Proceedings of the IASS 2024 Symposium
Redefining the Art of Structural Design
August 26-30, 2024, Zurich Switzerland
Philippe Block, Giulia Boller, Catherine DeWolf,
Jacqueline Pauli, Walter Kaufmann (eds.)

Data evaluation of a Form-Force-Pattern mechanism; A Machine learning approach on stripe segmentation of minimal shell structures

Pablo BAQUERO*, Effimia GIANNOPOULOU[a], Kalliopi VALSAMIDOU[b]

*UNIR
Av. de la Paz, 137, 26006 Logroño, La Rioja
pablonicolas.baquero@unir.net

[a] iBAG, Faberarium
[b] Aristotle University of Thessaloniki

Abstract

Previous research has examined the intersection between mesh relaxation processes and weighted meshGraph representations, as design tools, for the fabrication and construction of self-sustained minimal shell skins. As an outcome, stripe patterns of flexible but rigid material, have been used as a construction logic and analyzed in relation to the stress lines and structural performance. However, a discussion was raised about the appropriate tools for the generation of the stripe morphology controlled by stress directions. Here, the performance reference criteria of structural stripe design for minimal shell structures are introduced, focusing on the design strategies which have impact on the whole stripe pattern. A database is generated with the meshGraph weights unequally distributed in favor of stress lines and boundary naked edges as input directions. A Recurrent Neural Network is tested to predict an optimal pattern organization balancing structural and fabrication criteria. Pearson correlations coefficients were investigated between sets of data. To evaluate the results, the mean squared error is used to check between the predicted combinations of directional/stress lines from the recurrent model in comparison to the ones generated with traditional means using kruskal-graph weighted algorithms following the structural performance.

Keywords: Segmentation algorithms, Structural stripes, Minimal shell, Machine learning, Biological pattern, Recurrent Neural Network (RNN)

1. Introduction

A lot of research about generation, analysis and applications of biological inspired stripe/strip patterns comes from computer graphics (Knöppel et al. [1]) and computational biology (Kondo and Asai [2], Shoji et al. [3]). Also, advances in graphics and in the 3d printing technology initiated a investigation of creation of metamaterials, resulting in structures extremely anisotropic, with high stiffness along channels, (Tricard et al. [4]) or differentiable stripe patterns that target materials with high stiffness contrast along locally controllable directions (Maestre et al. [5]). Other design strategies use vector- or cross-field integration methods to generate the geometry as well as the topology of the pattern, whose singularities correspond to the ones of the vector field. This feature-based exploration approach can be applied to the principal stress directions for mechanical efficiency or to the principal curvature directions for fabrication properties (Oval et al.[6]).

Again, the importance of directionality of the pattern, from the structural perspective is stated, as well as the need for conceptual and practical tools for topology finding patterns for structural design at the early stages of the project (Oval et al. [6]). Also, other design approaches use a machine learning

Copyright © 2024 by <name(s) of the author(s) as listed above>
Published by the International Association for Shell and Spatial Structures (IASS) with permission.

algorithm to speed up the topological design exploration of compression-only shell structures with planar faces, considering both structural performance (with graphic statics) and construction constraints (Zheng et al. [7],[8]), or a Strip-decomposable quadrilateral (SDQ) meshes aligned to user-defined directions, while also incorporating desirable properties to the strips for fabrication of minimal surfaces (Mitropoulou et al. [9]). Also a stress-line-based computation and materialization framework adopting a number of geometric criteria, minimizes the reliance on FEA (Tam and Mueller [10]). Here also a gap in stress-line-inspired design and tools is stated.

