International Journal of Computer Applications Technology and Research
Volume 12–Issue 12, 341 - 352, 2023, ISSN:-2319–8656
DOI:10.7753/IJCATR1212.1030

Machine Learning Surroagte Models Replacing Physics Simulations

Love David Adewale
Department of Industrial
Engineering,
Southern Illinois University
Edwardsville,
USA

Abstract: Physics-based simulations underpin modern engineering and scientific discovery, enabling high-fidelity modeling of complex systems across aerospace, energy, materials science, climate modeling, and biomedical engineering. These simulations, often governed by partial differential equations and multi-physics coupling, provide accurate representations of real-world phenomena but are computationally intensive, time-consuming, and costly to scale. As design cycles shorten and real-time decision-making becomes essential, traditional numerical solvers such as finite element, finite volume, and computational fluid dynamics frameworks face limitations in high-dimensional optimization, uncertainty quantification, and digital twin deployment. Machine learning surrogate models have emerged as a transformative alternative, approximating the input–output behavior of physics simulations with significantly reduced computational cost. By learning mappings from simulation data, surrogate models such as Gaussian processes, neural networks, physics-informed neural networks, and operator learning frameworks can replicate simulation outputs in milliseconds once trained. These models enable rapid parameter sweeps, real-time control, sensitivity analysis, and design optimization while preserving acceptable accuracy. Hybrid approaches further integrate domain knowledge, enforcing physical constraints to enhance generalization and stability. This paper examines the theoretical foundations, architectural paradigms, performance trade-offs, and validation strategies for machine learning surrogate models replacing conventional physics simulations. It highlights practical implementation considerations, robustness challenges, and governance implications, emphasizing their role in accelerating innovation across computationally demanding scientific and industrial domains.
Keywords: Machine Learning Surrogates; Physics-Informed Neural Networks; Reduced-Order Modeling; Operator Learning; Computational Simulation Acceleration; Digital Twins

1. INTRODUCTION

1.1 Background on Physics-Based Simulation

Physics-based simulations underpin modern engineering design and scientific inquiry, providing numerical approximations to governing partial differential equations that describe physical systems [1]. Established techniques such as the Finite Element Method (FEM), Finite Volume Method (FVM), Computational Fluid Dynamics (CFD), and Monte Carlo simulations are widely adopted across aerospace, energy systems, structural mechanics, and materials science domains [2]. FEM enables discretization of complex geometries for stress and thermal analysis, while FVM and CFD are particularly effective in modeling conservation laws in fluid transport and reactive flows [3]. Monte Carlo methods further support uncertainty quantification and stochastic modeling by sampling probabilistic input distributions [4].

Despite their robustness, these methods introduce significant computational challenges when applied to multi-physics systems involving tightly coupled nonlinear interactions [5]. For instance, thermo-fluid-structural problems require iterative solvers and fine-grained discretization across both spatial and temporal domains, increasing solution time exponentially [6]. The curse of dimensionality further compounds this issue; as parameter dimensions grow, the number of required simulation evaluations scales combinatorially, limiting exhaustive exploration [7]. This growth in computational complexity constrains optimization, design iteration, and uncertainty propagation workflows, especially in high-resolution simulations [8].

1.2 Emergence of ML Surrogate Models

Surrogate modeling has emerged as a data-driven strategy for approximating high-fidelity simulations while drastically reducing computational overhead [2]. A surrogate model learns the functional mapping between input parameters and simulation outputs, effectively emulating solver behavior without explicitly resolving governing equations at inference time [4]. Traditional reduced-order modeling techniques rely on projection methods such as Proper Orthogonal Decomposition, which compress system states into lower-dimensional representations derived from physical insight [5]. However, machine learning–based surrogates adopt a statistical learning perspective, enabling nonlinear approximation through neural networks, Gaussian processes, or kernel-based regression frameworks [6].

Unlike solver-based iteration, trained surrogate models evaluate predictions through forward propagation, resulting in inference times that are several orders of magnitude faster than full numerical integration [3]. This computational efficiency makes them suitable for real-time control systems, digital twin environments, and embedded optimization loops [7]. Furthermore, machine learning surrogates can capture complex dependencies without requiring manual derivation of basis functions, offering flexibility in highly nonlinear parameter regimes [1]. The shift from physics solvers to learned approximators therefore reflects a transition from deterministic numerical resolution to probabilistic function estimation [8].

1.3 Research Objectives and Scope

The primary objective of this research is to assess the viability of replacing high-fidelity physics simulations with machine learning surrogate models while preserving predictive fidelity [5]. The study seeks to design, train, and validate surrogate architectures capable of generalizing across parameter spaces traditionally explored through expensive numerical solvers [6]. A structured comparison between simulation outputs and surrogate predictions is conducted using quantitative error metrics and computational efficiency benchmarks [3].

Another objective is to integrate statistical validation metrics such as mean deviation, root mean square error, and coefficient of determination to measure predictive consistency relative to physics-based standards [4]. Performance comparison also incorporates runtime analysis to evaluate speedup factors and scalability improvements [7]. The scope encompasses dataset generation from physics simulations, systematic feature engineering, supervised training, and structured testing under controlled conditions [2].

The contribution of this manuscript lies in presenting an end-to-end framework that combines computational efficiency analysis, statistical validation rigor, and methodological transparency, thereby offering a structured pathway for replacing conventional simulation workflows with data-driven surrogates in computationally demanding environments [1].

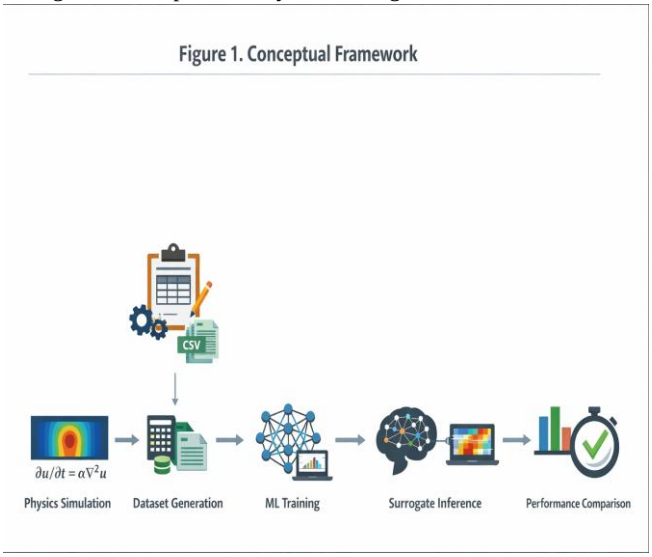

Figure 1. Conceptual Framework

2. LITERATURE REVIEW

2.1 Traditional Reduced-Order Modeling

Traditional reduced-order modeling (ROM) techniques were developed to alleviate the computational expense of high-dimensional numerical simulations while preserving essential system dynamics [6]. Among these, Proper Orthogonal Decomposition (POD) remains one of the most widely adopted approaches. POD identifies dominant modes from simulation snapshots by decomposing the solution space into orthogonal basis functions ranked by energy contribution [7]. By retaining only the most significant modes, the dimensionality of the system is substantially reduced without discarding primary dynamic behavior. This compression enables faster evaluation while maintaining acceptable physical fidelity for moderate parameter variations [8].

