

A small-scale Artificial Intelligence module for the initial design phases of the machinery facility layouts in the shipbuilding industry

T W Akash¹ and O T Gudmestad^{1,2*}

¹ Western Norway University of Applied Sciences (HVL), Norway

² University of Stavanger (UiS), Norway

*Corresponding author: otgudmestad@gmail.com

Abstract. A deadline-based environment often degrades the quality of designs from a naval architect. This research proposes a non-human (AI) designer to automate preliminary marine designs, focusing on 2D topology optimization for ships' engine rooms. The study investigates design automation and its potential impacts on the maritime industry. The methodology involved technical datasets from built ships to train Artificial Neural Networks (ANN) and Generative Adversarial Networks (GAN) using machine learning (ML) platforms. Data from 152 real ships, 750 virtual ships, and 6,996 synthesized ships. Quality was assured using a Conditional Tabular Generative Adversarial Network (CTGAN), with results validated with regression methods & confusion matrices. Research findings were presented with training sessions, ML results, data validation, and industry impacts. In unsupervised sessions, the success rate for design generation was nearly 0%. After the supervised sessions, success rates improved to 15% for ANN, 0.06% for GAN, and 0.15% for CTGAN. Result validation showed that the primary AI modules are on a development trend despite low success rates. Research limitations include a relatively small dataset and feature loss in geometry and length of supervised learning. However, the ANN module demonstrated potential for future research scopes.

1. Introduction

Since a motor-driven ship propulsion completed its maiden voyage around one hundred and ten years ago, the 'Engine room' became one of the most important factors in the industry of ship design. Most of the explosive and inflammable goods are kept inside this specific area, which establishes this as the most dangerous enclosed space on a marine vessel. All the machines for power generation and distribution are the basic components of an engine room, which includes the engine and generator, rotatory devices for propulsion, pumps for circulation of fluids, ventilation, process and instrumentation design – piping systems (P&ID systems) and safety equipment for emergency purposes. These machinery items are cautiously considered in all further calculations, while a

vessel's primary design is prepared with respect to individual weights, location of centroid, and accessible service spaces according to maritime classification societies.

However, a ship construction project does not always become fruitful; sometimes ending up with a complete lack of success when there is a disagreement between the interested parties. Naval designers often work under professional liability, full of stressful hours utilizing their creativity on a project, without knowing that the project might not become a fruitful one for the later construction phase. These unsolved problems might lead a naval designer to the vicious path of '*Design fixation*' – a reluctance to produce creative designs due to a monotonous state of mind [1].

A neatly prepared technical design output from the designers is one of the basic requirements when organizations discuss their bilateral interests and financial aspects of a project. As a consequence, it is assumed that the primary works in the initial design phase should be less human-involving and more machine-oriented so that the human-dependent working hours in the early stage of design could be minimized. Involving Machine Learning (ML) to perform the primary design works might be one of the technical solutions. ML is the preliminary ground for developing an algorithm-based statistical platform that will read, analyze, predict, and convert numeric data into technical designs in an autonomous manner. This study develops and compares two autonomous AI models – an Artificial Neural Network (ANN) and a Generative Adversarial Network (GAN) – designed to generate designs without human intervention.

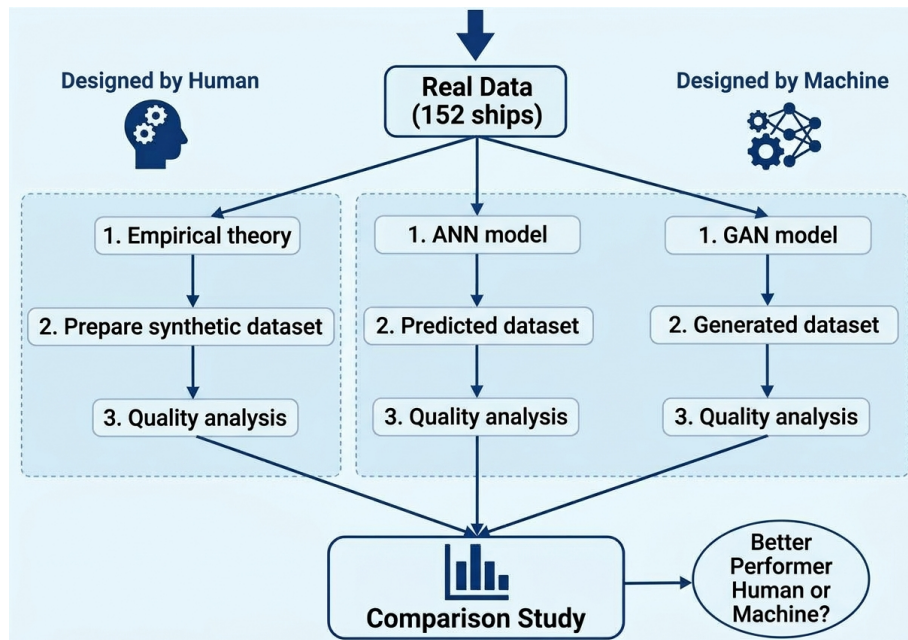


Figure 1. The structured workflow for the research concept.

Evaluating metrics like validation, error percentage, and generation time, the research builds upon prior merits of these automated ML systems – in other terms, a small-scale AI. The ANN and GAN modules are implemented across platforms including MATLAB, Python (Google Colab and GitHub), and Microsoft Excel (for dataset management and operation in a spreadsheet platform). Ultimately, the methodology and model outputs are compared across these environments as per the represented flowchart in Figure 1, with future research scopes highlighted in the concluding section of this research paper.

2. State-of-the-art

Since the year 2015, automation in ML has been playing a major role in numerous business sectors, specifically booming in the engineering industry [2]. Computer implementation of many statistical algorithms that were proposed earlier in the 20th century by prominent mathematicians and the development of feasible datasets with BIG DATA, i.e., extremely large and complex datasets that traditional data processing tools cannot easily analyze, have helped researchers generate mathematical resource libraries and numerous lines of code for AI development. ~~Many text-based AI tools have been demonstrated to serve their selective user groups, such as designers, programmers, entrepreneurs, engineers, and data scientists.~~ Historically, the primary emphasis in the optimization of ship design has been directed towards enhancing external physical performance and ensuring structural integrity. A considerable body of research has been devoted to the minimization of structural weight, manufacturing costs, and production timelines through traditional optimization methodologies applied to ship structures [30]. Similarly, the deployment of sophisticated computational methodologies is predominantly focused on hull form optimization and three-dimensional deformation techniques, which are specifically intended to augment hydrodynamic efficiency [31] of the ship's hull. The effective resolution of spatial boundary problems, which are governed by external fluid dynamics, is consistently achieved through a sustained focus, both previously and prospectively, on the external configuration of vessels [32] and their hydrodynamic characteristics during the preliminary design stage [33]. But there is a huge gap in the shipbuilding industry when it comes to generating CAD designs with the help of AI or ML. After a literature search on the Google Scholar website with the words 'Machine Learning' and 'Ship Designing' together, only one relevant paper out of 1050 research papers was found relating to the ML approach for ship designing in terms of CAD geometry optimization [3]. One possible reason for such unavailability of research works aligned to the present topic might be that complex spaces like engine rooms are governed by strict classification society constraints, machinery connectivity logic, and safety protocols rather than exterior hydrodynamic resistance, requiring an entirely different computational approach. When the existing body of literature discusses the internal spatial arrangement of marine vessels, it largely depends on conventional, non-generative computational methods. Detailed surveys of ship arrangement techniques indicate a significant dependence on traditional deterministic mathematical approaches instead of autonomous AI [34]. For example, researchers have frequently employed heuristic methods, such as Genetic Algorithms (GAs), to iteratively enhance and optimize the layouts of ocean-going ship compartments [35]. Additionally, modern methods use explicitly rule-based 3D parametric modeling systems to create general arrangement plans [36], while earlier seminal frameworks for intelligent ship arrangements utilized operations research and multi-objective optimization algorithms to address spatial configurations [37].

