A Digital Twin Framework for Real-Time Damage Localization and Stress Field Prediction in Plate Structures

Colin Driggs*Aiden Holley†, Dylan Monge‡, Ricky Montano§, Zahra Sotoudeh ¶
California State Polytechnic Institute, Pomona, CA 91768

This paper reports on the progress of the ongoing Cal Poly Pomona Digital Twin Technology for Aerospace Engineering (CPP-DiTTA) project. In its present form, CPP-DiTTA employs a set of trained neural networks to predict the location of damage in a plate structure from a limited number of strain-gauge measurements, estimate the corresponding stress field under the applied load, and forecast the stress response under alternative loading conditions. These capabilities allow users not only to identify damage relative to the pristine configuration but also to evaluate how the structure behaves under different loads and to make more informed decisions about structural performance. CPP-DiTTA includes a visualization module that renders the predicted stress fields and is deployed on a publicly accessible website to support educational use and demonstration.

I. Introduction

In the literature, the term *digital twin* is used to describe simulations with varying levels of data integration and connectivity to a physical asset: detailed simulations without data exchange (*digital models*), simulations with one-way data flow from the physical asset to the digital replica (*digital shadows*), and fully integrated systems with bidirectional data exchange (*true digital twins*).[1] Although many articles use the term *digital twin,* most reported implementations fall into the first two categories. Kritzinger et al. show that only 18% of surveyed smart-manufacturing papers involve any form of two-way data exchange, and nearly all of those are conceptual rather than operational. While no equivalent quantitative review exists for aerospace engineering, the same trend holds. Hu et al. note that many aerospace-oriented digital twin models remain limited to one-way data flow and insufficient bidirectional integration, which continue to be central challenges for developing high-fidelity digital twins.[2]

Against this backdrop, Ferrari and Willcox argue that digital twins in aerospace deliver value not by recreating systems in exhaustive detail but by serving as fit-for-purpose tools that integrate physics-based models, data-driven methods, and continual bidirectional updating throughout a system's life cycle.[3] They further emphasize that a digital twin must be treated as an engineered asset in its own right, with a structured life cycle spanning conception, development, validation, operation, updating, and eventual disposal. Recent reviews reinforce this perspective. Xiong and Wang trace the evolution of aviation digital twins from early NASA and AFRL concepts to modern, multidimensional models that rely heavily on data fusion, advanced simulation, and emerging IT systems, but they also highlight that substantial engineering challenges remain.[4] Li et al. clarify common misconceptions and outline the essential components of aerospace digital twins, physical sensing, virtual analytical models, and data connectivity, while comparing physics-based, data-driven, and hybrid approaches.[5] Additional surveys extend these ideas to the space domain, from satellite systems and VR-based simulators to lunar mission planning,[6] and even to digital twins of the space environment itself, noting the unique opportunities and constraints such systems entail.[7]

*Undergraduate Students, Aerospace Engineering Department
†Undergraduate Students, Aerospace Engineering Department
‡Undergraduate Students, Computer Science Department
§Undergraduate Students, Aerospace Engineering Department; student authors contributed equally; their names appear in alphabetical order.
¶Professor, Department of Aerospace Engineering, AIAA Associate Fellow.

In this work, we adopt the AIAA definition of a digital twin: "A digital twin is a set of virtual information constructs that mimics the structure, context, and behavior of an individual/unique physical asset, is dynamically updated with data from its physical twin throughout its life cycle, and informs decisions that realize value".[8] This definition distinguishes digital twins from traditional simulations, digital models, and shadows by requiring real-time data exchange, diagnostic capability, and a feedback mechanism that influences the physical system. Only a limited number of studies demonstrate such fully realized digital twins in aerospace engineering.[3,9] The Cal Poly Pomona Digital Twin Technology for Aerospace Engineering (CPP-DiTTA) project aims to help close this gap by providing an accessible, web-based demonstrator that introduces students and researchers to the essential components of a true digital twin, albeit for a simple structural system. This paper reports on the ongoing development of the CPP-DiTTA initiative.

II. CPP-DiTTA Framework

The CPP-DiTTA aims to establish an educational and functional digital twin framework for structural health monitoring. Using only a limited set of strain-gauge measurements, the current demonstrator predicts the damage location and reconstructs the corresponding stress field in real time. It then estimates the stress distribution of the damaged structure under alternative loading conditions. The entire system is implemented as a publicly accessible website `https://digitaltwintech.studio/`.

The current CPP-DiTTA version moves toward a true digital twin by explicitly incorporating a human-in-the-loop component. The digital twin receives data from the physical asset via strain-gauge sensors, enables users to evaluate stress fields under different load scenarios, and supports informed decisions about appropriate loading. The operator then applies the selected load to the physical structure, completing a human-in-the-loop bidirectional exchange between the physical asset and its digital twin. Fig. 1 shows a schematic of CPP-DiTTA.