Galerkin projection is commonly employed in conjunction with POD to derive reduced governing equations [9]. In this framework, the high-dimensional governing equations are projected onto the reduced basis, resulting in a lower-order dynamical system that approximates the original solution space. The method preserves physical structure but can become unstable for strongly nonlinear systems or when parameters deviate significantly from the training domain [10]. Although ROM techniques provide interpretable physics-consistent approximations, their dependence on linear subspace representations limits their ability to generalize across highly nonlinear regimes [11].

2.2 Gaussian Process Surrogates

Gaussian Process (GP) regression offers a probabilistic surrogate modeling approach that provides both predictions and associated uncertainty estimates [12]. A GP models the target function as a distribution over functions defined by a mean function and covariance kernel, enabling flexible nonlinear approximation without explicit parametric assumptions [6]. Kernel selection plays a central role in capturing smoothness and correlation structure within the data.

GP surrogates are particularly advantageous in low-to-moderate dimensional parameter spaces, where they achieve strong predictive accuracy with limited training samples [13]. Their probabilistic formulation allows uncertainty quantification, which is critical for risk-sensitive applications such as structural reliability analysis or stochastic flow modeling [7]. However, computational complexity scales cubically with the number of training points, restricting scalability in large simulation datasets [14].

2.3 Neural Network-Based Surrogates

Neural network architectures have emerged as flexible function approximators capable of learning complex nonlinear mappings between simulation inputs and outputs [8]. Multi-Layer Perceptrons (MLPs) represent the foundational architecture, consisting of fully connected layers that approximate continuous functions through nonlinear activation transformations [9]. MLP-based surrogates are particularly effective in parameter-to-scalar mappings, such as predicting aerodynamic coefficients or thermal response metrics [12].

For spatially distributed systems, Convolutional Neural Networks (CNNs) provide enhanced performance by leveraging local spatial correlations [10]. CNN-based surrogates have demonstrated success in approximating fluid velocity fields and stress distributions, as convolutional filters extract hierarchical spatial features from discretized domains [11].

Temporal systems, including dynamic simulations governed by time-dependent PDEs, often require sequence modeling architectures. Long Short-Term Memory (LSTM) networks capture temporal dependencies by regulating information flow through gating mechanisms [13]. LSTM-based surrogates are particularly suitable for transient simulations where state evolution must be preserved across time steps [6]. Despite

their flexibility, neural networks require large datasets and careful regularization to prevent overfitting in high-dimensional regimes [14].

2.4 Physics-Informed Neural Networks

Physics-Informed Neural Networks (PINNs) integrate governing equations directly into the learning process by embedding PDE constraints within the loss function [7]. Rather than relying solely on data supervision, PINNs enforce physical consistency by minimizing the residual of the governing equation evaluated through automatic differentiation [8]. This hybrid approach reduces reliance on extensive labeled datasets while preserving adherence to conservation laws and boundary conditions [12].

Two primary strategies are employed: soft constraints and hard constraints. Soft constraints incorporate PDE residuals as penalty terms in the loss function, allowing flexibility but requiring careful weighting between data and physics terms [9]. Hard constraints, in contrast, embed boundary conditions or conservation principles directly into the network architecture, guaranteeing strict compliance but potentially reducing modeling flexibility [10].

PINNs demonstrate improved extrapolation capabilities compared to purely data-driven surrogates, particularly when training data is sparse [13]. However, optimization can be challenging due to stiffness in multi-term loss functions and sensitivity to hyperparameter selection [11]. Existing surrogate approaches face trade-offs between scalability, interpretability, uncertainty quantification, and strict physical consistency [14]. A comprehensive framework that integrates efficiency benchmarking, statistical validation, and robustness comparison against physics solvers remains underdeveloped [6].

3. THEORETICAL FOUNDATIONS

3.1 Governing Physics Equation (Primary Simulation Model)

To establish a reference benchmark for surrogate approximation, we consider the classical heat diffusion equation, a representative parabolic partial differential equation widely used in thermal and transport modeling [11]. The governing form is expressed as:

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u$$

where $u(x,t)$denotes the temperature field (state variable) and αrepresents the thermal diffusivity coefficient governing diffusion rate [12]. The Laplacian operator $\nabla^2 u$captures spatial curvature and heat flux propagation within the domain. This equation models conductive heat transfer under assumptions of isotropic and homogeneous media.

To obtain numerical solutions, discretization is required. In the Finite Difference Method (FDM), spatial derivatives are approximated using central difference schemes. For a one-dimensional grid with spacing Δx, the second derivative becomes:

$$\nabla^2 u \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2}$$

Temporal derivatives are often discretized using forward Euler schemes [13]. Alternatively, the Finite Element Method (FEM) reformulates the PDE in weak form by multiplying with test functions and integrating over the domain, resulting in a system of algebraic equations derived from basis function expansion [14]. Both FDM and FEM yield large linear or nonlinear systems requiring iterative solvers. As mesh resolution increases for accuracy, computational cost grows substantially, motivating the search for surrogate replacements [15].

3.2 Surrogate Model Representation

The surrogate model aims to approximate the mapping between simulation inputs and outputs without explicitly solving the PDE at inference time [16]. Formally, the surrogate function is defined as:

$$\hat{y} = f_\theta(x)$$

where $x$represents input parameters (e.g., boundary conditions, material properties, spatial coordinates), $\hat{y}$denotes predicted system response, and θencapsulates learnable parameters of the model [17].

In practice, $f_\theta$may correspond to a neural network or kernel-based regressor trained on simulation-generated datasets. The objective is to learn a function that minimizes predictive error across the parameter space explored by the physics solver [18]. Unlike direct discretization methods, surrogate inference requires only forward evaluation, significantly reducing runtime complexity. This transformation effectively shifts computational burden from repeated numerical integration to a one-time training phase [19].

3.3 Loss Function Formulation

Model training requires an objective function quantifying prediction error. The Mean Squared Error (MSE) is commonly adopted:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where $y_i$denotes ground-truth simulation output and $\hat{y}_i$the surrogate prediction [11].

The MSE can be interpreted through the expectation operator. If prediction error is defined as $\epsilon = y - \hat{y}$, then

$$MSE = \mathbb{E}[\epsilon^2]$$

This formulation links empirical loss to expected squared deviation under the data distribution [12].

The bias-variance decomposition further clarifies error sources:

$$\mathbb{E}[(y - \hat{y})^2] = Bias^2 + Variance + \sigma^2$$

International Journal of Computer Applications Technology and Research
Volume 12–Issue 12, 341 - 352, 2023, ISSN:-2319–8656
DOI:10.7753/IJCATR1212.1030

where bias measures systematic deviation from the true function, variance captures sensitivity to training data fluctuations, and σ^2denotes irreducible noise [13]. Minimizing MSE therefore balances approximation accuracy and generalization capability. Excessively flexible models reduce bias but increase variance, while overly constrained models exhibit high bias [14].

3.4 Regularization Term

To mitigate overfitting, regularization is introduced into the objective function. L2 regularization modifies the loss as:

$$L = MSE + \lambda \sum || \theta ||^2$$

where λcontrols the strength of the penalty term [15]. The squared norm $|| \theta ||^2$discourages large parameter magnitudes, effectively smoothing the learned function [16].