On the other hand, generative design is not a new terminology in the vast area of civil engineering and architecture of land-based infrastructure. ~~Generative Adversarial Networks (GANs) are used in some aspects of ship design, but most of these publications are limited to the generation of hulls (3D space optimization problem).~~ However, the topology optimization of 2D plans for internal arrangements remains vastly unexplored. An early research study conducted by two Japanese professors back in 1987, envisioning the conceptual development of an intelligent and automated system for generating the basic designs of ships at the primary levels of shipbuilding projects [4] – acts as the conceptual foundation for this research. The foundational work, established by Professors Akagi and Fujita, directly inspires the current research.

Although the Artificial Neural Network (ANN) and Generative Adversarial Network (GAN) architectures – that have been developed in the present research – are based on well-known deep

learning principles, their tailored application and structural adjustments in this research represent significant and novel contributions. In contrast to earlier studies that attempt to adapt traditional linear algorithms for spatial layout generation, a unique mathematical framework of interconnected parametric relationships has been designed to extract and convert physical CAD geometries into practical numerical matrices. Additionally, an innovative use of the Conditional Tabular Generative Adversarial Network (CTGAN) has been showcased in this paper. In previous works of other researchers, traditional GANs and CTGAN structures have been mainly employed for image generation or solely for synthetic text-data creation; however, we effectively utilized the advanced CTGAN framework to learn and synthesize non-imagery, tabular ship datasets. By converting spatial coordinates into tabular variables, we trained these sophisticated models to independently generate 2D engine room layouts. This establishes a fully data-driven novel approach that circumvents the limitations of traditional parametric restrictions, marking the first effective application of generative AI for optimizing marine facility layouts.

3. Theoretical framework

3.1. Selection of statistical methods

For design generation without human help, statistical models capable of handling non-linear data features are required. Regression-based and other linearization methods can cause unverified deviations in the prediction process. Specifically, in the current project, multi-objective optimization algorithms relying on the *Pareto front* (a boundary of possible and efficient solution points of a multi-objective optimization problem, the most probable region for a convergent solution) demonstrated infeasible solutions [5] for CAD conversions after 8,000 iterations. Because Artificial Neural Networks (ANN) are well-equipped for prediction operations with large datasets [6] and Generative Adversarial Networks (GAN) have two separate neural networks to supervise each other in synthetic data generation [7], these ANN and GAN modules were chosen for their proven particular capabilities. By synergizing Artificial Neural Networks (ANN) for robust numerical pattern recognition and Generative Adversarial Networks (GAN) for advanced tabular data generation, this study formulates a highly efficient, dual-model framework. Utilizing ANN-derived benchmarks for iterative GAN *Hyperparameter* (parameters or variables that are to be tuned with manual operation to find the best outcome of training) optimization significantly accelerates small-scale AI development within the naval design sector.

3.2. Artificial Neural Networks (ANN) and Levenberg-Marquardt Algorithm (LMA)

According to two old revolutionary research studies in the artificial intelligence domain as the foundational pillars in the early 20th century; firstly, McCulloch and Pitts developed the theoretical playground for machine learning by proposing the definition of *the activation function* (activates the sum of input vectors along with their respective bias values) for neural networks [8], and later, separately, further research was conducted by Rosenblatt [9]. The simplest form of mathematical expression for a *Perceptron* (a single neural network system with only one-layer neurons, the foundational unit of a large neural network) can be expressed as follows:

$$y = \begin{cases} 1 & \text{if } \sum_{i=1}^n x_i \geq \theta \\ 0 & \text{if } \sum_{i=1}^n x_i < \theta \end{cases} \quad (1)$$

$$f(x) = \theta(w_i \cdot x_i + b_i) \quad (2)$$

Where $y = f(x)$ is the output function, θ is the *Activation function*, w_i is the weight vector, x_i is the input vector, and b_i is the random bias. Building upon these two foundational neural network equations, subsequent research largely derived its methodologies, including the *Levenberg-Marquardt algorithm* (an iterative optimization technique to quickly train, especially, small-to-medium-sized feedforward networks). The Levenberg-Marquardt algorithm (LMA) was utilized in our research due to its superior training performance (i.e., speed and efficiency in data loss, etc.) over *Bayesian Regularization* (a mathematical technique used in ML in training artificial neural networks to prevent overfitting of data) for the datasets utilized. LMA [10] relies on the Sum Square Error (SSE). Derived from this error formulation, the Levenberg-Marquardt algorithm (LMA) updates the weight vector utilizing *Hessian* and *Jacobian* matrices, constituting the theoretical matrix operations exclusive to the artificial neural network (ANN) module in this study. There are different types of GAN models in the IT sector, depending on their working scopes and usability. The Generative Adversarial Network (GAN) and Conditional Tabular Generative Adversarial Network (CTGAN) architectures employ a distinct training paradigm; rather than relying on second-order matrix computations, these modules utilize stochastic gradient-based optimizers. This methodological divergence facilitates cross-validation of training efficacy and synthetic data generation outputs between the respective frameworks.

3.3. Conditional tabular generative adversarial networks (CTGAN)

A generative adversarial network consists of a combination of two neural networks (Generator and Discriminator) that compete in a minimization-maximization approach. Although the mathematical formulation of CTGAN derives from foundational principles concerning '*Perceptron*' and '*Activation function*', the comprehensive mathematical expressions for CTGAN exhibit heightened complexity and are elaborated in subsequent sections (i.e., Chapter 3, pp 28-30) of [11], on which the current work of this research paper is based. The conditional tabular variation (CTGAN) was adopted to handle non-imagery tabular data. The synthetic data generation workflow has been showcased for visual comprehension in Figure 2.

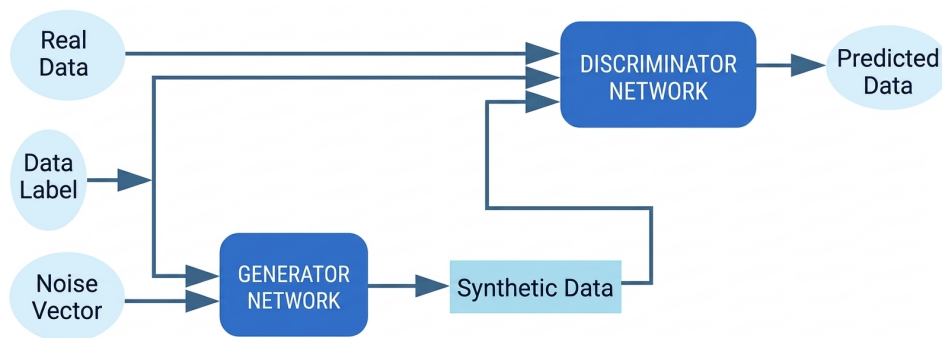


Figure 2. Simple working flowchart of the CTGAN network system (original flowchart retrieved from www.geeksforgeeks.com).

Following the methodology established by [11], this study adopts the foundational theorems governing the training and synthetic data generation of Generative Adversarial Networks (GANs).

The corresponding mathematical formulations implemented for experimental programming in the current research paper – including *Cumulative Loss Functions* (combination of generator and discriminator losses that propagates the adversarial training forward over time), *Optimal Training Criteria* (Nash equilibrium – where the generator network deceives the optimal discriminator network), *Helper Functions* (auxiliary utility supporting dataset analysis or synthetic output generation), and *Binary Cross-entropy* (a loss measuring discriminator classification of real inputs vs. synthetic results), etc. – are explained in Chapter 3, pp. 32 – 39 of [5] by the first author of this paper. In essence, the Generator discovers the underlying probability distribution with a given set of training samples. The optimization is executed utilizing the *Adaptive Moment Estimation* (ADAM) method, an adaptive algorithm that automatically adjusts training speeds to guide a model towards optimal performance.