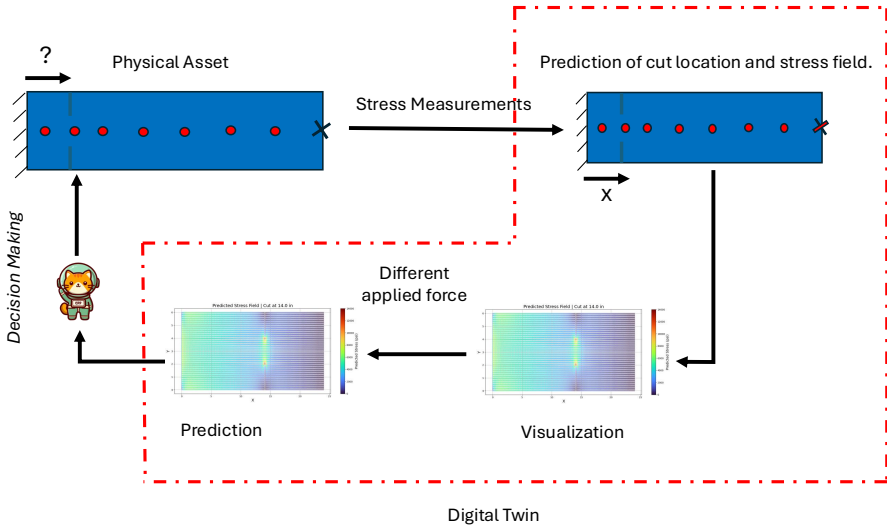


Figure 1: *CPP-DiTTA framework*

III. The Physical Asset

The physical asset is a rectangular aluminum plate (Al 6061–T6) with dimensions $24 \times 6 \times 1$ in. We begin with known structural and geometric properties corresponding to a pristine (undamaged) plate. The plate is clamped at one end and subjected to a point force at the free end. Damage is introduced by machining two symmetric cuts (0.1667×2 in) at a specified location measured from the clamped end.

Strain gauges are mounted along the length of the plate at the mid-chord line. The initial configuration used ten equally spaced sensors positioned from 4 in to 22 in measured from the clamped edge. A random

forest algorithm was applied to rank sensor importance for predicting the damage location. Based on this analysis, the sensors at 8, 12, and 14 in were removed from the final configuration. The resulting instrumented structure, therefore, uses seven strain gauges. Figure 2 shows one such plate.

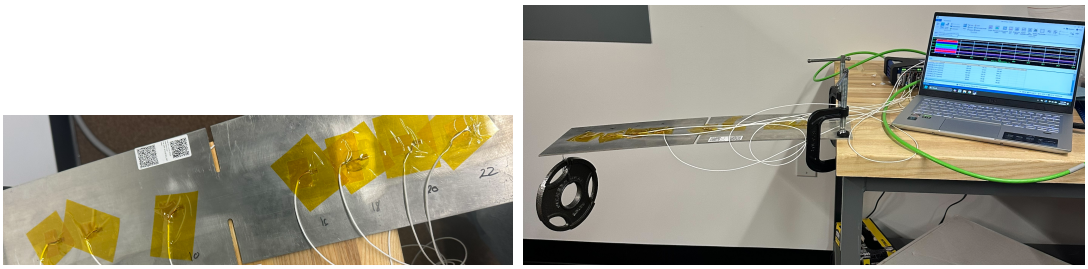


Figure 2: *Experimental setup*

A. Database and Finite Element Modeling

The physical system was modeled in ANSYS[10] to generate the training data required for our predictive framework. The pristine and damaged plate configurations were used to validate the ANSYS simulations and verify the full digital twin pipeline. We created a light dataset consisting of 80 test cases at different damage locations and applied forces of 2.5, 5, 7.5, 10, 12, and 15 lb, resulting in a total of 480 ANSYS runs. The ANSYS models also include gravitational loading.

The finite element model is built in ANSYS using the Solid186 element, with automatic mesh generation and the Sparse Matrix Direct Solver. Solid186 provides quadratic displacement behavior and is well-suited to capturing bending-dominated responses in slender structures, making it appropriate for the plate geometry considered here. Figure 3 displays the mesh convergence study. We used more than 50,000 nodes per model to ensure convergence and to resolve stress gradients near the damage. The Sparse Matrix Direct Solver was selected for its robustness and consistency across a wide range of stiffness conditions, which is especially important when evaluating multiple damage locations and load levels in batch simulations. We verified our ANSYS calculations against experimental measurements for several damage locations and

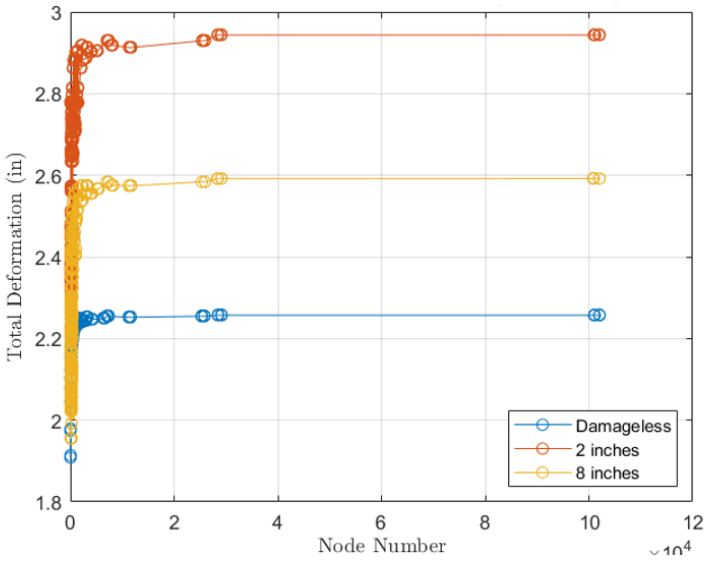
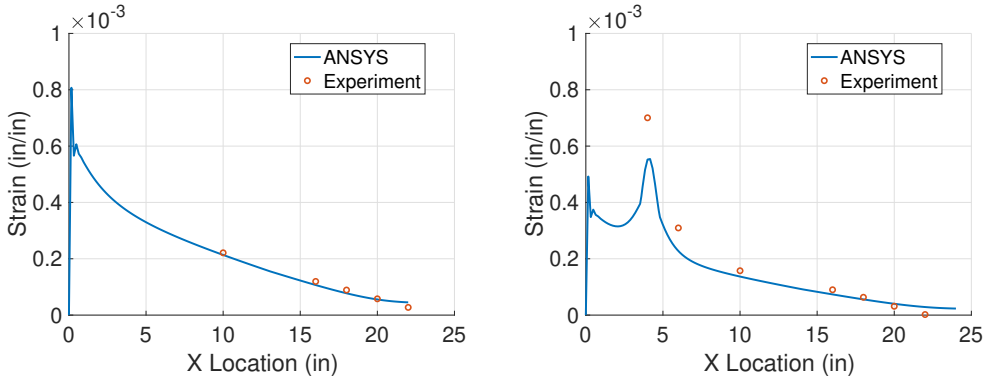