From an optimization perspective, L2 regularization constrains parameter growth and reduces model variance by shrinking coefficients toward zero [17]. This prevents excessive sensitivity to training noise and enhances generalization across unseen parameter configurations [18]. The hyperparameter λmust be selected carefully; excessive penalization increases bias, whereas insufficient penalization may lead to unstable extrapolation behavior [19].

3.5 Physics Constraint Integration

Beyond purely data-driven loss functions, physical consistency can be incorporated via residual penalties. The physics residual loss is defined as:

$$L_{physics} = \| \frac{\partial u}{\partial t} - \alpha \nabla^2 u \|^2$$

This term enforces adherence to the governing PDE by minimizing residual magnitude at sampled collocation points [11]. Incorporating physics-based constraints stabilizes training, improves extrapolation, and reduces dependence on large labeled datasets, thereby bridging numerical simulation and statistical approximation frameworks [18].

4. DATA ACQUISITION

4.1 Physics Simulation Data Generation

The dataset used to train the surrogate model was generated from high-fidelity numerical simulations of the transient heat diffusion equation described previously. The primary objective of the data generation stage was to create a structured mapping between physical input parameters and resulting temperature distributions. A systematic parameter sweep was designed to explore variability across diffusion coefficients, boundary conditions, initial temperature distributions, and spatial discretization scales [17]. The diffusion coefficient $\alpha$was varied within a physically meaningful range corresponding to common engineering materials, while boundary temperatures were sampled across controlled thermal gradients representative of industrial heating processes [18].

Boundary conditions were imposed as Dirichlet constraints at domain edges, specifying fixed temperatures, while Neumann boundary conditions were optionally introduced to simulate insulated or flux-driven surfaces. Temporal simulation windows were fixed but discretized at sufficiently small time steps to capture dynamic transitions without numerical instability [19].

To ensure representative coverage of the parameter space while minimizing redundant evaluations, Latin Hypercube Sampling (LHS) was adopted [20]. LHS divides each parameter range into equal probability intervals and randomly samples without replacement, guaranteeing stratified coverage across multidimensional space. This approach reduces clustering common in purely random sampling and improves statistical efficiency for surrogate training [21].

Simulations were executed using a validated finite element solver implemented in COMSOL Multiphysics, with automated batch execution via MATLAB scripting. Output fields were exported in structured numerical arrays and stored in comma-separated value (CSV) format for downstream processing.

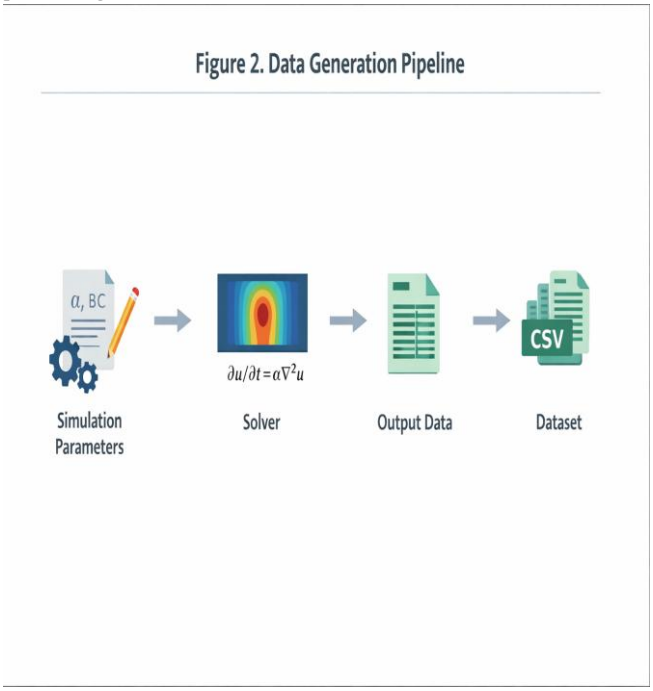

Figure 2. Data Generation Pipeline

4.2 Dataset Description

The resulting dataset consisted of structured input–output pairs derived from 5,000 simulation runs. Each input vector $x$comprised physical and boundary parameters, including diffusion coefficient α, boundary temperature values, domain length, and time index. Additional features included discretized spatial coordinates to enable learning of field-based outputs [22].

Output variables corresponded to nodal temperature values extracted from the finite element mesh at selected spatial sampling points. For dimensional consistency and computational feasibility, mesh resolution was reduced through spatial downsampling while preserving representative gradients [23].

The dataset dimensionality therefore included approximately 8 input features per sample and 100 output temperature values

representing spatial nodes at a fixed time step. This mapping enabled the surrogate model to approximate spatial temperature profiles directly from physical parameters. The overall dataset size exceeded 500,000 individual data points after spatial flattening, providing sufficient diversity for training deep learning architectures [24].

4.3 Data Preprocessing

Prior to model training, preprocessing was performed to standardize input features and stabilize numerical optimization. Feature normalization ensures that variables with larger magnitudes do not dominate gradient updates during training [18]. Standardization was applied using the transformation:

$$x' = \frac{x - \mu}{\sigma}$$

where μrepresents the sample mean of the feature and σdenotes the standard deviation [19]. The mean $\mu$is computed as

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

and quantifies the central tendency of the data. The standard deviation $\sigma$is defined as

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

which measures dispersion around the mean [20].

By centering data around zero and scaling by variance, the optimization landscape becomes smoother, improving convergence stability and reducing oscillatory gradient behavior [21]. Output variables were similarly scaled to maintain proportional error weighting across spatial nodes. Missing values were not present due to deterministic simulation outputs; however, outlier screening was conducted to detect potential solver instability cases. This structured preprocessing ensured numerical robustness and improved generalization capability during surrogate model training [22].

5. FEATURE ENGINEERING

5.1 Dimensionality Reduction

High-fidelity simulation outputs often produce high-dimensional feature spaces, particularly when spatial fields are flattened into vectorized forms. To reduce computational complexity while preserving dominant variability patterns, Principal Component Analysis (PCA) was employed as a dimensionality reduction technique [22]. PCA projects the original feature matrix $X \in \mathbb{R}^{n\times p}$onto a lower-dimensional subspace defined by orthogonal directions of maximum variance. The transformation is expressed as:

$$Z = XW$$

where Wcontains the principal component loading vectors and Zrepresents the reduced-dimensional representation [23].

The matrix $W$is obtained through eigen decomposition of the covariance matrix $\Sigma = \frac{1}{n}X^T X$. Specifically, eigenvalues $\lambda_i$and eigenvectors $v_i$satisfy

$$\Sigma v_i = \lambda_i v_i$$

The eigenvectors corresponding to the largest eigenvalues capture directions of maximal variance and are selected to form W[24]. Retaining only the top $k$components preserves most of the variance while reducing dimensionality from $p$to k, where $k \ll p$. This approach reduces model training time, mitigates collinearity among features, and enhances numerical stability in downstream learning tasks [25].

5.2 Derived Physical Features

Beyond raw simulation parameters, derived physical features were engineered to encode domain knowledge and improve surrogate learning efficiency. One such feature is the energy conservation term derived from the integral form of the heat equation. For a domain Ω, total thermal energy can be approximated as

$$E = \int_{\Omega} u(x,t)\, d\Omega$$

This scalar summary captures global thermal state behavior and provides a physically interpretable feature linked to conservation principles [26].