4. Methodology

4.1. A ship's general arrangement represents a facility layout optimization

The ship design problem is mathematically portrayed as a 'Multivariable optimization' (a mathematical algorithm of finding the ideal combination of multiple input factors to achieve the best possible outcome) where different objective functions represent tasks related to machinery items (Figure 3). Client requirements and maritime classification society rules act as constraints. The scope of this research is constrained to these boundaries.

4.2. Parametric relationships formulation

A primary limitation of this study was the scarcity of suitable maritime datasets within established public repositories (i.e., GitHub, Kaggle, Hugging Face, etc.). Training advanced computational models, such as Artificial Neural Networks (ANN), Generative Adversarial Networks (GAN), and Conditional Tabular GANs (CTGAN), typically requires substantial data volumes to ensure robust convergence and generalization.

To address this data deficiency, an algorithm was developed to formulate various geometrical and mechanical ship parameters as dependent variables mathematically linked to two primary independent variables: Length Overall (LOA) and Block Coefficient (CB). Consequently, any given input for the ship's independent variables – LOA and CB – systematically yields the corresponding values for the remaining dependent variables, which are then structured into an $m \times n$ sized typical matrix (i.e., a '28x152' sized dataset matrix that we collected at the beginning of the research activity) for onward tabular data processing.

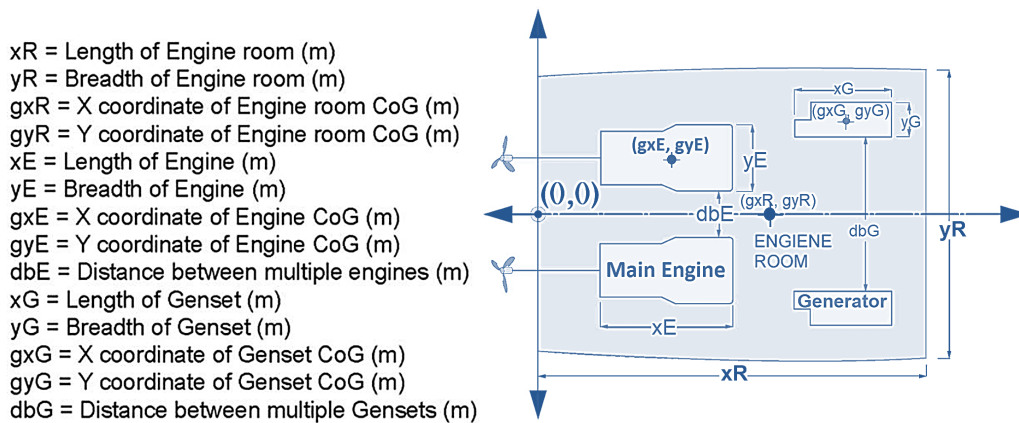


Figure 3. Physical representation of the engine room coordinate system.

The algorithm was implemented within a Visual Basic for Applications (VBA) spreadsheet environment. To prevent the generation of physically implausible or arbitrary synthetic data, a hybrid model was constructed. This model integrates established parametric relationships of ship particulars sourced from validated academic and professional maritime literature. Utilizing this verified framework, an initial synthetic training dataset comprising 6,996 virtual ship models was generated. This significantly augmented the original empirical dataset of only 152 ship records, thereby providing the requisite data volume necessary to enhance the predictive accuracy and reliability of the machine learning algorithms, which are demonstrated in Chapter 4, pp 42 to 49, in [5].

To provide an exemplary scenario for this methodology, ship's dimensional variables must be expressed in terms of independent variables, such as the ship's length overall (LOA) and block coefficient (C_B), in terms of facility layout optimization problems [12] in maritime and mechanical engineering as well. For example, the brake horsepower (BHP) of the ship's main engine is defined as a function of the ship's total resistance and velocity [5].

For instance, according to the steps mentioned in Chapter 4, pp 44-45 of [5], using a dataset derived from engine manufacturers' catalogs, the relationship between an engine's geometric particulars and its BHP (i.e., brake horsepower) was formulated logarithmically, Figure 4:

$$xE = |0.991 \ln (\text{BHP}) - 4.721| \quad (3)$$

$$yE = |0.416 \ln (\text{BHP}) - 1.578| \quad (4)$$

Similarly, the generator power (P_{GEN}) and dimensions were established as given in Figure 5:

$$xG = |0.776 \ln (P_{\text{GEN}}) - 1.084| \quad (5)$$

$$yG = |0.265 \ln (P_{\text{GEN}}) - 0.117| \quad (6)$$

Accordingly, the other 25 dependent variables – except the 'Serial number' column (i.e., ship's identification numbers as 1, 2, 3, ..., 151, 152, etc.), LOA and CB – were functionally interlinked through some modified parametric equations. The mathematical models adapted for this research were based on the foundational scholarly works written by various academic and corporate teams of researchers, such as [14 – 19], as well as established industry and regulatory guidelines [20 – 25]; all of these intellectual entities are internationally well recognized in the academic, shipping, and maritime engineering sector.

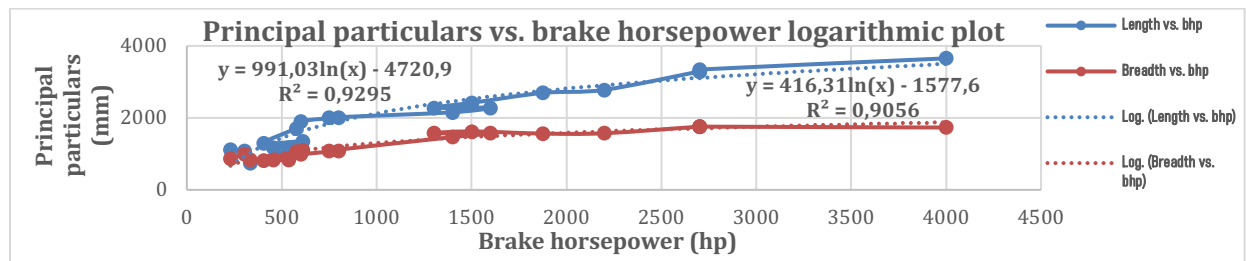


Figure 4. Engine particulars as a function of brake horsepower [13].

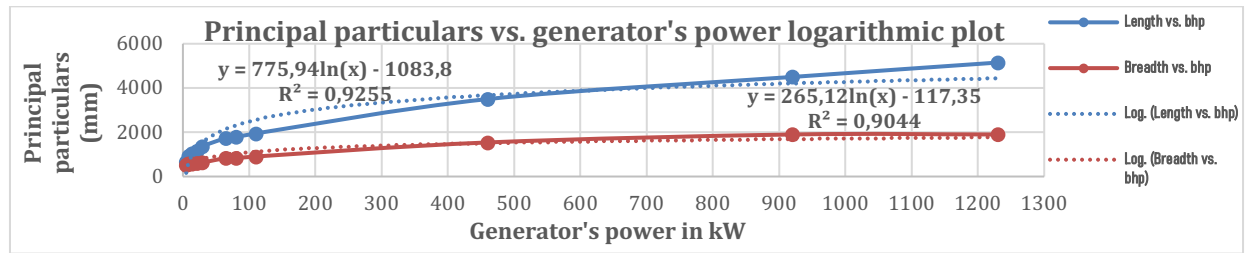


Figure 5. Generator particulars as a function of kW capacity [13].

Following the above methods, a mid-sized dataset (i.e., 6996 ship data) of simulated ship profiles was developed. As outlined in the subsequent section, this synthetic data was engineered specifically to train the selected deep learning frameworks, namely the Artificial Neural Network (ANN), Generative Adversarial Network (GAN), and Conditional Tabular GAN (CTGAN) models. However, all these dataset preparations began with organizing the data from 152 seagoing marine vessels.