Figure 3: *Mesh convergence study*

applied loads. Figure 4 shows the ANSYS and experimental results for a pristine plate and for a plate with damage at 4 in from the clamped end under a 2.5 lb force. The discrepancy in the results may be due to

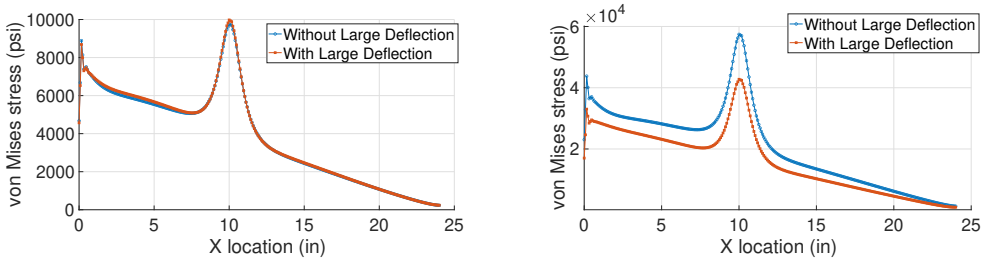
the experimental setup, including the clamping conditions. Further investigation is underway. For larger



(a) Pristine plate (b) Damaged plate

Figure 4: *Experimental verification of ANSYS modeling for the Pristine plate and the damaged plate with cuts at 4 in from the clamped end.*

forces, we analyzed the structure using ANSYS *Large Deflection Analysis*. In ANSYS, Large Deflection is a nonlinear setting that accounts for significant geometric changes (large rotations and strains) and stiffness variations during deformation. The results with and without the large deflection option are nearly identical for small applied loads; however, the impact of large deformation becomes significant as the applied force increases, as expected. Figure 5 (a) shows the von Mises stress along the span of the plate for a cut at 10 in under 2.5 lb and 15 lb forces. The solver without large deflection predicts higher stresses, and a similar trend is observed in the deformation field. Figure 5 (b) also shows the out-of-plane deformation (perpendicular to the plate). Using the simulation without large deflection produces a more conservative digital twin while reducing computational cost. Therefore, we did not use the large deformation solver to generate our database.

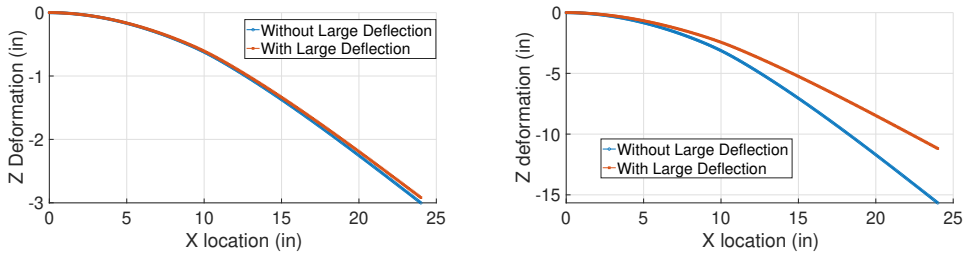


(a) 2.5 lb force (b) 15 lb force

Figure 5: *von Mises stress (psi) with and without large deflection analysis in ANSYS for the damaged plate with a cut at 10 in from the clamped end.*

IV. Digital Twin and Predictive Models

The CPP-DiTTA framework includes two embedded predictive models, both implemented as neural networks (NNs) with different architectures. The first model, which uses the seven strain-gauge measurements as input, predicts the damage location. The second NN predicts the von Mises stress at a given point in the plate under an applied force. This model generates the stress field for the measured load, and the user can subsequently use it to estimate the stress field under alternative loading conditions. Both NNs follow a fully connected residual architecture implemented in PyTorch,[11] designed to promote stable training, maintain effective gradient flow, and provide sufficient representational capacity to capture the nonlinear physical



(a) 2.5 lb force (b) 15 lb force

Figure 6: *Deformation with and without large deflection analysis in ANSYS for the damaged plate with a cut at 10 in from the clamped end.*

relationships in the plate.

A. Prediction of the Damage Location

We used seven sensor inputs to train a neural network (NN) for predicting the damage location. The NN architecture consists of three hidden layers with 128, 64, and 32 nodes, respectively. Dropout of 5% in the first hidden layer and 1% in the second is applied to mitigate overfitting. The Rectified Linear Unit (ReLU) is used as the activation function across all layers. We allocated 80% of the data for training and reserved the remaining 20% as an unseen test set for verification. The model was trained using the Adam optimizer with a learning rate of 0.005. A learning-rate scheduler automatically reduced the learning rate by 50% when the validation loss plateaued for five epochs, and early stopping was triggered if the validation loss did not improve for 10 epochs. Although the model was trained for up to 200 epochs, early stopping halted training at epoch 90. Figure 7 shows the loss function values for both the training and validation sets. Training and validation losses track closely, indicating minimal overfitting.