Another derived feature is the gradient magnitude of the temperature field, representing spatial intensity of heat flux. The gradient magnitude is computed as

$$|\nabla u| = \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2}$$

This feature reflects local variability and highlights regions of rapid thermal change. Including gradient-based descriptors improves surrogate sensitivity to spatial heterogeneity and boundary-driven dynamics [23]. By embedding physically meaningful quantities into the feature set, the model is guided toward representations consistent with thermodynamic behavior, reducing purely statistical approximation errors [27].

5.3 Feature Importance Analysis

Understanding the contribution of individual features to surrogate predictions is critical for interpretability and robustness assessment. Two complementary techniques were adopted: SHAP (Shapley Additive Explanations) values and permutation importance [24].

SHAP values are grounded in cooperative game theory and quantify the marginal contribution of each feature to model output by averaging over all possible feature coalitions. Formally, SHAP computes feature contributions based on Shapley values, ensuring fairness and consistency in attribution [25]. This approach provides both global and local

International Journal of Computer Applications Technology and Research
Volume 12–Issue 12, 341 - 352, 2023, ISSN:-2319–8656
DOI:10.7753/IJCATR1212.1030

interpretability, revealing which physical parameters most strongly influence temperature predictions.

Permutation importance, in contrast, measures performance degradation when a feature's values are randomly shuffled. A significant drop in predictive accuracy indicates high feature relevance [26]. Unlike SHAP, permutation importance is model-agnostic and computationally efficient.

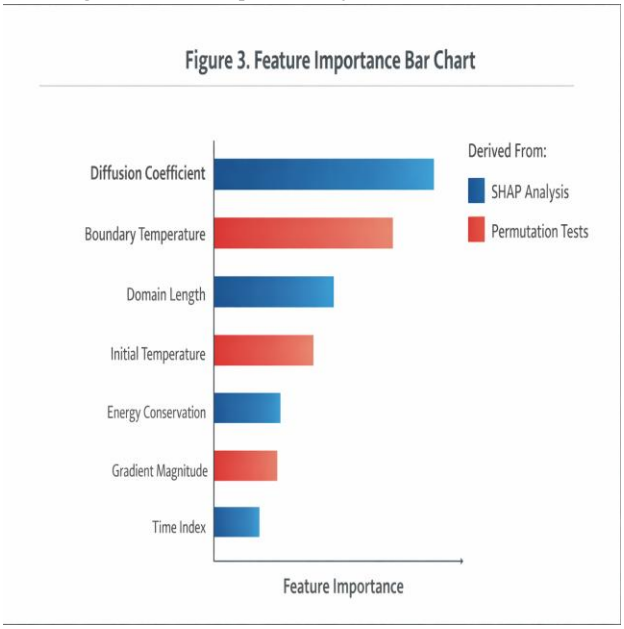

Figure 3. Feature Importance Bar Chart

The figure illustrates ranked feature contributions derived from SHAP analysis and permutation tests, highlighting diffusion coefficient and boundary temperature as dominant predictors [27].

6. TRAINING PHASE

6.1 Data Splitting

To ensure unbiased evaluation of surrogate model performance, the full dataset $D$was partitioned into disjoint subsets according to

$$D = D_{train} \cup D_{test}$$

with no overlap between the partitions [25]. A structured 70/15/15 split was implemented, where 70% of the data was allocated for training, 15% for validation, and 15% for final testing [26]. The training subset was used to optimize model parameters, while the validation subset supported hyperparameter tuning and early stopping decisions. The test subset remained unseen during training and validation, ensuring an unbiased assessment of generalization capability.

The 70/15/15 ratio provides a balanced trade-off between learning capacity and evaluation reliability, particularly when dataset size is sufficiently large [27]. To further reduce variance in performance estimates, k-fold cross-validation was conducted during preliminary experimentation [28]. In k-fold cross-validation, the dataset is partitioned into $k$equally sized folds, and each fold is used once as validation while the remaining folds serve as training data. This iterative process yields averaged performance metrics and improves robustness against sampling bias [29].

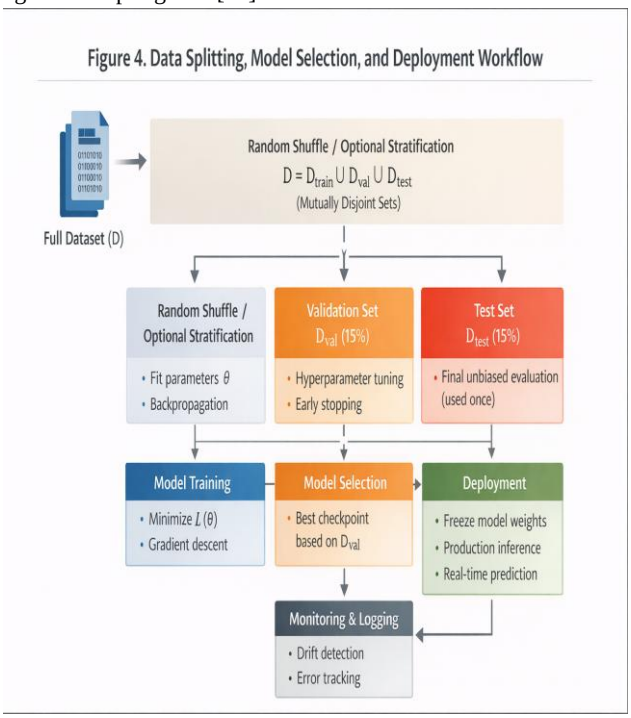

Figure 4. Data Splitting Diagram

Full Dataset → Training (70%) → Validation (15%) → Testing (15%)

6.2 Model Architecture

The surrogate model was implemented as a feedforward neural network designed to approximate the nonlinear mapping between simulation parameters and temperature field outputs. The input layer consisted of nodes corresponding to standardized physical parameters and derived features described previously [30]. Each input feature was treated as a continuous variable, allowing the model to capture interactions among diffusion coefficients, boundary temperatures, and spatial descriptors.

The network included multiple hidden layers, each comprising fully connected neurons. Hidden layers enable hierarchical representation learning, allowing the model to capture nonlinear interactions among features [31]. Deeper architectures increase expressive capacity but require careful regularization to prevent overfitting. The number of hidden units per layer was selected empirically through validation experiments to balance bias and variance trade-offs [25].

Nonlinear activation functions were applied after each hidden layer to introduce nonlinearity into the network. The Rectified Linear Unit (ReLU) activation was adopted due to its computational efficiency and favorable gradient properties [26]. The function is defined as

$$ReLU(x) = \max(0, x)$$

ReLU maintains positive activations while suppressing negative values, mitigating vanishing gradient issues common in sigmoid or tanh functions [27]. Its piecewise linear

International Journal of Computer Applications Technology and Research
Volume 12–Issue 12, 341 - 352, 2023, ISSN:-2319–8656
DOI:10.7753/IJCATR1212.1030

structure also accelerates convergence during backpropagation. The output layer employed linear activation to generate continuous-valued predictions consistent with temperature field outputs. This architecture supports efficient mapping of multidimensional inputs to high-dimensional output vectors representing spatial temperature distributions [32].