4.3. Dataset preparation and augmentation

The procedures of numerical data extraction from a group of real ship design files started with a Rhinoceros (a computer software for computer-aided designing (CAD) – in architectural & mechanical engineering) 3D plugin (a user-specific extension of software that runs additional programs for advanced users) – 'Grasshopper' (a Generative-algorithm-based extension of Rhinoceros software that runs geometric operations with user-defined algorithms) – which was used to extract geometric coordinates of intended shipboard machinery items from the 2D CAD drawings (General Arrangement plans, Figure 6) of 152 real seagoing ships. However, at the outset of the research, the author manually verified the conversion of all scanned drawing formats (*.pdf) into vector formats (*.dxf). This preliminary preprocessing step was essential prior to implementing the computational workflows within the Grasshopper plugin. The plugin's algorithm performed CAD entity selection, bounding box incorporation, and mathematical extraction of lengths, breadths, areas, and centroids. The resulting [152x28] dataset provided the foundational artifacts for initiating the neural network model development. Due to the technical inappropriacy of the small-sized dataset containing only 152 samples for deep learning, augmented datasets were generated utilizing the formulated parametric relationships and a 'Random Distribution' function (i.e., this function maps the possible outcomes of a random variable to their respective probabilities) using a spreadsheet platform. A medium augmented dataset of 750 virtual ships and a larger dataset containing 6,996 synthetic ships were developed to prevent overfitting and test the stability of the models.

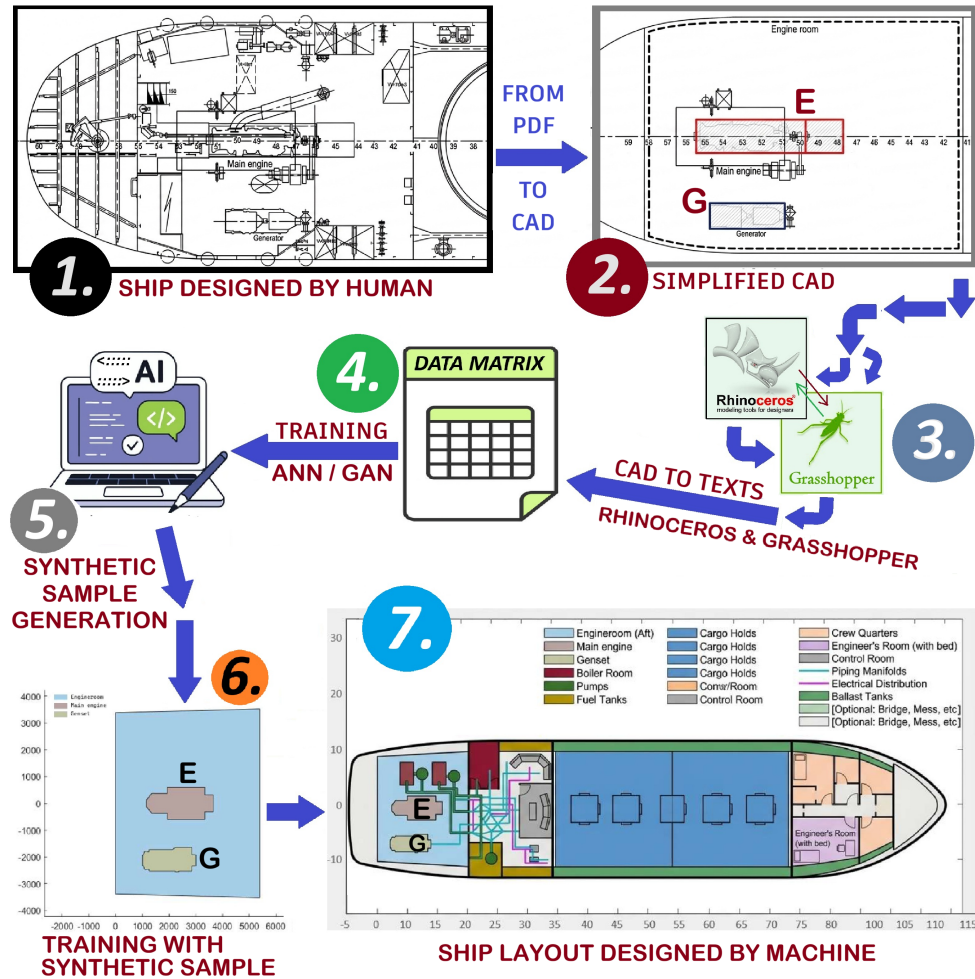


Figure 6. Dataset preparation & augmentation workflows in their respective order. Note: All the stock vectors are royalty-free; only the corporate logos of Rhinoceros and Grasshoppers are retrieved from their official website.

5. Development of AI models

5.1. ANN network architecture

ANN module was implemented in MATLAB utilizing a shallow-to-deep transition architecture. The dataset was partitioned into fifteen (15) predictor variables (inputs) and twelve (12) response variables (outputs), excluding the ship ID variable (integer). Although the network initially comprised ten (10) hidden layers, this architecture was subsequently refined to three (03) hidden layers to mitigate severe overfitting. The detailed processes and programming scripts are provided in [5], sections no. 5.1 and 5.2, pp. 58-68.

The '**trainlm()**' function (the applied computer program of the theoretical 'Levenberg-Marquardt Algorithm' in the MATLAB environment) executed the learning with the MSE as the performance evaluator. The dataset was divided into 60% training, 20% validation, and 20% testing data ratios – which are displayed in Figures 7a and 7b, accordingly.

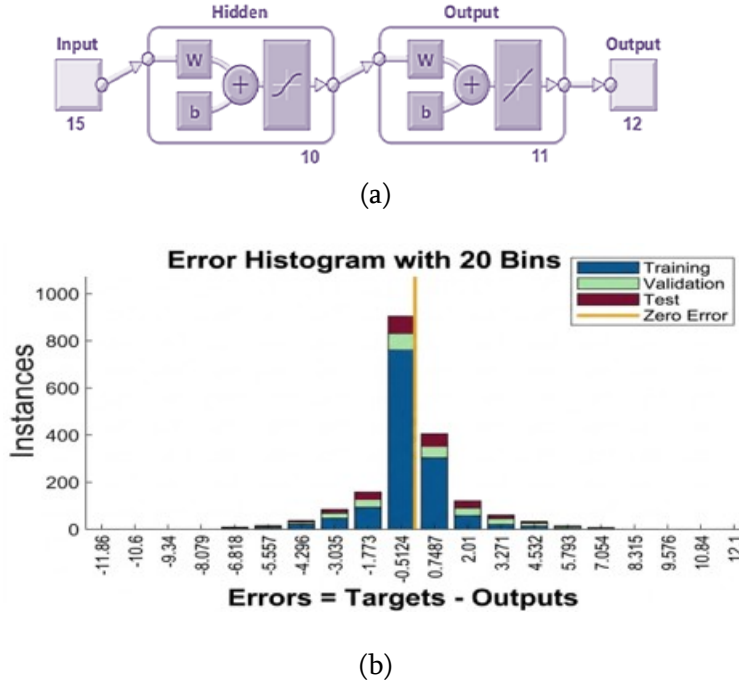


Figure 7. (a) ANN modules, network architecture and (b) ANN modules, training histogram in MATLAB, showing the deviation from target output

For instance, the error histogram in Figure 7b, visualizes the predictive accuracy of the ANN by displaying how the errors – the differences between the actual targets and the model's outputs—are distributed across the training (60%), validation (20%), and test datasets (20%). The tall concentration of data instances right at the central "Zero Error" line indicates that the model's predictions are highly accurate (overfitting), suggesting its hyperparameters (i.e., hidden layers) should be tuned accordingly.

5.2. GAN and CTGAN network architectures

The GAN module was scripted in the *MATLAB* programming environment (Figure 8), adopting the *ADAM Optimizer*, with hyperparameters including a mini-batch size of 27 and a gradient decay factor of 0.5. To cross-validate and circumvent local machine limitations, a Conditional Tabular Generative Adversarial Network (CTGAN) was implemented in *Python* via the *Google Colab* programming platform (a cloud-based programming platform where users can run any Python programs without dependency on different Python libraries and massive hardware requirements – which might not be achieved in local computers while analyzing big data). The CTGAN utilizes a complex setup comprising a Generator, Discriminator, and Residual neural networks [14] (See Figures 9 and 10) optimized specifically for discrete and continuous numerical tabular features.