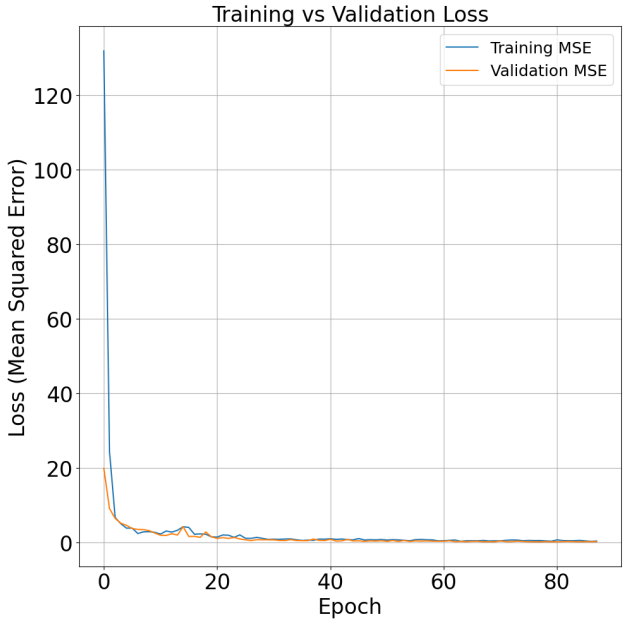


Figure 7: *Convergence of NN training for predicting damage location*

A baseline linear regression model was trained on the same seven sensor inputs for comparison. Linear regression assumes a linear relationship between sensor readings and damage location. Figure 8 compares the predicted damage locations from the NN and the linear model against the actual data. Table 1 reports

the R^2 score and mean absolute error (MAE) for both models, demonstrating that the NN substantially outperforms the linear regression baseline.

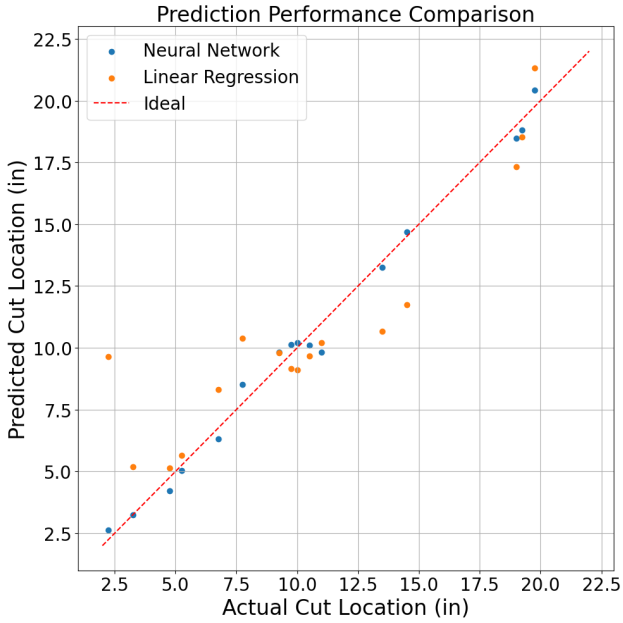


Figure 8: *Prediction of cut location, NN and linear model*

error	Linear model	NN
R^2	0.80	0.99
MAE (in)	1.71	0.41

Table 1: *R^2 score and mean absolute error (MAE) for the NN and the linear model in predicting the damage location*

B. Choosing the Training Dataset

Traditionally, datasets are divided 80/20 into training and validation subsets. Before creating the NN model for predicting the stress field, we examined two strategies for splitting the dataset. To isolate the effect of the split itself, we trained NNs to predict the stress field at a fixed applied force (2.5 lb). Each data point corresponds to a single mesh node and includes the 3D coordinates (x, y, z) (in inches), the damage location, and the von Mises stress (in psi). These NNs therefore predict the stress given the spatial coordinates and location of the damage. We tested two different approaches for constructing the training set:

1. **Model-based split:** Use 80% of the ANSYS simulation runs (damage-location cases) and include *all* mesh nodes from those runs in the training set.

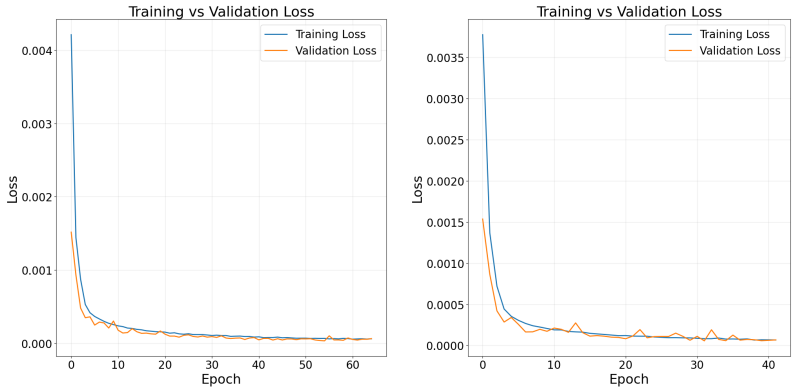
2. **Global random split:** Combine all data points across all simulations and randomly assign 80% of the points to the training set, without regard to model grouping.

In each case, the remaining 20% of the data was used for validation.

Both experiments use the same neural network architecture, a moderately deep feed-forward model designed to learn the nonlinear mapping from geometric features to von Mises stress. The input layer feeds into a 128-unit dense projection layer with ReLU activation. This is followed by three sequential Residual Blocks, each containing two fully connected 128-unit layers with ReLU activations and 10% dropout. These blocks increase the model's depth while preserving effective gradient flow. In total, the network consists of one input layer, six hidden layers within the residual blocks, and a two-stage output head, yielding nine trainable layers (excluding normalization and activation operations). After the residual stack, the representation is