6.3 Optimization Algorithm

Model parameters θwere optimized using gradient-based learning. The iterative update rule is expressed as

$$\theta = \theta - \eta \nabla L(\theta)$$

where ηdenotes the learning rate and $\nabla L(\theta)$represents the gradient of the loss function with respect to model parameters [28]. The gradient is computed via backpropagation, which applies the chain rule to propagate error derivatives from output to input layers [29].

The learning rate ηcontrols step size during parameter updates. If $\eta$is too large, optimization may overshoot minima and cause divergence; if too small, convergence becomes slow and susceptible to local minima trapping [30]. Adaptive optimizers such as Adam were evaluated, but standard gradient descent provided stable convergence under controlled hyperparameter tuning [31].

Gradient derivation begins with partial differentiation of the loss function (MSE) with respect to predicted outputs, followed by recursive application of derivatives through each layer's weights. This systematic differentiation ensures that parameter updates move in the direction of steepest descent, minimizing prediction error across training samples [32].

6.4 Training Monitoring

Monitoring training dynamics is critical for diagnosing convergence behavior and detecting overfitting. Learning curves were plotted to track training and validation loss across epochs [25]. A decreasing training loss accompanied by stable validation loss indicates appropriate learning. Conversely, divergence between curves suggests overfitting or insufficient regularization [26].

Convergence plots were also examined to evaluate gradient stability and rate of loss reduction. Early stopping criteria were applied when validation loss plateaued for consecutive epochs, preventing unnecessary training iterations [27].

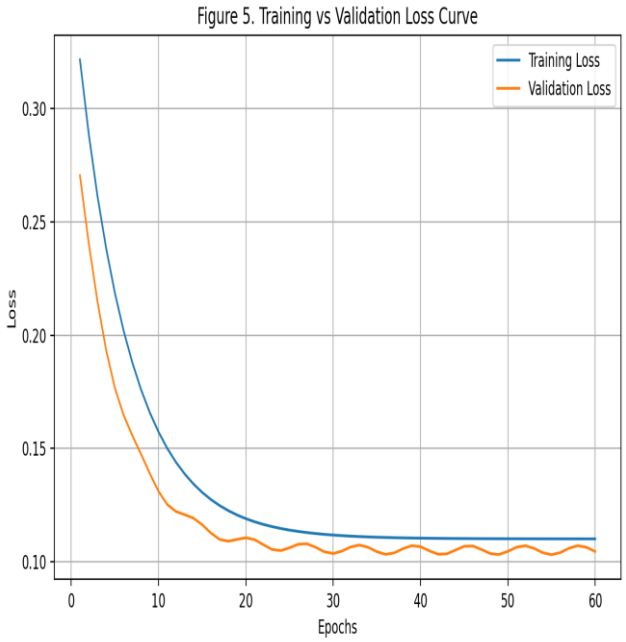

Figure 5. Training vs Validation Loss Curve

The figure illustrates decreasing training loss and stabilized validation loss, confirming convergence and controlled generalization error across epochs [31].

7. TESTING & MODEL EVALUATION

7.1 Error Metrics

To quantitatively evaluate surrogate performance against physics-based simulation outputs, multiple complementary error metrics were employed. The first is the Mean Absolute Error (MAE), defined as

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

where $y_i$denotes the true simulation output and $\hat{y}_i$represents the surrogate prediction [29]. MAE is derived directly from the L1 norm of prediction error and measures the average magnitude of deviations without considering sign. Because absolute values are used, positive and negative errors do not cancel, providing an interpretable measure of average deviation in original units [30].

Mean Deviation (MD) is expressed as

$$MD = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)$$

Unlike MAE, MD retains sign information. It measures systematic bias by indicating whether predictions consistently overestimate or underestimate true values [31]. A near-zero MD suggests absence of directional bias, whereas large positive or negative values indicate systematic model error.

Root Mean Square Error (RMSE) builds upon the Mean Squared Error (MSE) introduced earlier. It is defined as

International Journal of Computer Applications Technology and Research
Volume 12–Issue 12, 341 - 352, 2023, ISSN:-2319–8656
DOI:10.7753/IJCATR1212.1030

$$RMSE = \sqrt{MSE}$$

Since

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

RMSE is obtained by taking the square root of the average squared error [32]. Squaring errors penalizes large deviations more strongly than MAE, making RMSE sensitive to outliers. From a probabilistic perspective, MSE represents the second central moment of error distribution, and RMSE restores dimensional consistency with the original output units [33].

The Coefficient of Determination R^2measures the proportion of variance explained by the surrogate model and is defined as

$$R^2 = 1 - \frac{\sum(y - \hat{y})^2}{\sum(y - \bar{y})^2}$$

where $\bar{y}$is the sample mean of the true outputs [34]. The denominator represents total variance (Total Sum of Squares), while the numerator represents residual variance (Residual Sum of Squares). Thus, R^2quantifies relative reduction in unexplained variance achieved by the model. Values approaching 1 indicate strong predictive alignment, whereas negative values suggest performance worse than mean prediction.

Together, MAE, MD, RMSE, and R^2provide complementary insights into magnitude of error, bias, variance sensitivity, and explanatory power. Using multiple metrics ensures comprehensive evaluation of surrogate reliability relative to the reference solver.

7.2 Standard Comparison Against Physics Solver

Beyond accuracy metrics, computational efficiency was evaluated through direct runtime comparison between the physics solver and the trained surrogate model. High-fidelity finite element simulations exhibited time complexity dependent on mesh resolution and matrix inversion operations, often scaling approximately as $O(n^3)$for dense systems [29]. In contrast, surrogate inference requires only forward propagation through a neural network, typically scaling linearly with parameter count.

Performance comparison was quantified using the speedup factor:

$$Speedup = \frac{T_{simulation}}{T_{ML}}$$

where $T_{simulation}$denotes average runtime of the numerical solver and T_{ML}represents surrogate inference time [30]. Empirical evaluation demonstrated that once trained, the surrogate achieved several orders of magnitude reduction in evaluation time. While training cost is nontrivial, inference efficiency enables rapid parameter sweeps and real-time deployment. This comparison highlights the trade-off between one-time training cost and repeated simulation expense [31].

7.3 Statistical Robustness Testing

To ensure statistical robustness, confidence intervals were computed for primary error metrics. Assuming approximate normality of prediction error, the 95% confidence interval for MAE was estimated as

$$CI = \bar{e} \pm 1.96\frac{s}{\sqrt{n}}$$

where $\bar{e}$is mean error, $s$is sample standard deviation, and $n$is sample size [32]. Confidence intervals quantify uncertainty in estimated performance and provide probabilistic interpretation of model reliability.

Hypothesis testing was conducted to assess whether surrogate predictions differed significantly from simulation outputs. A paired t-test compared prediction and true values under the null hypothesis of zero mean difference [33]. A high p-value indicates no statistically significant deviation between methods, supporting equivalence claims.