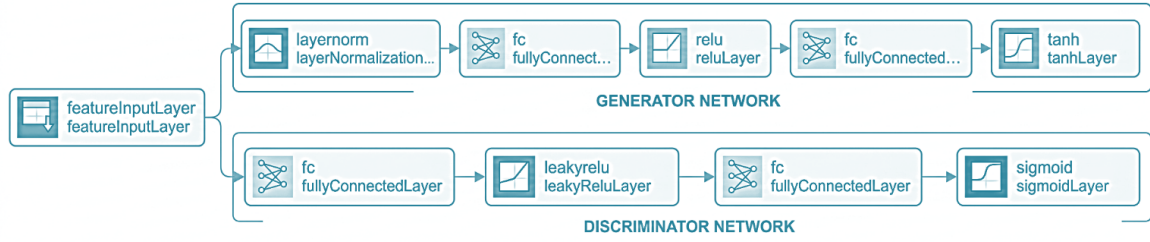


Figure 8. GAN network architecture deployment in MATLAB.

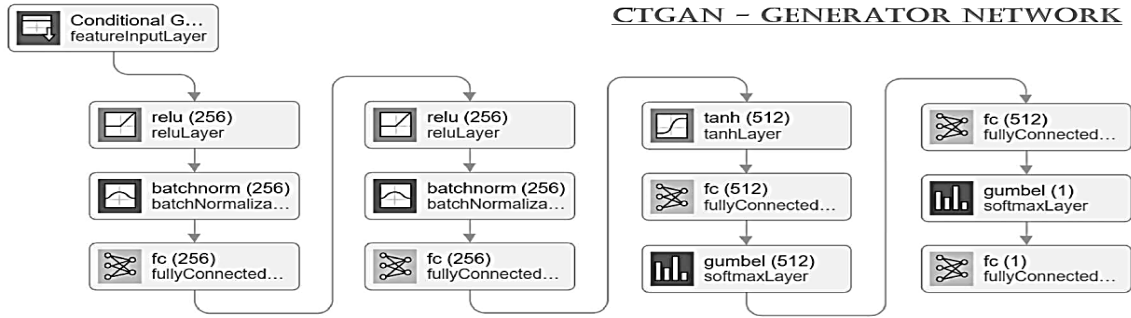


Figure 9. CTGAN network architecture deployment in Python.

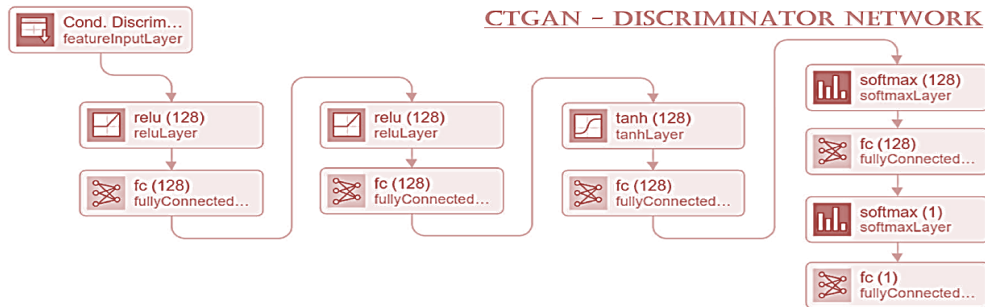


Figure 10. CTGAN network architecture deployment in Python.

The elaborated network architectures for the ‘*Generator*’ and ‘*Discriminator*’ sub-networks are provided in [5], chapter 5, section 5.3, pp 68–69 and section 5.6, pp 84–87, in detail for the peers and the readers of this paper as well. The first version of *CTGAN* module for ship’s engine room layout generation – developed by the first author – has been released for beta-testing at a publicly shared location [15] on Google’s cloud server. A standard network architecture adapted from *MATLAB* was trained utilizing the *ADAM* optimizer. The ‘*probDist()*’ (i.e., author-defined customizable script for calculation of probability distribution over ship data) hyperparameter was assigned a value of ≈ 0.035 , empirically derived via single-iteration *Monte Carlo* curve fitting on the foundational 152-ship dataset, for the networks shown in Figures 8–10.

This inverse relationship between probability distribution value vs. discriminator stability is manifested as the vanishing gradient problem [28] due to conflicting *Kullback-Leibler* (an asymmetric maximization approach to review the divergence between two probability distributions to increase the accuracy of the optimization model) and *Jensen-Shannon* (a symmetric minimization approach to find out the similarities between two datasets in generative adversarial networks (GAN) to enhance the learning phase of the model) divergence objectives. Ultimately, each training session

generated 150 synthetic ship profiles, which were subsequently filtered to isolate the 6–15 vessels per cycle that strictly satisfied practical marine design criteria according to the research objectives. The *GAN* exhibited significant hyperparameter sensitivity as per Table 1; notably, increasing *Monte Carlo* simulations induced Discriminator instability.

Table 1. Monte Carlo Simulations' Influence over the Training of the GAN Module

Number of simulations	Respective probability distribution value (Maximum)	Generator loss percentage	Discriminator loss percentage
1	0.035	67%	67%
100	0.185	<45%	<30%
1000	0.202	50%	<45%
10000	0.222	<45%	>55%

6. Results and discussion

During the dataset pre-processing, outlier cleaning reduced the dataset from $[6996 \times 28]$ to $[750 \times 28]$ virtual ships to fulfill *Google Colab's* computational limits, rapid training, and to prevent unexpected runtime restarts that could lead to progression loss. Subsequently, the AI modules were evaluated using a '*Confusion matrix*' (a classification system that evaluates an ML model's performance by comparing its predicted labels against the real values, i.e., Figure 11) – confirming that the module followed the expected development.

A direct comparison among the three deployed networks—Artificial Neural Networks (ANN), Generative Adversarial Networks (GAN), and Conditional Tabular Generative Adversarial Networks (CTGAN)—highlights significant performance variances; see Figure 12.

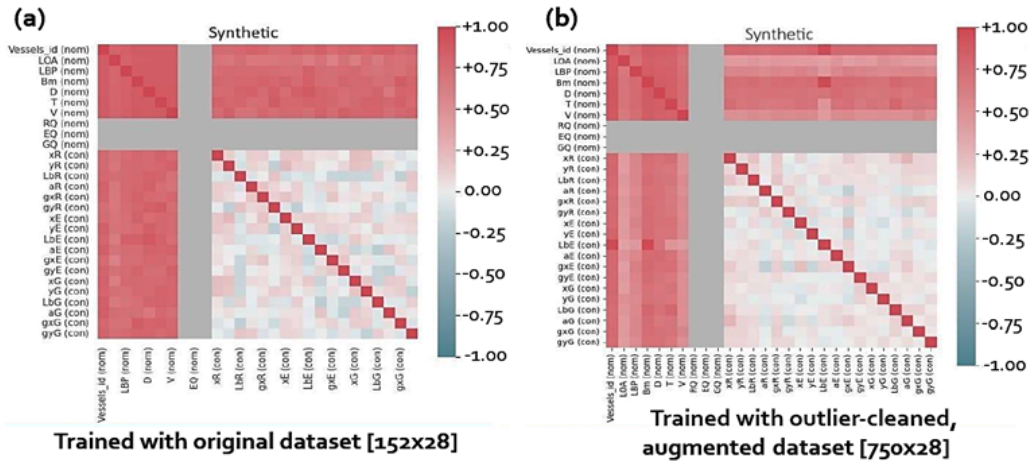


Figure 11. Confusion matrix comparison between (a) the original and (b) the augmented datasets.