The advantages of structural stripes, a primary research premise of THEVERYMANY is that “the recombination of panels into stripes reduces the number of parts for assembly while also augmenting structural properties” (Fornes [11]). In the architectural design field, a lot has been investigated geometrically about discrete freeform surfaces like rectifying strip patterns (Wang et al. [12]). For shell structures, there are different approaches but few tools oriented to segmentation strategies especially for CNC flat materials with good cracking resistance, high bending fatigue life, light weight, good toughness, as Polypropylene (PP)[13], or similar. Also, most approaches use analytical ways of computation, which makes it difficult for architects to implement on projects as a pre-rationalization method (Stach [14]) in the early stage of the design process.

2. Methodology

Previous research in the field proposed a methodology to enhance the user’s intuition by examining the intersections between machine learning, physics-based simulations, structural performance and fabrication patterns of branching shell structures. The main objective of that experiment was to test the structurality of segmented thin shells of branching topologies so that they would self-support and withstand an additional weight apart from the material itself, considering at the same time the material usage and stripes as a construction logic. Specifically, the user’s learning goals to examine were: How the branching topology and spring strength affect the number of stripes, material usage and deflection. Finally, it was possible to build and validate a state of the art prediction mechanism of shell and pattern generation to reduce computational cost (Figure 1). The extracted data sets served as a first filter of visualising which attributes are affected and/or mostly affecting each other and to achieve a better understanding of which control parameters that define the geometric characteristic of the shell and pattern are influencing mostly the structural performance, material usage and number of stripes. In most cases the ANN gave accurate approximations, given a new branching topology, it could immediately calculate the fabrication statistics (Giannopoulou et al. [15], [16]).

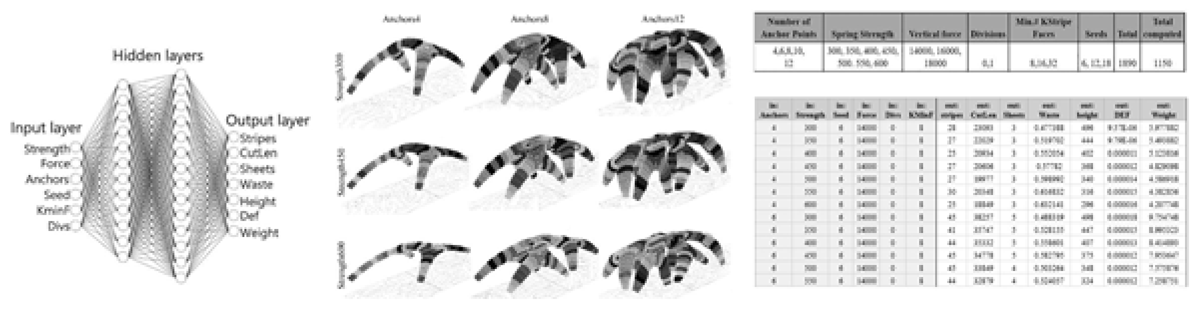

Figure 1. (left) Layout of the artificial neural network, with 6 nodes for the input layer, 12 and 14 nodes in the two Sigmoid hidden layers and 7 in the output layer. (mid and right) Data sets in the format of CSV data with the corresponding images. 1800 iterations and 1150 computed.

In this article we introduce a parallel experiment by using intersections between weighted-mesh representations (Nejur and Steinfeld [17]) and a Recurrent Neural Network (RNN) to explore the design space of segmentation patterns and the selection of an efficient organization of stripe patterns. The computational framework has the advantage of incorporating simplified design tools and procedures in a single parametric workflow which manifest a unified patterning system. The method produces data sets in order to analyse and evaluate, as a user learning process, parallel alternatives of segmentation patterns on a predefined geometry. The importance here is given to identify, by trial and error, the appropriate attributes, referring to the specific geometric characteristics inside the parametric model, which are more important based on the *performance reference criteria of structural stripes design*. The hypothesis is that the neural network may give more accurate results, if the training data involve only those.

We can characterize the performance reference criteria of structural stripe design for minimal shell structures as: *Orientation (direction), Amount of Stripes, Length (number of faces), Valence (bifurcation or not), Topology (open or closed), Angles between faces (stripe's curvature) Frequency, Width, Connectors, Boundary conditions, Inner/directional lines (curvature, stress, forces, etc), Manufacturing constraints, Material properties*.