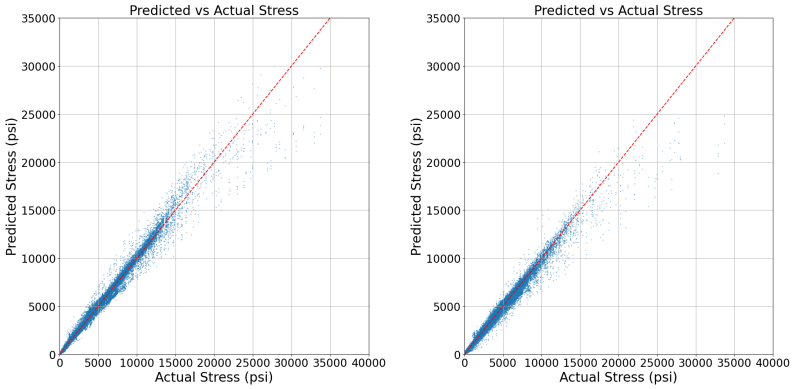
compressed through a 128–64–1 output pathway, with a final Softplus activation to enforce nonnegative stress predictions. Training uses a weighted Huber loss that assigns greater weight to high-stress samples, improving accuracy in physically critical regions. This fixed architecture is used consistently across the two data-handling strategies described above.

Both models perform well. Figure 9 shows the convergence of both models. Figure 10 shows their predictive performance, where each point represents the NN-predicted stress at a mesh node across different damage locations. The strong alignment of points along the line ($y = x$) indicates excellent predictive accuracy for both approaches. Table 2 reports the R^2 score and mean absolute error (MAE), demonstrating that both strategies for splitting the data yield valid and reliable models.



(a) Model-based split (b) Global random split

Figure 9: *The loss function for the NN models trained using two different dataset-splitting strategies*



(a) Model-based split (b) Global random split

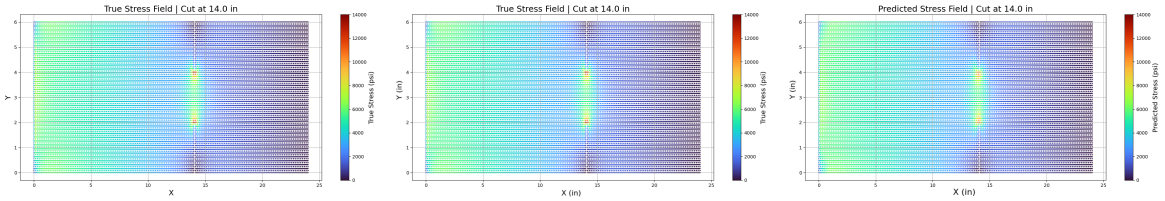
Figure 10: *Performance of NN models trained using two different dataset-splitting strategies*

error	Model-based split	Global random split
R^2	0.9942	0.9930
MAE (psi)	46.39 psi	50.10

Table 2: *R^2 score and mean absolute error (MAE) for two NN models using different training datasets*

Figure 11 shows the stress field for a plate with damage at 14 in from the clamped end. Figure 11(a) presents the ANSYS simulation results, and Figures 11(b) and (c) show the corresponding NN predictions obtained using the model-based split and global random split, respectively. Both models closely reproduce the

structural stress field, indicating that the choice of data-splitting strategy has minimal impact on predictive performance for this configuration.



(a) ANSYS results (b) Model-based split (c) Global random split

Figure 11: *Prediction of stress field for a plate with damage at 14 in under 2.5 lb load using two NN models trained with different dataset-splitting strategies*

C. Prediction of the Stress Field

Once the cut location is predicted, we use a more advanced neural network (NN) to estimate the stress field in the damaged structure. This model predicts the von Mises stress at any point in the plate, given its spatial coordinates, the damage location, and the applied load. The training data for this model come from a comprehensive set of ANSYS finite element simulations. Each simulation corresponds to a different cut location and applied load case, capturing the resulting stress distribution across the plate. The complete dataset contains 480 simulation files and more than 19 million data points. Each data point corresponds to a single mesh node and includes the 3D coordinates ((x, y, z) in inches), the damage location (in inches), the applied load (in pounds), and the von Mises stress (in psi). The input layer, therefore, consists of five neurons representing these quantities.

The network architecture includes two hidden layers, each with 512 neurons. A LayerNorm operation is applied at each layer to stabilize training and normalize activations. The activation function is the Gaussian Error Linear Unit (GELU), which provides smooth nonlinear behavior and improved robustness over ReLU for continuous physical data. We apply a 1% dropout rate for mild regularization and include a residual skip connection ($x + f(x)$) to support stable gradient propagation. This residual pathway helps the network learn both low- and high-order stress behaviors, reflecting the layered nature of physical interactions. A final compression stage reduces the dimensionality from 512 to 128 and then to a single output neuron representing the predicted von Mises stress.

A weighted Huber loss function is used to balance smooth convergence while placing additional emphasis on high-stress regions (the upper 5% of stress values receive higher loss weights). This encourages the model to focus on physically critical stress concentrations, such as those near the damage and under higher applied loads. The model is trained with the AdamW optimizer, using a learning rate of 3×10^{-4} and a weight decay of 10^{-5}. A CosineAnnealingLR scheduler with $T_{\max} = 8$ gradually reduces the learning rate to refine convergence. The batch size is 2048, selected to balance GPU efficiency and stable gradient estimates.

To accommodate the large dataset, all 19 million data points are stored as memory-mapped arrays (.mmap), allowing training without exceeding system RAM. The dataset was split 80/20 into training and validation subsets, using the global random split, ensuring uniform coverage across damage locations and load levels. Because the dataset was split 80/20 across all damage locations and load levels, the training and validation subsets share the same underlying physical regimes. As a result, the surrogate model is primarily evaluated on its ability to interpolate within the domain represented by the ANSYS simulations rather than extrapolate beyond it. This approach is appropriate for the current demonstrator, whose goal is to reproduce the structural response within the operating envelope of the physical system rather than to predict behavior outside that range. Figure 12 shows the loss function values for both the training and validation sets. Figure 13 shows stress field predictions using three different models for comparison. In Fig. 13(a), each point represents the stress predicted by the NN at a mesh node for different damage locations and applied force levels. The strong alignment of points along the best-fit line ($y = x$) indicates excellent predictive accuracy. The fitted regression line closely matches the ideal relationship, with a slope of 0.999, confirming that the network accurately reproduces the stress distribution from the ANSYS simulation data. For comparison, we also evaluate a linear regression model (Fig. 13(b)) and a polynomial regression model (Fig. 13(c)).