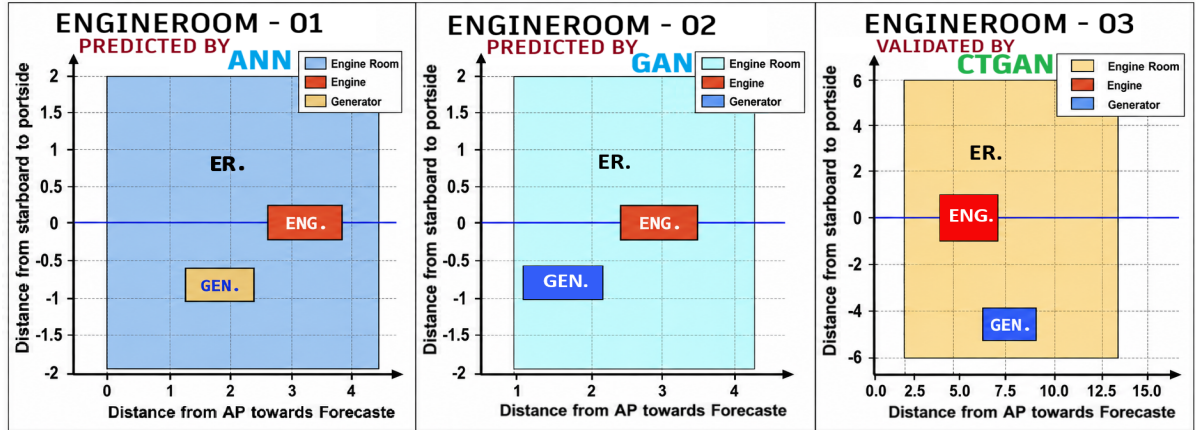


Figure 12. Comparison of the best synthetic engine-room outputs from ANN, GAN with CTGAN – having the same ship particulars data as input for these models.

Following extensive error testing, the CTGAN error testing was deemed out of scope for architectural debugging, as the module was developed by [14] and was utilized solely as a primarily validation-checking benchmark for comparison against the novel ANN and GAN modules – since there were not any other models available to assess those ANN and GAN models' predictive outcomes at the time of research. Afterwards, the primary validation-checking with CTGAN's forecasted engine room layout, arithmetical performance evaluations ultimately demonstrated that [5] the ANN module was the most successful, achieving a 15% success rate in generating valid synthetic ships, compared to 0.06% for the GAN module and 0.15% for the modified CTGAN model.

6.1. ANN & GAN networks validation

6.1.1. ANN module validation

Error testing and debugging of the ANN module were crucial steps in improving the model's reliability. In the initial unsupervised training session, the success rate of synthetic ship generation was nearly 0%. Before debugging, the training and testing states were highly unstable, yielding inadequate regression values (R-values).

However, after careful debugging procedures in the supervised training session, the performance improved significantly, boosting the success rate to 15%. The updated network architecture of the ANN module (Figure 13) and its corresponding error histogram (Figure 14) validate this structural improvement.

These enhanced outputs from the updated ANN module are visually confirmed by comparing the generated engine-room drawings. The initial predictions made before debugging (Figure 15) failed to fulfill ship design criteria. In contrast, the predictions after the network update successfully demonstrated the proper convergence of the engine and generator inside the engine room boundary.

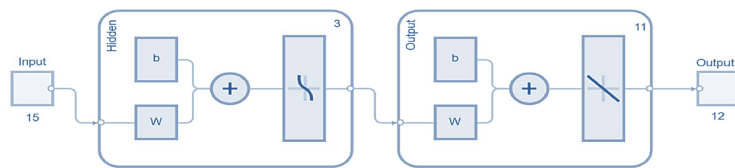


Figure 13. The updated network architecture of the ANN module.

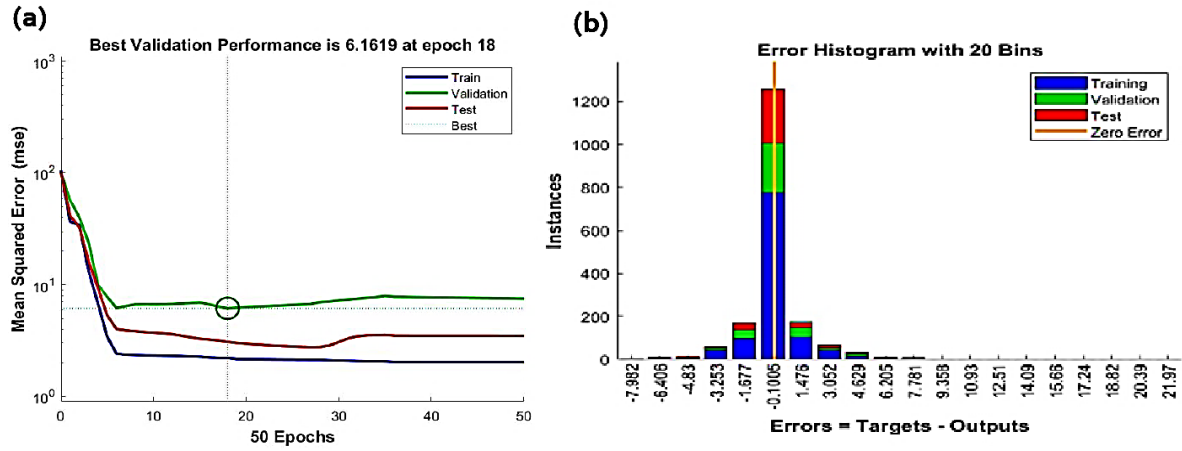


Figure 14. The (a) training performance and (b) error histogram of the updated ANN module.

Referring to [5] (Chapter 5, section 5.7, pp 99, Table 5.8) the baseline benchmark vessel specifications—including the ship's principal particulars (LOA = 15.00 m, LBP = 12.40 m, Bm = 4.60 m, D = 2.00 m, T = 1.40 m, V = 18.00 knots), the required quantities (RQ = EQ = GQ = 1), and the fixed engine room spatial parameters (xR = 4.13 m, yR = 4.14 m, LbR = 0.99 m, aR = 17.10 m, gxR = 2.07 m, gyR = 0.00 m)—alongside synthetic ship configurations predicted by the debugged artificial neural network (ANN) module. Across the three generated synthetic ship machinery room samples (Nr.1, Nr.2, and Nr.3), the ANN model yields consistent main engine spatial arrangements with longitudinal coordinates xE ranging from 0.79 m, 0.88 m, and 1.07 m, transverse coordinates yE from 0.30 m, 0.47 m, and 0.51 m, and engine lengths LbE bounded between 1.98 m and 2.19 m.

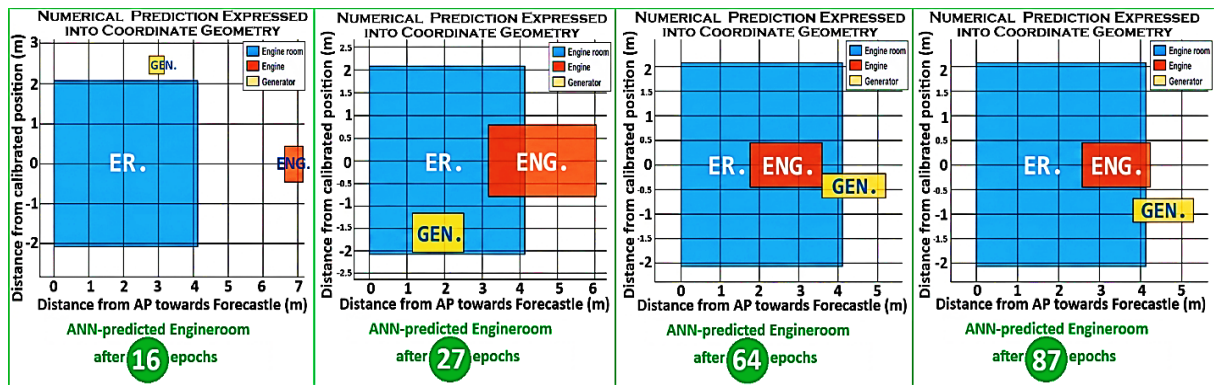


Figure 15. ANN model's drawing generation BEFORE debugging. (Refined for readability – based on the original results and figures from [5], Ch. 5, Sec. 5.1, pp 62-64)

On the other hand, the predicted generator attributes maintain low-dimensional variance with mean values of $xG = 0.85$ m, $yG = 0.48$ m, generator length $LbG = 1.98$ m, and offset coordinates $gxG = 1.52$ m and $gyG = -0.99$ m, demonstrating the debugged framework's (Figure 15) reliability in synthesizing structured tabular data arrays for machinery spatial optimization.