2.1. Case Study

The current case study uses a machine learning approach to examine weighted meshGraph representations as design tools for hyperboloid topologies. Those types of topologies have the advantage of self support and multiplying in space in all directions. The relaxed mesh creates columns which can adapt to different scenarios. While previous experimental data sets included attributes of shell topological variations and dynamic relaxation parameters by assigning a weighted mass/stiffness to each node, here the input relaxed mesh geometry remains the same, keeping always the same number of faces, while variate the meshGraph weights on the nodes of the meshGraph. That produces variations of Minimal Span Trees, translated to stripe patterns. The meshGraph weights are unequally distributed for the boundary naked edges and the stress lines as directional lines (Figure 2).

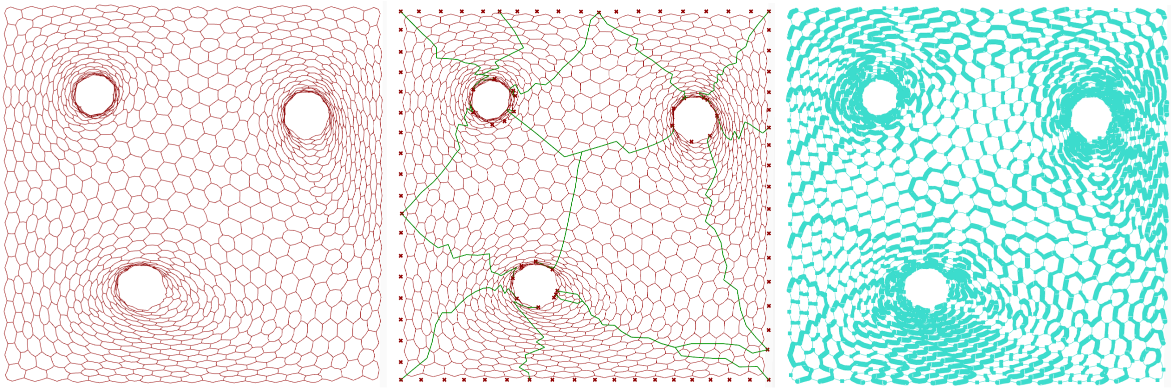

Figure 2. (left to right) MeshGraph from mesh, MeshGraph with adopted stress lines and naked edges, MeshGraph with applied weights.

2.1.1. Geometry Preparation

The predefined geometry is a three jointed hyperboloid shell structure. The pipeline follows as: Design of bottom and top boundary conditions > Design of surfaces based on adapted boundary division rules > Converse to quad meshes > Triangulate quad meshes > Relax mesh > Generate meshGraph > Introduce directionality with customized edge weights > Apply the Kruskal segmentation algorithms.

2.1.2. User Learning Goals

The goal of this experiment is to analyze the morphology of the pattern on a predefined shell skin. The learning goals to examine are: How targeted weights affect the pattern morphology and consequently the fabrication and structural performance. The learning process suggests that the combination of some attributes indicates lower deflection values and increased efficiency of the fabrication process which is balancing assemble time due to the connecting elements (stripes).

2.1.3. Machine Learning Goals

The main intuition behind the use of neural network models was to see the performance of such models in capturing dependencies and patterns in shell structure related data. The main goal was to predict combinations of the control lines on each of the three columns that perform better, given specific values of weights on the stress lines and naked edges, amount of stripes, overall structure weight, structural deflection and bifurcations.