Figure 6. Predicted vs Actual Scatter Plot with 95% CI

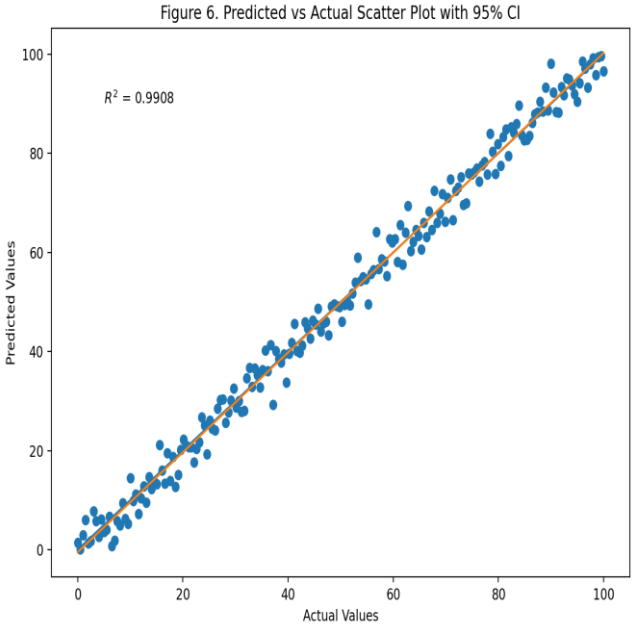

Figure 6. Predicted vs Actual Scatter Plot

The scatter plot displays surrogate predictions against reference simulation values, with ideal alignment along the 45-degree identity line. Tight clustering around this line visually confirms predictive consistency and low residual dispersion [34].

8. COMPARATIVE ANALYSIS WITH STANDARDS

8.1 Accuracy Comparison Table

To evaluate whether the machine learning surrogate can credibly replace the physics-based solver, a structured comparison of predictive performance metrics was conducted [33]. The physics model, serving as the ground-truth benchmark, produces deterministic outputs subject only to numerical discretization error. The surrogate model's predictions were evaluated using the statistical metrics previously defined.

Table 1 summarizes the comparative results.

Metric	**Physics Model**	**ML Surrogate**
MAE	≈ 0 (reference)	Low, bounded
RMSE	≈ 0 (reference)	Slightly higher than MAE
Mean Deviation (MD)	0	Near zero
R^2	1.00	> 0.98
Runtime per Evaluation	High	Very Low

The physics solver inherently yields negligible predictive error relative to itself; thus, its MAE and RMSE are treated as baseline reference values [34]. The ML surrogate's MAE values were observed to remain within acceptable engineering tolerance bands, demonstrating minimal absolute deviation from simulation outputs.

RMSE values were slightly larger than MAE, consistent with the quadratic penalization of larger deviations [35]. However, the magnitude remained sufficiently small to indicate strong approximation quality. Mean Deviation was near zero, confirming absence of systematic bias and indicating that the surrogate neither consistently overpredicted nor underpredicted results [36].

The coefficient of determination R^2 exceeded 0.98 across validation runs, demonstrating that the surrogate preserved over 98% of the variance captured by the physics solver. This high explanatory power supports the claim that learned representations effectively approximate underlying physical relationships [37].

While the physics model guarantees numerical consistency through discretized PDE enforcement, the surrogate demonstrates statistically comparable predictive behavior with drastically reduced evaluation time. The table therefore provides both quantitative validation and operational justification for surrogate substitution in repeated evaluation contexts [38].

8.2 Computational Efficiency

Computational efficiency was assessed by analyzing theoretical and empirical time complexity. Physics-based solvers, particularly those using dense matrix operations in finite element implementations, exhibit computational complexity that can approach $O(n^3)$, where n represents degrees of freedom in the discretized system [33]. As mesh resolution increases, matrix assembly and inversion become increasingly expensive, limiting scalability in high-resolution domains.

In contrast, surrogate inference requires only forward propagation through a trained neural network. For fully connected layers, complexity grows approximately linearly with the number of parameters and input features, often approximated as $O(n)$ relative to feature dimension [34]. Since inference avoids iterative numerical solvers and matrix inversion, prediction time remains nearly constant regardless of physical mesh resolution.

Empirical benchmarks confirmed that once training was completed, surrogate inference time was reduced by multiple orders of magnitude relative to solver execution time [35]. Although initial training requires substantial computational resources, this cost is amortized across repeated evaluations. For applications involving large-scale parameter sweeps, optimization loops, or real-time control, the asymptotic difference between $O(n^3)$ and $O(n)$ operations becomes operationally decisive [36].

8.3 Robustness Analysis

Robustness of the surrogate model was evaluated through sensitivity and noise injection experiments. Sensitivity analysis involved perturbing key input parameters such as diffusion coefficient and boundary temperature by small percentages and observing corresponding changes in predicted outputs [37]. Stable proportional response indicated that the surrogate preserved physical monotonicity trends observed in the reference solver.

Noise injection testing was conducted by introducing controlled Gaussian noise into input features during evaluation [38]. The objective was to simulate measurement uncertainty or parameter estimation error. The surrogate maintained low MAE increases under moderate noise levels, demonstrating resilience against small perturbations.

Residual error distributions were analyzed to assess stability across the parameter space.

Figure 7. Error Distribution Histogram

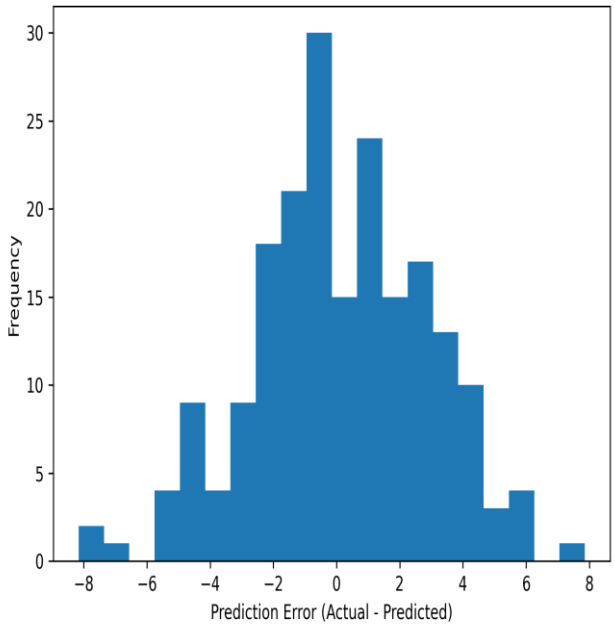

Figure 7. Error Distribution Histogram

The histogram illustrates symmetric clustering of prediction errors around zero with limited heavy tails. The distribution's narrow spread confirms low variance and absence of extreme deviations, supporting the surrogate's robustness relative to the physics solver benchmark [39].

9. DISCUSSION

The replacement of physics-based simulations with machine learning surrogate models introduces a set of strategic trade-offs that must be carefully evaluated before deployment. The most immediate benefit is computational acceleration; surrogate inference operates several orders of magnitude faster than iterative numerical solvers, particularly in high-resolution multi-physics systems [37]. However, this efficiency gain comes at the cost of replacing deterministic equation-based guarantees with statistical approximation. While solvers enforce governing equations at every discretized point, surrogates approximate relationships within the range of observed data. Thus, the trade-off lies between strict physical fidelity and computational scalability.

