the project into tasks. In terms of milestones, we define three internally: **1)** Literature Review Completed, **2)** Design Plan Work Completed, and **3)** Implementation and Testing Completed. Reports are not included in these internal milestone descriptions. A summary of the project execution plan, vertically divided by milestone and course deliverable tasks (tasks defined by feature requirements), is provided as a Gantt chart in Figure 6.

Some key elements within the provided Gantt chart are the milestone and deliverable deadlines and dependency ordering of literature review to design plan down to implementation for the different design components. The project update deadline is on November 15$^{\text{th}}$, or just into week 3 of November, the project presentation is December 14$^{\text{th}}$, or middle of week 2, and the final report is due on December 16$^{\text{th}}$, the end of that same week. Based on the work breakdown within the chart, each filled cell can be mapped to approximately 4 hours of expected work to obtain an estimate of project effort that slightly overshoots the course recommendation for total project work hours. The project group has no issue with this light increase in workload. The Milestone 1 deadline is set for middle of week 2 in November (32 hours of expected work), Milestone 2 by end of November week 4 (56 hours), and 3 by end of week 1 December (80 hours).

To date, progress has remained reasonably on schedule compared to the chart provided at the proposal phase, however a few items were delayed due to issues of group member sickness in W1 November. At this point there is little anticipated risk to overall project execution; development should be completed in good time.

Due to the large planned scope of the project, there is a significant risk that the available time will be insufficient to design and implement all desired features. To mitigate this risk the model fitting design component has been designated as lower importance; if necessary this aspect of the project can be cut. Doing so would save 40 hours of allocated time, based on the chart timeline breakdown, that can then be reallocated to the test data integration plots and test plot navigation portions of the project as needed.

6 Conclusions

The GraceFall tool will tackle a challenging visualization problem, cleanly displaying highly varied scalar data series to avoid the large potential for visual clutter and obscured properties of interest. If successful, the tool will greatly aid engineers in developing accelerated reliability stress test procedures and tuning models for explaining physical degradation phenomenon. The large scope of this project necessitates a clear execution plan which has been developed to maximize the potential of constructing an effective solution.

References

[1] M. Ali, A. Alqahtani, M. W. Jones, and X. Xie. Clustering and classification for time series data in visual analytics: A survey. *IEEE Access*, 7:181314–181338, 2019.

[2] J. Bernard, N. Wilhelm, M. Scherer, T. May, and T. Schreck. Timeseriespaths: Projection-based explorative analysis of multivariate time series data. 2012.

[3] M. Booshehrian, T. Möller, R. M. Peterman, and T. Munzner. Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making. *Computer Graphics Forum*, 31(3pt3):1235–1244, 2012. doi: 10.1111/j.1467-8659.2012.03116.x

[4] T. Fujiwara, J. K. Li, M. Mubarak, C. Ross, C. D. Carothers, R. B. Ross, and K.-L. Ma. A visual analytics system for optimizing the performance of large-scale networks in supercomputing systems. *Visual Informatics*, 2(1):98–110, 2018.

[5] I. Hill, P. Chanawala, R. Singh, S. A. Sheikholeslam, and A. Ivanov. Cmos reliability from past to future: A survey of requirements, trends, and prediction methods. *IEEE Transactions on Device and Materials Reliability*, 22(1):1–18, 2022. doi: 10.1109/TDMR.2021.3131345

[6] I. Hill and M. Tang. Gracefall. GitHub, 2022. Available: https://github.com/ianrmhill/gracefall.

[7] S. P. Kesavan, T. Fujiwara, J. K. Li, C. Ross, M. Mubarak, C. D. Carothers, R. B. Ross, and K.-L. Ma. A visual analytics framework for reviewing streaming performance data. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 206–215. IEEE, 2020.

[8] M. Krstajic, E. Bertini, and D. Keim. Cloudlines: Compact display of event episodes in multiple time-series. *IEEE transactions on visualization and computer graphics*, 17(12):2432–2439, 2011.

[9] J. Li, K. Zhang, and Z.-P. Meng. Vismate: Interactive visual analysis of station-based observation data on climate changes. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 133–142. IEEE, 2014.

[10] L. Liu and R. Vuillemot. Groupset: A set-based technique to explore time-varying data. In *Computer graphics Forum (Proc. Eurovis), 2022*, 2022.

[11] K. Matkovic, D. Gracanin, M. Jelovic, and H. Hauser. Interactive visual steering - rapid visual prototyping of a common rail injection system. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1699–1706, 2008. doi: 10.1109/TVCG.2008.145

[12] D. Moritz and D. Fisher. Visualizing a million time series with the density line chart. *arXiv preprint arXiv:1808.06019*, 2018.