For this task a Recurrent Neural Network (RNN) architecture was selected. RNNs are a pivotal advancement in the field of artificial intelligence, particularly when dealing with sequential data. Recurrent Neural Networks (RNNs) are a type of neural network architecture which is mainly used to detect patterns in data. Such data can be handwriting, text or numerical time series. Additionally, they are also applicable to images or videos if they get decomposed into a series of patches and treated as a sequence. (Schmidt [18]). The selection of such models for a geometric task stems in the fact RNNs excel in data that are interconnected and their position matters, even though time is not present in this specific geometric task spatial relationships between points, edges, and faces form sequences that RNNs are able to capture. As seen in tasks of 3D object recognition or reconstruction, in which the order points are processed matters. RNNs, with their ability to handle sequential data, can effectively model these spatial sequences and capture complex geometric patterns.

Geometric objects often exhibit hierarchical structures, such as nested shapes or multi-level representations. This makes them well-suited for tasks like parsing complex geometric scenes or understanding hierarchical relationships in geometric data. In 3D reconstruction tasks other works approach the problem by encoding symmetry characteristics of common objects while preserving long-range structural coherence, and describe objects of varying complexity with a compact representation. Such works that sequentially generated primitives, were based on sequentially generating handwriting strokes and the PixelRNN model that aimed at creating sequences of pixels that form a natural image (Chuhang Zou et al. [19]).

More specifically a many to many LSTM (Long Short-Term Memory) architecture was used, many to many LSTMs are a recurrent neural network architecture specifically designed to handle tasks where multiple input sequences are used to generate multiple output sequences. LSTMs are widely used in speech and language generation tasks. The key element of the LSTM is the use of memory cells that allow for constant error flow during training. Other recurrent neural networks are not able to assign the same credit to all inputs and therefore, are very limited to the solutions they will find. LSTM networks excel in considering all input information at each phase of learning, no matter where they are located in the input sequence. (Arras et al. [20]).

2.2. Dataset Description

Based on the structural stripes performance reference criteria, mesh segmentation algorithms are being explored in depth (Figure 3). The database was generated as a .csv file, alternating control lines and applied weights. Table 1 demonstrates a pre-sample of the database which include numerical values of weights on the stress lines (*Vertical weights*), weights on the naked edges (*Naked edge weights*), identification of stress lines and naked edges - referring to the control lines (stress lines, boundary lines) and the applied node weights - amount of stripes, stripe lengths (number of faces/nodes on meshGraph), shell average deflection values, node valence (bifurcation pattern) and total weights of the shell.

Table 1. Sample of database before processing. Columns in order: 1. *Vertical weights*, 2. *Naked edge weights,* 3-4-5 *Combination of stress lines and naked edges,* 6. *Amount of Stripes,* 7. *Deflection values*, 8. *Total weights,* 9. *Length of stripes*, 10. *Bifurcations.*

0.6	0.6	0	0	0	72	0.000723	4958.6429	1 To 56	60
0.8	0.6	0	0	0	68	0.000796	4924.569	1 To 54	64
1	0.6	0	0	0	70	0.000788	4925.7776	1 To 53	64
0.6	0.8	0	0	0	60	0.000616	4874.6937	1 To 57	28
0.8	0.8	0	0	0	65	0.000732	4987.5152	1 To 53	61
1	0.8	0	0	0	70	0.000797	4925.1913	1 To 58	65
0.6	1	0	0	0	60	0.000616	4874.6937	1 To 57	28
0.8	1	0	0	0	58	0.000633	4888.5981	1 To 57	29
1	1	0	0	0	68	0.000705	4971.3436	1 To 60	54