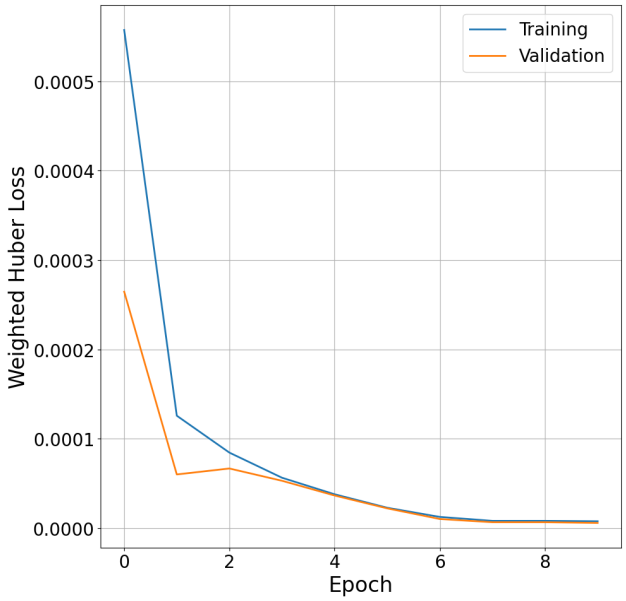
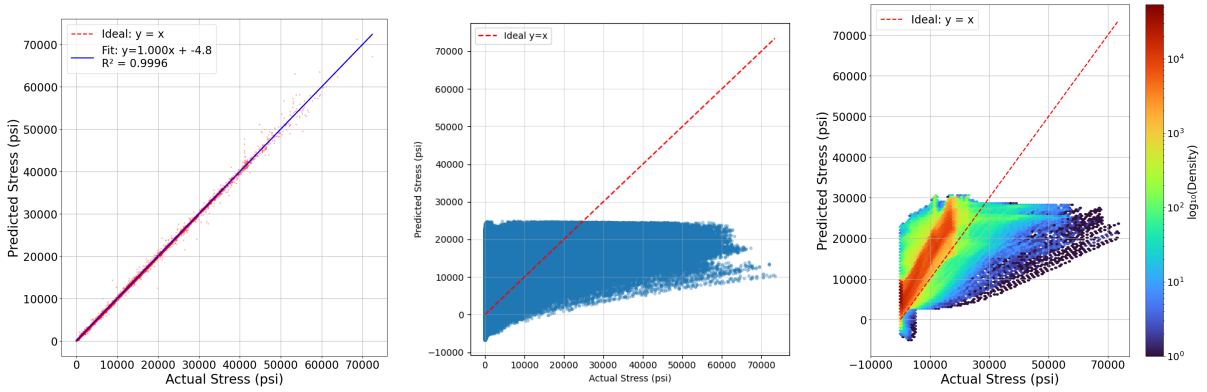


Figure 12: *Convergence of NN training for stress field*

The linear regression model was trained using the same dataset and preprocessing pipeline as the NN. It uses the same five input features ((x, y, z), damage location, and applied load) and predicts the corresponding von Mises stress. As expected, this linear baseline cannot capture the nonlinear interactions or cross-feature dependencies present in physical stress fields, such as bending behavior or stress concentration near the damage. In Fig. 13(b), the dashed line ($y = x$) represents the ideal prediction. The linear model systematically underestimates high stress values and exhibits an apparent saturation at approximately 25,000 psi, illustrating its inability to represent nonlinear stress behavior.

To provide a stronger traditional benchmark, a polynomial regression model was developed using the same normalized dataset. This model extends the linear approach by introducing nonlinear terms and interaction effects. For this study, a third-order polynomial was selected to capture more complex curvature and cross-dependencies between geometry and load. It includes all third-order polynomial combinations of the five core inputs ((x, y, z), damage location, and applied load). In Fig. 13(c), the dashed line ($y = x$) again represents the ideal response. Although the polynomial model incorporates nonlinear terms, its predictions still fail to reproduce the true stress distribution, with most values clustering between 0 and 25,000 psi.



(a) NN model (b) Linear model (c) Polynomial model

Figure 13: *Prediction of stress field using a NN, a linear regression model, and a polynomial regression model*

Table 3 reports the R^2 score and mean absolute error (MAE) for the NN, the linear regression model, and the polynomial regression model. MAE values appear large because stress magnitudes span a wide range, especially near damage-induced stress concentrations.