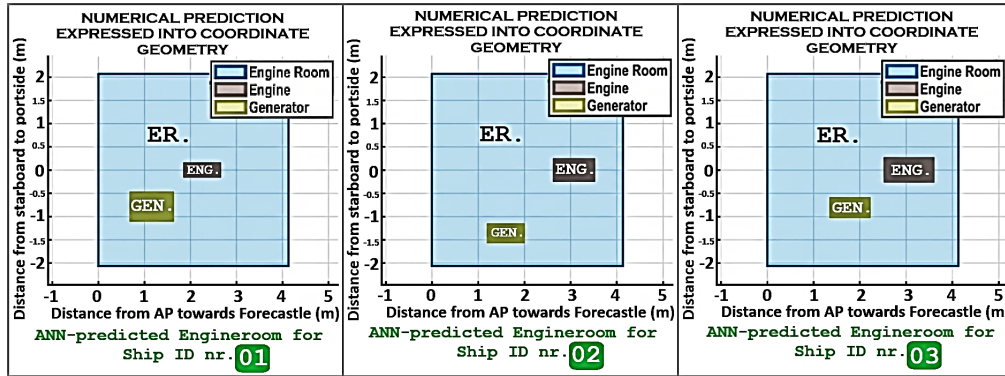


Figure 16. ANN model's drawing generation AFTER debugging. (Refined for readability – based on the original results and figures from [5], Ch. 5, Sec. 5.7, pp 100-101)

6.1.2. GAN module validation

Adapting MATLAB's image-processing GAN for tabular data initially caused severe convergence failures and loss instability. Debugging (Figure 17) revealed a critical setup flaw, which was corrected by increasing the validation frequency from 5 to 50 [5]. Consequently, the loss performance stabilized, and training convergence was successfully achieved.

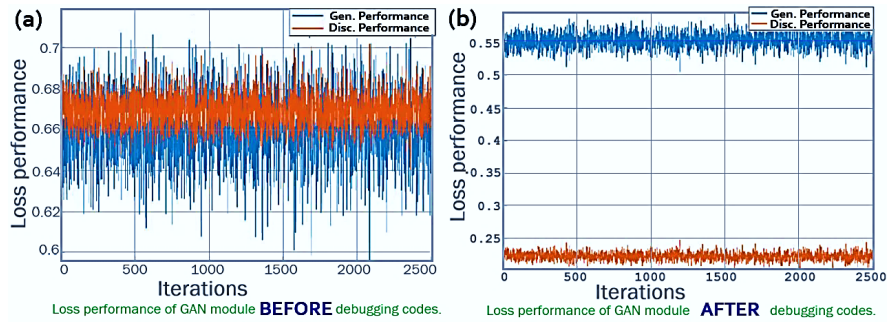


Figure 17. GAN modules' loss performance (a) before and (b) after the debugging operation.

The Generator loss (~55%) is more than twice the Discriminator loss (<25%)—a crucial indicator of effective GAN training [16]; thus, the updated module is expected to demonstrate improved performance in predicting synthetic ships within the MATLAB platform. The practical outcome of this debugging is evident in the generated ship engine-room CAD samples. The geometric plots before debugging yielded invalid spatial topologies. However, post-debugging, the GAN successfully reconstructed a true-positive synthetic ship (Figure 18) that met the basic criteria.

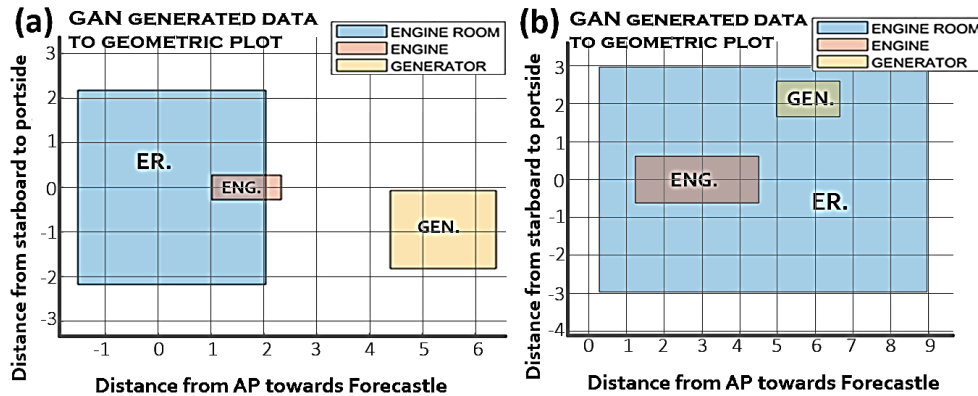


Figure 18. GAN-generated engine-room (a) before and (b) after the debugging operation.

6.2. Major limitations of the research

While the ANN module proved promising, several limitations must be addressed for future scalability, Table 2.

Table 2. Limitation aspects of the research

Major factors	Details
Lack of previous literature	A notable scarcity of previous research works regarding the direct application of AI to marine CAD generation.
Dataset size constraint	The original training dataset was relatively small, as it contains only 152 real ships.
Data degradation	Design feature loss occurred during the translation process from visual CAD geometries to tabular text (CAD-to-Text) conversions.
Code stability	Unresolved hidden bugs within the programming scripts.
Manual dependency	Supervised ML framework required a substantial duration of human involvement to tune the hyperparameters efficiently.
Lack of financial analytics	Unavailability of survey data from naval designers, shipbuilders, and clients to compare similar products from market history.

7. Conclusion and future scopes

This research successfully established a foundational, small-scale Artificial Intelligence framework capable of autonomously generating primary 2D CAD layouts for shipboard machinery. By comparative analysis of ANN, GAN, and CTGAN models, the study revealed that Artificial Neural Networks (ANN) are currently the most reliable architecture for predicting the tabular coordinate data required for topological optimization of engine rooms.

Although the success rates - peaking at 15% for the ANN module [5] - indicate that fully autonomous ship design is still in its infancy, the research proves that machine learning can mathematically replicate and satisfy basic general arrangement constraints.

If integrated into the early phases of maritime design, such tools have the potential to significantly reduce human-dependent working hours and mitigate the industry-wide problem of designer fixation. This current research was an implementation of a small-scale machine learning ensemble

of various modules. There are quite a few huge works to do to achieve the main objective of the research – developing the datasets, algorithms, codes, optimization constraints, etc. for every item for a marine vessel– regardless of its classes, types, and purposes. To make software, there is a bright scope for software developers to design a user interface for the non-programmer technical CAD designers. Besides, to upgrade these small-scale ML code snippets to a fully operational AI CAD designer, there is a lot of works to do in bridging an LLM (a Large Language Model) that assists people in explaining and generating their ideas into technical or artistic tasks to the interface of the software that is based on this research.

However, the immediate trajectory for this research is the transition from 2D space optimization to full 3D topology optimization. Expanding the algorithmic logic to include the Z-axis will allow the AI to account for the volumetric constraints of the hull form, vertical ventilation routing, and realistic multi-deck machinery placement. Integrating a larger empirical dataset of 3D point clouds, as per Figure 19, directly into a refined ANN model promises to yield a highly robust, commercially viable generative design tool for naval architects.