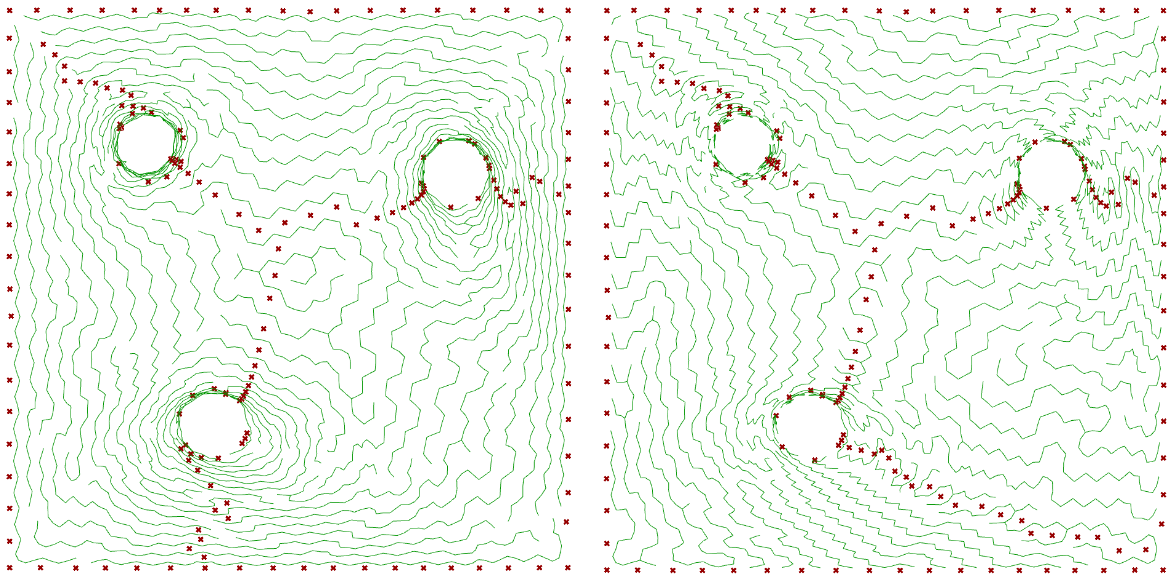

Figure 3. Sample images of mesh segmentations showing two different patterns. (Left) Verticals Weight 0.6, Naked Edges Weight 1. (Right), Verticals Weight 1 , Naked Edges Weight 0.6

2.3. Data preprocessing

Once the dataset is read from the .csv file all headers are removed and are converted into numpy arrays. To prepare the data to be fed for the LSTM model the dataset is split into input and output sequences. These sequences are of shape [576,6] as input with each row containing the data of Vertical edge weights, Naked edge weights, Amount of Stripes, Deflection values, Total weight and Bifurcations for prediction of the three stress lines that are given as the corresponding output sequences of shape [576,3]. Additionally due the variant values of the data, all data were normalized between [0-1] for the training process. After the data are divided into sequences they are reshaped as 3D arrays to be given to the LSTM input layer.

2.4 Model Architecture

The neural network model was built using the Keras API with TensorFlow backend. The model follows an encoder decoder architecture and consists of two Long Short-Term Memory (LSTM) layers with 240 neurons each (Figure 4). The higher dimensionality allows the network to learn more complex patterns in the data. The first LSTM layer serves as an encoder, capturing temporal dependencies and reducing the sequence dimensionality to a fixed-size representation followed by a RepeatVector layer that replicates this representation to match the original sequence length, in order to maintain the temporal structure of the input sequence when generating the output sequence. At the end of the encoder a dropout layer is added, with a dropout rate of 0.2. Dropout is added to prevent overfitting by randomly setting a portion of the input data to zero during training. This forces the network to learn more robust features improving generalization performance on unseen data. In this case the dropout layer with a dropout rate of 0.2 randomly sets 20% of the input units to zero during training, helping to prevent overfitting in the LSTM layers. Another LSTM layer acts as a decoder, reconstructing the sequence while preserving the learned temporal patterns. Finally, a Dense layer outputs the predictions. The model is compiled with the Adam optimizer and Mean Squared Error (MSE) loss function.