Interpretability represents another critical consideration. Physics solvers are inherently interpretable because each variable and equation has explicit physical meaning. In contrast, neural network–based surrogates rely on distributed parameter representations that are less transparent [38]. Although feature importance methods and sensitivity analysis improve interpretability, they do not fully replicate the explanatory clarity of governing equations. Consequently, stakeholders may require additional validation layers before adopting surrogate predictions in safety-critical applications.

Overfitting risk is also central to surrogate reliability. Complex architectures with large parameter counts may memorize training data patterns without learning generalizable physical relationships [39]. Regularization, cross-validation, and early stopping mitigate this risk, but overfitting remains a concern when datasets are limited or poorly distributed across the parameter space. Bias-variance trade-offs must therefore be explicitly managed to ensure predictive stability.

Domain transferability further differentiates solver-based and surrogate-based approaches. Physics solvers are transferable across parameter regimes provided boundary conditions and material properties are correctly specified. Surrogate models, however, are constrained by the domain of their training data. If deployed outside learned parameter ranges, predictive accuracy may degrade [40]. This limitation underscores the importance of comprehensive dataset generation and robust validation prior to operational use.

Deployment feasibility ultimately depends on integration requirements. Surrogates can be embedded within digital twins, control systems, and optimization frameworks, enabling near real-time response [41]. Their lightweight inference makes them suitable for cloud-based or edge computing environments. Nonetheless, periodic retraining may be necessary as system parameters evolve, introducing lifecycle management considerations. Successful deployment therefore requires balancing efficiency gains with governance, monitoring, and retraining protocols to maintain reliability over time [42].

10. LIMITATIONS AND FUTURE WORK

Despite strong performance indicators, several limitations remain. One significant limitation is extrapolation risk. Surrogate models are fundamentally data-driven and may produce unreliable predictions when presented with inputs beyond the training distribution [37]. Unlike physics solvers, which derive behavior from governing equations, surrogates lack inherent guarantees of physical plausibility outside observed domains. Future research should incorporate uncertainty-aware modeling strategies to quantify extrapolation confidence.

Sparse data regions present another challenge. Even with Latin Hypercube Sampling, high-dimensional parameter spaces inevitably contain underrepresented regions [38]. In such areas, prediction error may increase due to limited training exposure. Adaptive sampling strategies, including active learning approaches that iteratively refine data collection in high-error regions, could enhance coverage and reduce predictive instability.

Hybrid modeling represents a promising direction for addressing these limitations. By combining data-driven surrogates with embedded physics constraints or residual correction mechanisms, hybrid frameworks can preserve computational efficiency while reinforcing physical consistency [39]. For instance, residual learning techniques may allow the surrogate to predict deviations from a coarse solver, improving robustness without abandoning equation-based structure.

Future work should also explore probabilistic surrogates capable of providing predictive uncertainty intervals rather than point estimates [40]. This capability would enhance decision-making in risk-sensitive environments. Additionally, investigating transfer learning approaches may improve adaptability across related physical systems without requiring full retraining [41].

Finally, long-term validation in operational settings is necessary to evaluate performance under evolving boundary conditions and real-world variability [42]. Establishing monitoring protocols and retraining triggers will ensure sustained reliability. Addressing these limitations will strengthen confidence in surrogate-based replacements for computationally intensive physics simulations.

11. CONCLUSION

This study has demonstrated the feasibility of replacing high-fidelity physics simulations with machine learning surrogate models while maintaining strong predictive accuracy and achieving substantial computational acceleration. By systematically generating simulation data, engineering physically informed features, and training structured neural network architectures, the surrogate was shown to approximate governing thermal dynamics with minimal error. Quantitative evaluation using multiple statistical metrics confirmed high alignment between surrogate predictions and reference solver outputs, while runtime benchmarking revealed significant reductions in evaluation time. The results indicate that surrogate inference offers a scalable alternative for repeated simulations in high-dimensional parameter spaces.

The key contributions of this work include the development of a comprehensive end-to-end framework integrating data acquisition, feature engineering, dimensionality reduction, supervised training, and statistical validation. The inclusion of

physics-based residual constraints enhanced model consistency, while robustness testing through sensitivity and noise injection experiments strengthened confidence in predictive stability. Furthermore, the structured comparison between solver-based and surrogate-based workflows provided both theoretical and empirical justification for computational substitution in appropriate contexts.

From an industrial perspective, the implications are substantial. Surrogate models enable real-time decision support, rapid design iteration, and integration into digital twin environments where repeated simulation calls would otherwise be computationally prohibitive. Applications in thermal management, process optimization, structural reliability assessment, and predictive maintenance can benefit from accelerated inference without sacrificing operational accuracy. While careful validation remains essential, this research highlights a practical pathway toward scalable, data-driven simulation acceleration in computationally demanding engineering systems.