By pioneering a data-driven path to fill the vast literature void in automated maritime CAD generation, this study conceptually yet effectively reduces naval designers' professional design fixation by automating monotonous spatial hurdles through artificial intelligence. Ultimately, this research establishes a foundational milestone for intelligent shipbuilding, transforming early-stage layout generation from a repetitive manual burden into a streamlined, forward-looking paradigm.

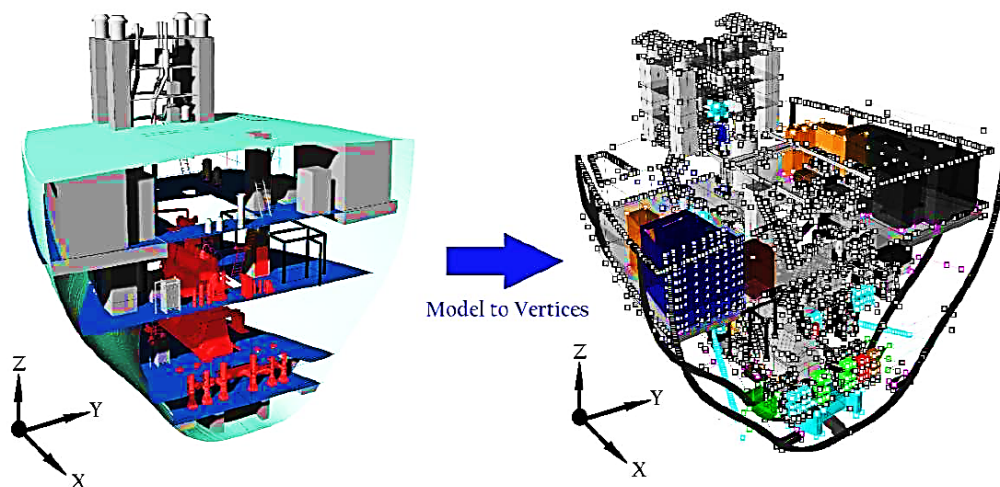


Figure 19. For a conceptual visualization of a random 3D engine room projected in the XY, XZ, and YZ planes – surface and curvatures converted to vertices (point-cloud) grids.

References

- [1] Crilly N 2015 Fixation and creativity in concept development: The attitudes and practices of expert designers. *Des. Stud.* **38** pp 54-91
- [2] Firican G 2022 The History of Machine Learning. *Lights On Data*
- [3] Huang L *et al* 2022 Machine learning in sustainable ship design and operation: A review. *Ocean Engineering* **266**
- [4] Akagi S and Fujita K 1987 Building an expert system for the preliminary design of ships. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing* **1** pp 191-205
- [5] Akash TW 2024 *MSc Thesis* Western Norway University of Applied Sciences (Haugesund)

- [6] Schmidgall S *et al* 2023 Brain-inspired learning in artificial neural networks: A review. *APL Machine Learning* **2** 021501 pp 1-14
- [7] Aggarwal A, Mittal M and Battineni G 2021 Generative adversarial network: An overview of theory and applications. *Int. J. of Inf. Mgmt. Data Insights* **1** pp 2667-0968
- [8] McCulloch WS and Pitts W 1943 A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Math. Biophys.* **5** pp 115-133
- [9] Rosenblatt F 1958 The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain. *APA* **65** 13602029 pp 386-408
- [10] Yu H and Wilamowski B 2011 Levenberg–Marquardt Algorithm *Industrial Electronics Handbook* (Levenberg–Marquardt Training Vol. **5**) 2nd edn (Alabama: CRC Press) pp 12.1-12.15
- [11] Goodfellow IJ *et al* 2014 Generative Adversarial Networks *Adv. Neural Inf. Process. Syst.* **3** pp 1-9
- [12] Daifuku M *et al* 2014 Design methodology using topology optimization for anti-vibration reinforcement of generators in a ship's engine room. *J. Eng. Mar. Environ.* **230** pp 221-226
- [13] Cummins Inc 2024 Marine Products Guide *Cummins*
- [14] Misra SC 2016 *Design Principles of Ships and Marine Structures* (New York: CRC Press)
- [15] Savitsky D 2003 On the Subject of High-Speed Monohulls. *Greek Section of the SNAME* p 2
- [16] Holtrop J and Mennen GGJ 1982 An approximate power prediction method. *Int. Shipbuild. Prog.* **29**
- [17] Molland AF *et al* 2017 Hull Surface Finish *Ship Resistance and Propulsion: Practical Estimation of Ship Propulsive Power*. (Reductions in Propulsive Power and Emissions, Ch 17) 2nd edn (Cambridge: Cambridge University Press) p 458
- [18] van Manen JD and van Oossanen P 1988 Resistance. *Principles of Nav. Arch.* (Alexandria USA: SNAME) **II** pp 72-118
- [19] Parsons MG 2003 Parametric Design. *Ship Des. & Constr.* (New Jersey: SNAME) **317** pp 11-16
- [20] IMO 2022 Guidelines on The Method of Calc. of the Attained EEDI for New Ships. *MEPC364(79)* p 10
- [21] IMO 2015 Adoption of The Int. Code of Safety for Ships Using Gases. *MSC391[95]*
- [22] Caterpillar Inc 2015 Appl. & Inst. Guide – Eng. Room Ventilation. *Power Syst.* **11** p 3
- [23] SOLAS 2022 Permeability *Regulation-7.3*
- [24] NMA 1990 Nordic Boat Standard – Commercial Boats less than 15 meters. S 3.3(d) p 24
- [25] Cruise Mapper 2015 Cruise Ship Engine Power, Propulsion, Fuel
- [26] Xu L *et al* 2019 Modeling Tabular data using Conditional GAN. *NeurIPS* **32** pp 7335-7345
- [27] Akash TW 2024 Conditional Tabular GAN for Synthetic Ship Data Generation. *Google Colab* (repository https://colab.research.google.com/drive/1u4MpTvHpAL_51liSvTDisg_pKOF62LE)
- [28] Lee JY and Choi SI 2020 Improvement of Learning Stability of Generative Adversarial Network Using Variational Learning. *MDPI Applied Sciences* **10** p 4528
- [29] Brownlee J 2021 How to Identify and Diagnose GAN Failure Modes. *Machine Learning Mastery*
- [30] Philippe Rigo P and Caprace JD 2011 Optimisation of Ship Structures *MARTECH* pp 925-944
- [31] Wei D, Wang L, Yang J, Sun X, Ji X and Liu X 2025 Hull form Optimization with a new three-dimensional deformation strategy *Nature – Scientific Reports* **15** 41804

- [32] Bertram V, Heimann J and Hochkirch K 2023 Ship hull optimization – past, present, prospects. *Ship Tech. Research* **71** pp 82-91
- [33] Aung MZ, Nazemian A, Boulougouris E, Wang H, Duman S and Xu X 2024 Establishment of a design study for comprehensive hydrodynamic optimisation in the preliminary stage of the ship design *Ships & Offshore Str.* **19** pp 793-806
- [34] Kim KS and Roh MI 2025 Review of ship arrangement design using optimization methods *J. of Comp. Des. & Eng.* **12** pp 100–121
- [35] Aditya V, Banerjee A and Venkataraman S 2021 Optimisation of Naval Ship Compartment Layout Design Using Genetic Algorithm *Proceedings of the Design Society* **1** pp 2339-2348
- [36] Padolecchia D, Marchesi NT and Bucci V 2025 3D Parametric Modeling for Ship Concept Design: A Flexible Approach to General Arrangement Plan *IOS Prog. in Marine Sc. & Tech.* **10** pp 1028-1037
- [37] Parsons MG, Chung H, Kate E, Kirtley N, Daniels AS, Liu S and Patel J 2009 Intelligent Ship Arrangements: A New Approach to General Arrangement *Naval Engrs. J.* **120(3)** pp 51-65