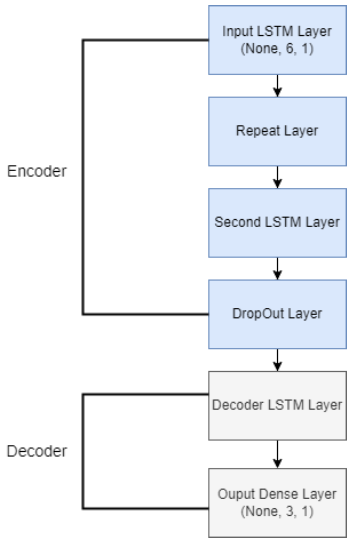

Figure 4. LSTM model architecture high level diagram

2.5. Model Training

As input data (x) the values of both vertical and naked edge weights, overall stripe number, overall structure weight, deflection and bifurcations were given. As output data (y) the combinations of the three different stress lines were given. The model was trained for a total of 30 epochs and the batch size is 3 samples per batch. The learning rate for the Adam optimizer was set to 0,001. The training progress is overseen by printing the Mean Squared Error (MSE) loss metric after each epoch. Additionally, the training data is split into training and validation sets, with 20% of the data reserved for validation to evaluate the model's performance on unseen data and prevent overfitting.

3. Results and Evaluation

Before training Pearson correlations coefficients were investigated between sets of data. The findings indicate that the *Vertical weights* are positively correlated with the *Deflection* by 0.4763, while the *Naked weights* and *Deflection* are negatively correlated by -0.543. On the other hand, both the *amount of Stripes* and *Deflection* as a pair are positively correlated with *Bifurcations* by 0.61763. The combinations of control lines, indicated with numbers from 0-3, since not continuous numbers but rather combinations, have no correlation with the rest of the data, as expected. So the model is trained to predict the most accurate combinations rather than continuous values.

During the training process, both training and validation loss were monitored using the WandbMetricsLogger callback. The Mean Squared Error (MSE) loss metric gradually decreased throughout the training process. In the first epochs, the loss values were relatively high, indicating that the model struggled to fit the training data well. However, as training progressed, MSE steadily decreased. The lowest point of the MSE loss was observed towards the end of the training process, particularly around epochs 19 and 20. This showcased that the model was learning to better predict the target values as training progressed. It's noteworthy that although the loss continued to decrease, it did not reach zero, indicating that there's still room for further improvement in the model's performance. Additionally, the validation loss, which measures the model's performance on unseen data, followed a similar trend, reflecting the model's ability to generalize. Overall, the decreasing MSE loss throughout the training process signifies the model's learning progress and its increasing capability to make accurate predictions. Further analysis and fine-tuning of the model might be necessary to improve its predictive capability and generalization to unseen data.

4. Conclusion

As a computational framework of modeling and evaluation of a construction system it permits continuous workflows which allow scientific data and methods to be implemented. As a qualitative approach of understanding shell and pattern behavior it has the potential to traverse domains preserving inside the morphogenetic process the intuitiveness of understanding and representation of a mechanism, while providing a way to build physical objects.

Additionally, in cases where geometry changes over time RNNs can be proven efficient since such geometry tasks involve temporal dependencies, where the geometric configuration changes over time. For instance, in analyzing the motion of objects during the application of forces on structures, past positions of points and edges affect future positions. RNNs would be very well suited at capturing such temporal dependencies, making them interesting models to indulge for tasks where understanding the evolution of geometric structures is important.

Acknowledgements

Research concept design: Baquero, Giannopoulou. RNN model architecture and training: Valsamidou

References

[1] F. Knöppel, K. Crane, U. Pinkall, and P. Schröder, "Stripe patterns on surfaces," *ACM Trans. Graph.*, vol. 34, no. 4, 2015.

[2] H. Shoji et al., "Origin of directionality in the fish stripe pattern," *Dev. Dyn.,* vol. 226, no. 4, pp. 627–633, 2003.

[3] S. Kondo and R. Asai, "A reaction-diffusion wave on the marine angelfish Pomacanthus," *Nature,* vol. 376, pp. 765–768, 1995.

[4] T. Tricard et al., "Freely orientable microstructures for designing deformable 3D prints," *ACM Trans. Graph.,* vol. 39, no. 6, 2020.