[13] S. W. Pae, H. C. Sagong, C. Liu, M. J. Jin, Y. H. Kim, S. J. Choo, J. J. Kim, H. J. Kim, S. Y. Yoon, H. W. Nam, H. W. Shim, S. M. Park, J. K. Park, S. C. Shin, and J. W. Park. Considering physical mechanisms and geometry dependencies in 14nm finfet circuit aging and product validations. In *2015 IEEE International Electron Devices Meeting (IEDM)*, pp. 20.6.1–20.6.4, 2015. doi: 10.1109/IEDM.2015.7409744

[14] G. Park, H. Yu, M. Kim, and C. H. Kim. An All BTI (N-PBTI, N-NBTI, P-PBTI, P-NBTI) Odometer based on a Dual Power Rail Ring Oscillator Array. In *2021 IEEE International Reliability Physics Symposium (IRPS)*, pp. 1–5. IEEE, Monterey, CA, USA, Mar. 2021. doi: 10.1109/IRPS46558.2021.9405181

[15] M. Steiger, J. Bernard, S. Mittelstädt, H. Lücke-Tieke, D. Keim, T. May, and J. Kohlhammer. Visual analysis of time-series similarities for anomaly detection in sensor networks. In *Computer graphics forum*, vol. 33, pp. 401–410. Wiley Online Library, 2014.

[16] T. Trautner and S. Bruckner. Line weaver: Importance-driven order enhanced rendering of dense line charts. In *Computer Graphics Forum*, vol. 40-3, pp. 399–410. Wiley Online Library, 2021.

[17] J. J. Van Wijk and E. R. Van Selow. Cluster and calendar based visualization of time series data. In *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis' 99)*, pp. 4–9. IEEE, 1999.

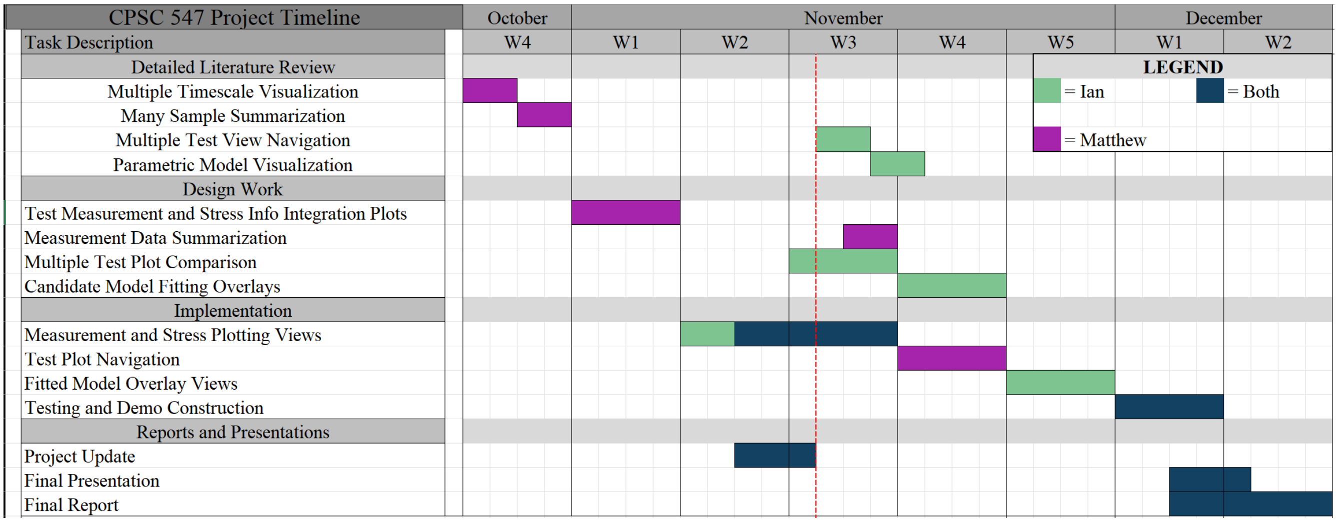

Fig. 6. Project work breakdown by task type and design component. Each filled cell represents approximately 4 hours of expected work.

[18] C. Yang and F. Feng. Multi-Step-Ahead Prediction for a CMOS Low Noise Amplifier Aging Due to NBTI and HCI Using Neural Networks. *J Electron Test*, 35(6):797–808, Dec. 2019. doi: 10.1007/s10836-019-05843-7

[19] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan. Exploratory analysis of time-series with chronolenses. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2422–2431, 2011.

[20] Y. Zhao, Y. Wang, J. Zhang, C.-W. Fu, M. Xu, and D. Moritz. Kd-box: Line-segment-based kd-tree for interactive exploration of large-scale time-series data. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):890–900, 2022. doi: 10.1109/TVCG.2021.3114865