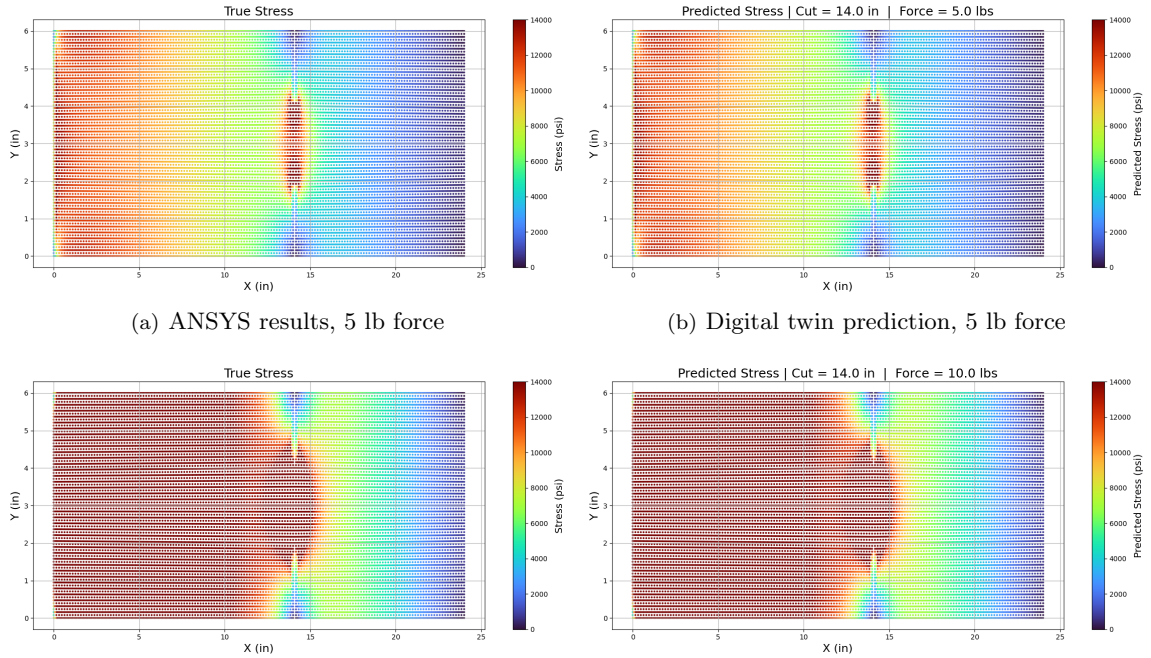
error	Linear model	polynomial model	NN
R^2	0.53	-0.0476	0.99
MAE (psi)	4,079.57	5790.75	0.6731

Table 3: *R^2 score and mean absolute error (MAE) for the NN, the linear, and polynomial models in predicting stress field*

Figure 14 compares the stress field computed using ANSYS (the true stress field) with the corresponding field predicted by the digital twin for a plate with damage at 14 in from the clamped end. Figures 14(a) and 14(b) show the results for a 5 lb applied force; in this case, stress values at some mesh nodes (though not all) were included in the training dataset. Figures 14(c) and 14(d) present the results for a 9 lb applied force, a load level not included in the training data. Thus, the digital twin predicts the entire stress field for a previously unseen applied load.

Figure 15 shows the results for a plate with damage at 11.4 in from the clamped end under a 9 lb applied force. Neither the 11.4 in damage location nor the 9 lb force level appears in the training dataset. Even in this entirely unseen case, the predicted stress field remains highly accurate.

In all scenarios, the predicted fields show strong agreement with the ANSYS results.



(a) ANSYS results, 5 lb force

(b) Digital twin prediction, 5 lb force

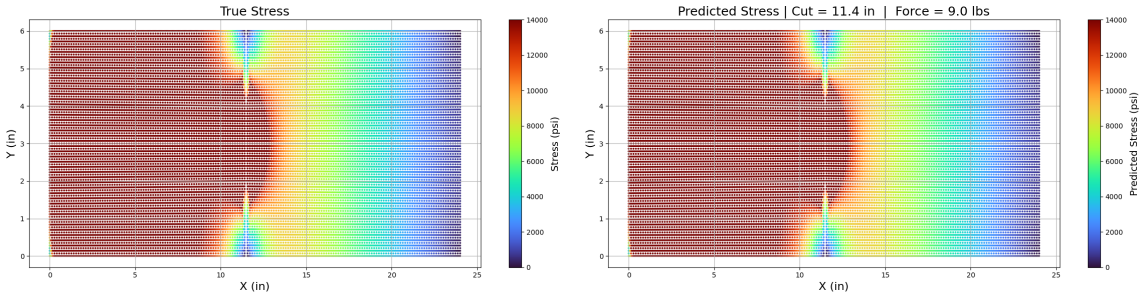
(c) ANSYS results, 9 lb force

(d) Digital Twin prediction, 9 lb force

Figure 14: *Stress field for a plate with the damage at 14 in from the clamped end, under 5 lb and 9 lb force (unseen load case)*

V. Website Implementation

We deployed CPP-DiTTA on a publicly accessible website (`https://digitaltwintech.studio/`) for ease of access and demonstration. After training, all predictive models are exported to the Open Neural Network Exchange (ONNX) format, which provides a standardized, framework-agnostic representation of the models. This enables deployment through ONNX Runtime Web (ORT), allowing the models to run



(a) ANSYS results (b) Digital twin prediction

Figure 15: *Stress field for a plate with the damage at 11.4 in from the clamped end, under 9 lb force (unseen damage-location and load case)*

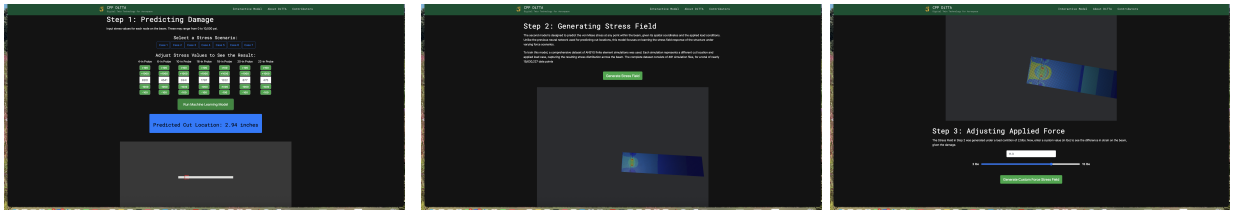
entirely in the browser using JavaScript APIs and libraries. All inference is performed on the client side.