[5] J. M. Maestre, Y. Du, R. Hinchet, S. Coros, and B. Thomaszewski, "Differentiable Stripe Patterns for Inverse Design of Structured Surfaces," *ACM Trans. Graph,* vol. 42, no. 4, pp. 1–14, 2023.

[6] R. Oval, M. Rippmann, R. Mesnil, T. Van Mele, O. Baverel, and P. Block, "Feature-based topology finding of patterns for shell structures," *Autom. Constr.,* vol. 103, pp. 185–201, 2019.

[7] H. Zheng, V. Moosavi, and M. Akbarzadeh, "Machine learning assisted evaluations in structural design and construction," *Autom. Constr.,* vol. 119, no. May, 2020.

[8] H. Zheng, H. Mofatteh, M. Hablicsek, A. Akbarzadeh, M. Akbarzadeh, "Dragonfly-Inspired Wing Design Enabled by Machine Learning and Maxwell's Reciprocal Diagrams", *Advanced-Science*, vol.10, 2023.

[9] I. Mitropoulou, A. Vaxman, O. Diamanti, and B. Dillenburger, "Fabrication-aware strip-decomposable quadrilateral meshes," *CAD Comput. Aided Des.,* vol. 168, no. July 2023, 2024.

[10] K.-M. M. Tam and C. Mueller, "Stress Line Generation For Structurally Performative Architectural Design," in Proceedings of 35th ACADIA: Computational ecologies: design in the Anthropocene, 2015, pp. 95–109.

[11] M. Fornes, "The Art of the Prototypical," *Archit. Des.,* vol. 86, no. 2, pp. 60–67, 2016.

[12] B. Wang, H. Wang, E. Schling, and H. Pottmann, "Rectifying Strip Patterns," *ACM Trans. Graph.*, vol. 42, no. 6, pp. 1–18, 2023.

[13] *fullstarxm.com/customized-0-5mm-color-manufacturer-polypropylene-plastic-sheet-matte_p2*. [Accessed April 10, 2024]

[14] E. Stach, "Structural morphology and self-organization," in *Design and Nature V: Comparing Design in Nature with Science and Engineering,* 2010, vol. 138, pp. 29–40.

[15] E. Giannopoulou, P. Baquero, A. Warang, A. Orciuoli, and A. T. Estévez, "Stripe Segmentation for Branching Shell Structures: A Data Set Development as a Learning Process for Fabrication Efficiency and Structural Performance," in Architecture in the Age of the 4th Industrial Revolution - Proceedings of the 37th eCAADe and 23rd SIGraDi Conference, 2019, vol. 3, pp. 63–70.

[16] E. Giannopoulou, P. Baquero, A. Warang, and A. T. Estévez, "Computational Workflow for Segmented Shell Structures : an ANN Approach for Fabrication Efficiency," in IASS Symposium - Form and Force, 2019.

[17] A. Nejur and K. Steinfeld, *"Ivy- Bringing a Weighted-Mesh Representation to Bear on Generative Architectural Design Applications,"* in Proceedings of 36th ACADIA:Posthuman Frontiers: Data, Designers, and Cognitive Machines, 2016, pp. 140–151.

[18] Schmidt, R.M. "Recurrent Neural Networks (RNNs): A gentle Introduction and Overview". [Online]. Available: ArXiv, abs/1912.05911, 2019.[Accessed April. 10, 2024].

[19] C. Zou, E. Yumer, J. Yang, D. Ceylan, and D. Hoiem, "3D-PRNN: Generating shape primitives with recurrent neural networks." in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning,* LNCS volume 11700, Springer 2019.

[20] L. Arras, J.A. Arjona-Medina, M. Widrich , G. Montavon, M. Gillhofer, K. Müller, S. Hochreiter, and W. Samek, "Explaining and Interpreting LSTMs". [Online].Available: ArXiv, abs/1909.12114. 2019.[Accessed April. 10, 2024].