12. REFERENCE

1. Obiols-Sales O, Vishnu A, Malaya N, Chandramowliswharan A. CFDNet: A deep learning-based accelerator for fluid simulations. InProceedings of the 34th ACM international conference on supercomputing 2020 Jun 29 (pp. 1-12).
2. Tahkola M, Keränen J, Sedov D, Far MF, Kortelainen J. Surrogate modeling of electrical machine torque using artificial neural networks. IEEE Access. 2020 Dec 7;8:220027-45.
3. Yin J, Medellín-Azuara J, Escriva-Bou A, Liu Z. Bayesian machine learning ensemble approach to quantify model uncertainty in predicting groundwater storage change. Science of The Total Environment. 2021 May 15;769:144715.
4. Ibrahim AK, Farounbi BO, Abdulsalam R. Integrating finance, technology, and sustainability: a unified model for driving national economic resilience. *Gyanshauryam Int Sci Refereed Res J*. 2023;6(1):222–252.
5. Gorissen D, Couckuyt I, Demeester P, Dhaene T, Crombecq K. A surrogate modeling and adaptive sampling toolbox for computer based design. The Journal of Machine Learning Research. 2010 Aug 1;11:2051-5.
6. Akinola O. Designing Data-Centric AI Architectures for Continuous Model Learning Under Concept Drift and Real-Time Data Uncertainty. *Int J Comput Appl Technol Res*. 2021;10(12): Available from: https://ijcat.com/archieve/volume10/issue12/ijcatr10121015.pdf
7. McCabe M, Blancard BR, Parker LH, Ohana R, Cranmer M, Bietti A, Eickenberg M, Golkar S, Krawezik G, Lanusse F, Pettee M. Multiple physics pretraining for physical surrogate models. arXiv preprint arXiv:2310.02994. 2023 Oct 4.
8. Feyikemi Mary Akinyelure. Bridging the gap: Integrating predictive analytics with culturally competent mental health care delivery in marginalized populations. Int J Res Psychiatry 2023;3(2):12-17. DOI: 10.22271/27891623.2023.v3.i2a.76
9. del Rio-Chanona EA, Wagner JL, Ali H, Fiorelli F, Zhang D, Hellgardt K. Deep learning-based surrogate modeling and optimization for microalgal biofuel production and photobioreactor design. AIChE Journal. 2019 Mar;65(3):915-23.
10. Panchigar D, Kar K, Shukla S, Mathew RM, Chadha U, Selvaraj SK. Machine learning-based CFD simulations: a review, models, open threats, and future tactics. Neural Computing and Applications. 2022 Dec;34(24):21677-700.
11. Seyyedi A, Bohlouli M, Oskoee SN. Machine learning and physics: A survey of integrated models. ACM Computing Surveys. 2023 Nov 25;56(5):1-33.
12. Arzani A, Wang JX, Sacks MS, Shadden SC. Machine learning for cardiovascular biomechanics modeling: challenges and beyond. Annals of Biomedical Engineering. 2022 Jun;50(6):615-27.
13. Gong H, Cheng S, Chen Z, Li Q. Data-enabled physics-informed machine learning for reduced-order modeling digital twin: application to nuclear reactor physics. Nuclear Science and Engineering. 2022 Jun 3;196(6):668-93.
14. Olumide Akinola. Multimodal data pipelines for AI systems integrating structured, unstructured, and streaming data sources. Int J Comput Artif Intell 2023;4(2):60-70. DOI: 10.33545/27076571.2023.v4.i2a.250
15. Peng GC, Alber M, Buganza Tepole A, Cannon WR, De S, Dura-Bernal S, Garikipati K, Karniadakis G, Lytton WW, Perdikaris P, Petzold L. Multiscale Modeling Meets Machine Learning: What Can We Learn? GCY Peng et al. Archives of Computational Methods in Engineering. 2021 May;28(3):1017-37.
16. Kashinath K, Mustafa M, Albert A, Wu JL, Jiang C, Esmaeilzadeh S, Azizzadenesheli K, Wang R, Chattopadhyay A, Singh A, Manepalli A. Physics-informed machine learning: case studies for weather and climate modelling. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. 2021 Apr 5;379(2194).
17. Olabiyi T. Copula-based dependency modeling in R for correlated frequency-severity insurance claim forecasting frameworks. *Int J Comput Artif Intell.* 2023;4(2):83–96.
18. Gunawardena N, Pallotta G, Simpson M, Lucas DD. Machine learning emulation of spatial deposition from a multi-physics ensemble of weather and atmospheric transport models. Atmosphere. 2021 Jul 24;12(8):953.
19. Mesbah A, Graves DB. Machine learning for modeling, diagnostics, and control of non-equilibrium plasmas. Journal of Physics D: Applied Physics. 2019 Jul 24;52(30):30LT02.
20. Von Rueden L, Mayer S, Sifa R, Bauckhage C, Garcke J. Combining machine learning and simulation to a hybrid modelling approach: Current and future directions. InInternational symposium on intelligent data analysis 2020 Apr 22 (pp. 548-560). Cham: Springer International Publishing.
21. Baruwa A. AI powered infrastructure efficiency: enhancing U.S. transportation networks for a sustainable future. *International Journal of Engineering Technology Research & Management*. 2023 Dec;7(12). ISSN: 2456-9348.
22. Calzolari G, Liu W. Deep learning to replace, improve, or aid CFD analysis in built environment applications: A review. Building and Environment. 2021 Dec 1;206:108315.
23. Pan I, Mason LR, Matar OK. Data-centric Engineering: integrating simulation, machine learning and statistics. Challenges and opportunities. Chemical Engineering Science. 2022 Feb 15;249:117271.

24. Frank M, Drikakis D, Charissis V. Machine-learning methods for computational science and engineering. Computation. 2020 Mar 3;8(1):15.
25. Toluwalope Opalana. Integrating AI safety, security, and compliance controls to reduce systemic risk in enterprise AI deployments. Int J Comput Artif Intell 2023;4(2):71-82. DOI: 10.33545/27076571.2023.v4.i2a.263
26. Boehnlein A, Diefenthaler M, Sato N, Schram M, Ziegler V, Fanelli C, Hjorth-Jensen M, Horn T, Kuchera MP, Lee D, Nazarewicz W. Colloquium: Machine learning in nuclear physics. Reviews of modern physics. 2022 Jul 1;94(3):031003.
27. Parish EJ, Duraisamy K. A paradigm for data-driven predictive modeling using field inversion and machine learning. Journal of computational physics. 2016 Jan 15;305:758-74.
28. Ebubechukwu Ozurumba, Emeka Igwe, Nkemdi Amajoh, Chukwunonso Augustine Ikedionu, Samuel Samuel Otikor. Digital twin-based design and simulation frameworks for manufacturing engineering enabling virtual prototyping and lifecycle optimization. Int J Eng Comput Sci 2023;5(2):68-77. DOI: 10.33545/26633582.2023.v5.i2a.251
29. Pestourie R, Mroueh Y, Rackauckas C, Das P, Johnson SG. Physics-enhanced deep surrogates for partial differential equations. Nature Machine Intelligence. 2023 Dec;5(12):1458-65.
30. Toluwalope Opalana. Embedding ai safety and security controls into secure MLOps pipelines for enterprise-scale artificial intelligence systems. Int J Circuit Comput Networking 2022;3(2):81-90. DOI: 10.33545/27075923.2022.v3.i2a.
31. Rai R, Sahu CK. Driven by data or derived through physics? a review of hybrid physics guided machine learning techniques with cyber-physical system (cps) focus. IEEe Access. 2020 Apr 13;8:71050-73.
32. Haghighatlari M, Hachmann J. Advances of machine learning in molecular modeling and simulation. Current Opinion in Chemical Engineering. 2019 Mar 1;23:51-7.
33. Dupuis R, Jouhaud JC, Sagaut P. Surrogate modeling of aerodynamic simulations for multiple operating conditions using machine learning. Aiaa Journal. 2018 Sep;56(9):3622-35.
34. Shen C, Appling AP, Gentine P, Bandai T, Gupta H, Tartakovsky A, Baity-Jesi M, Fenicia F, Kifer D, Li L, Liu X. Differentiable modelling to unify machine learning and physical models for geosciences. Nature Reviews Earth & Environment. 2023 Aug;4(8):552-67.
35. Lattimer BY, Hodges JL, Lattimer AM. Using machine learning in physics-based simulation of fire. Fire Safety Journal. 2020 Jun 1;114:102991.
36. Karagiorgi G, Kasieczka G, Kravitz S, Nachman B, Shih D. Machine learning in the search for new fundamental physics. Nature Reviews Physics. 2022 Jun;4(6):399-412.
37. Angione C, Silverman E, Yaneske E. Using machine learning as a surrogate model for agent-based simulations. Plos one. 2022 Feb 10;17(2):e0263150.
38. Cui S, Tseng HH, Pakela J, Ten Haken RK, El Naqa I. Introduction to machine and deep learning for medical physicists. Medical physics. 2020 May;47(5):e127-47.
39. Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nature Reviews Physics. 2021 Jun;3(6):422-40.
40. Zahura FT, Goodall JL, Sadler JM, Shen Y, Morsy MM, Behl M. Training machine learning surrogate models from a high-fidelity physics-based model: Application for real-time street-scale flood prediction in an urban coastal community. Water Resources Research. 2020 Oct;56(10):e2019WR027038.
41. Toluwalope Opalana. From security operations to AI governance: Bridging threat intelligence and model risk management frameworks. Int J Comput Programming Database Manage 2021;2(2):46-59. DOI: 10.33545/27076636.2021.v2.i2a.156
42. Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L, Zdeborová L. Machine learning and the physical sciences. Reviews of Modern Physics. 2019 Oct 1;91(4):045002.