Using ORT, the full inference pipeline can be executed on a fully static website. The site is hosted on GitHub Pages, which offers free hosting for public repositories and supports static site delivery. In general, web computation can be handled either statically or dynamically. A static site can only deliver files to the client and cannot execute server-side computations or generate new data on demand. Dynamic sites, by contrast, run computations on a server in response to client requests, reducing the computational burden on the client but increasing deployment complexity and long-term cost. Running inference locally on a static site more closely resembles a realistic digital twin deployment, where a local machine, not a remote server, executes the model.

The visualization module is implemented using Bevy,[12] a data-driven game engine written in Rust with WebAssembly (WASM) support. WASM provides a compact, efficient binary format designed for fast execution in the browser. ORT also uses WASM under the hood for performance, while exposing a JavaScript interface for model interaction.

The interaction between the website and the predictive models proceeds as follows. The user inputs strain or stress data through the web interface, which are then passed to the appropriate ONNX model for inference. Once the model completes its computation and returns the predicted fields, those results are forwarded to the visualization module for rendering.

Figure 16 shows three frames corresponding to the three modules of the CPP-DiTTA. In frame one (Fig. 16(a)), the user inputs the sensor data into the system, and the digital twin predicts the damage location. In frame two (Fig. 16(b)), the digital twin predicts the stress field for the current loading condition. Finally, the user may choose a different applied force, and the digital twin computes the corresponding stress field, as shown in frame three (Fig. 16(c)). The website also includes several test cases for demonstration.



(a) Frame one (b) Frame two (c) Frame three

Figure 16: *Three frames of CPP-DiTTA website demonstrator*

VI. conclusions

This work presented the current stage of the CPP-DiTTA framework, a digital twin demonstrator designed to perform real-time damage localization and stress-field prediction for a simple plate structure. Using a small number of strain-gauge measurements and two neural-network surrogate models, the system accurately identifies the damage location, and it reconstructs the corresponding stress distribution under both measured and user-selected loading conditions. The close agreement between neural-network predictions and ANSYS finite-element simulations demonstrates that lightweight, data-driven models can capture the essential structural response required for rapid decision-making.

With a human-in-the-loop, CPP-DiTTA moves toward the true definition of a digital twin: a construct that enables bidirectional data exchange between the physical asset and its virtual counterpart. The digital twin ingests sensor data from the plate, produces diagnostic and predictive outputs, and enables the operator to select new loading conditions, which are then applied to the physical asset. Although the structural system considered here is intentionally simple, CPP-DiTTA includes all core components of a digital twin: sensing, data ingestion, predictive modeling, visualization, and (human-in-the-loop) feedback to the physical system. Therefore, CPP-DiTTA serves as a practical, accessible technological demonstrator that illustrates how digital twins can be implemented and deployed in aerospace applications.

Future development of CPP-DiTTA will focus on expanding both the experimental and modeling capabilities of the framework. Experimentally, we plan to extend the sensing architecture to include fiber-optic strain sensors, which offer higher spatial resolution and greater robustness than traditional strain gauges. Additionally, we will explore a broader range of damage geometries and loading conditions in future development. These efforts will advance CPP-DiTTA toward a more comprehensive and fully realized digital twin for structural health monitoring.

VII. Acknowledgments

This material is based on research sponsored by AFRL under agreement number FA8650-24-2-2403. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes, notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors. They should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL or the U.S. Government.

References

[1]Kritzinger, W., Karner, M., Traar, G., Henjes, J., and Sihn, W., "Digital Twin in manufacturing: A categorical literature review and classification," *Ifac-PapersOnline*, Vol. 51, No. 11, 2018, pp. 1016–1022.

[2]Hu, W., Zhang, T., Deng, X., Liu, Z., and Tan, J., "Digital twin: A state-of-the-art review of its enabling technologies, applications and challenges," *Journal of Intelligent Manufacturing and Special Equipment*, Vol. 2, No. 1, 2021, pp. 1–34.

[3]Ferrari, A. and Willcox, K., "Digital twins in mechanical and aerospace engineering," *Nature Computational Science*, Vol. 4, No. 3, 2024, pp. 178–183.

[4]Xiong, M. and Wang, H., "Digital twin applications in aviation industry: A review," *The International Journal of Advanced Manufacturing Technology*, Vol. 121, No. 9, 2022, pp. 5677–5692.

[5]Li, L., Aslam, S., Wileman, A., and Perinpanayagam, S., "Digital twin in aerospace industry: A gentle introduction," *IEEE Access*, Vol. 10, 2021, pp. 9543–9562.

[6]Kim, M. U., "A survey on digital twin in aerospace in the new space era," *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, IEEE, 2022, pp. 1735–1737.

[7]Liu, W., Wu, M., Wan, G., and Xu, M., "Digital twin of space environment: Development, challenges, applications, and future outlook," *Remote Sensing*, Vol. 16, No. 16, 2024, pp. 3023.

[8]AIAA Digital Engineering Integration Committee and others, "Digital Twin: Definition & Value—An AIAA and AIA Position Paper," *AIAA: Reston, VA, USA*, 2020.

[9]Kapteyn, M. G., Knezevic, D. J., and Willcox, K., "Toward predictive digital twins via component-based reduced-order models and interpretable machine learning," *AIAA Scitech 2020 Forum*, 2020, p. 0418.

[10]"ANSYS Mechanical," ANSYS, Inc.

[11]Steiner, B., DeVito, Z., Chintala, S., Gross, S., Paske, A., Massa, F., Lerer, A., Chanan, G., Lin, Z., Yang, E., et al., "Pytorch: An imperative style, high-performance deep learning library," 2019.

[12]Bevy Contributors, "Bevy Engine: A Data-Driven Game Engine Built in Rust," `https://github.com/bevyengine/bevy`.