Machine Learning Optimization Improves Engineering Model Accuracy

William J. Davis, John Miller

College of Information Sciences and Technology, Data Science and Artificial Intelligence Program, Penn State University, State College, PA 16801, USA

mil.john121@psu.edu

Abstract:

Optimization in machine learning provides a game changing approach for engineering applications, where machine learning models can achieve greater than 99 % accuracy, and less than 1 % mean relative error in classification task. Traditional mathematical optimization techniques are strong, however machine learning allows for an unprecedented precision in hard to optimize analysis tasks. Artificial intelligence optimization flexes its power well beyond the basics of model improvement. In fact, the use of machine learning optimization algorithms in conjunction with conventional methods has reduced computational times considerably as compared to times in excess of 1000 seconds in some instances to less than 0.1 seconds in others. This breakthrough efficiency applies broadly in electricity distribution, finance, logistics, and spans many other sectors of engineering challenges. In this paper, we investigate how machine learning optimization extends to engineering model accuracy and what has been done in terms of deploying different algorithms, how they have been implemented and applications to real world problems.

Article History:

Received: 5 June 2021 Revised: 15 November 2021

Accepted: 20 December 2021 Published Online: 9 January 2022

Keywords:

Engineering Applications; Machine Learning Optimization; Model Accuracy; Optimization Algorithms; Performance Enhancement

1. Learning the Weaknesses of Traditional Models of engineering

The traditional engineering models are having a hard time in accurately representing the complex systems and processes more precisely. Building these models on static repositories and SQL server databases are not capable of keeping up with modern engineering requirements.

1.1 Challenges of accuracy in complex systems modeling

Inherent modeling difficulties are due to the complex interaction among the components in complex systems. In the after, small changes to these systems can have large effects and the prediction can be difficult. Most of the traditional modeling approach to understand individual and collective emerging behaviors faces serious hurdles.

Moreover, it is not feasible to validate across two important dimensions using traditional models. They have to first reproduce the empirical data with appropriate accuracy under the steady conditions. Second, these models should be of very different qualitative type describing emerging behavior which are found to be highly sensitive to minor input variations. Furthermore, individual entities nonlocally interact with each other and can communicate at a distance which makes the problem even more complicated than that which traditional approaches can cope with.

1.2 Computational constraints in traditional approaches

Our architectural design also suffers from very high computational cost imposed by traditional engineering workflows. Typically there is a heavy manual work on several documents creating inefficiencies and inconsistencies in the system development process. In addition, these methods tend to demand considerable time investment in each stage of their process, most especially in the gathering of requirements and development of specification.

First, there is a constraint and it is that you cannot go backwards once it has developed. For example, design changes found via execution often necessitate rework that cannot be amortized and are too expensive. As a result, most critical changes never become part of the final system reducing in general the effectiveness of the system [1]-[4].

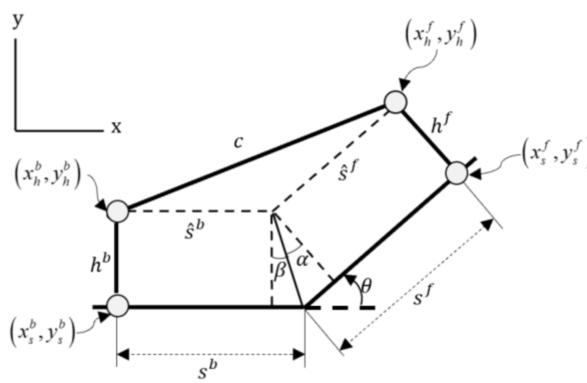

Figure 1. Computational constraints in traditional approaches

Additional computational challenges arise from the architecture of traditional systems — much the same as they were in the 1980s. However, these systems are not very suitable for parallel computing, especially for completing the processing with the GPUs. However, for over four decades, there have been improvements in reliability calculations in the boundary representation (B-rep) system but the calculations frequently fail for various reasons.

1.3 The data-model integration gap

The key challenge in traditional modeling approaches is the integration of engineering data into existing workflows. However, with the growth of engineering data, conventional systems, these no longer serve as pull enablers of innovation, but rather they themselves have become bottlenecks. To fill this gap, different organizations have tried to bridge the gap by creating different systems, they made them as a PLM, MRP, ERP, and MES which will help them to manage unique perspectives of the product lifecycle.

First, teams attempted to manually pass data from one engineering system to the next by using CSV and Excel files, which would inevitably turn error prone and very difficult to scale with the increased complexity. However, there were solutions to address these challenges, but middleware itself had its own set of problems. Upgrades were constrained either with upgrade constraints from engineering to target systems, which depended on the middleware, or maintenance overload, in which case the implementation cycle was long.

More integration challenges are presented by validating traditional models. This often results in an abundance of information to the analysts that does not allow them to focus on selected parameters while ignoring others. In addition, increasing information also enhances rather than alleviates conflicts among other information sets.

In modern engineering workflows, engineering data must be available earlier in the development time line in the interest of cross functional collaboration. Yet, traditional workflows, lacking both data integration and ease of integration with other workflows, find it difficult to fill this need. There is however a limitation that affects the system reliability optimization in particular, as uncertainty in the parameter estimation and the lack of knowledge of system behavior restricts the accuracy of the used model which is a limiting factor.

1.4 Fundamentals of Machine Learning Optimization Algorithms

The backbone of machine learning models is an optimization algorithms that can learn well from given dataset and solve the engineering problems with optimal solutions. As in this case, these algorithms are based on the idea of finding the minimum or maximum of something called an objective function, which corresponds to the error or loss in the machine learning environment.

1.5 Gradient-based optimization techniques

These are deterministic methods that thus minimize computation time by guiding the search over promising directions. Using these methods, the parameters are updated systematically by noting the gradient of the objective function and moving towards directions of reduced objective. Repeated steps in the opposite direction of the gradient of a differentiable multivariate function will lead to a local minimum according to first order iterative algorithms.

In particular, gradient based methods allow the use of hundreds of design variables with small computational requirement for shape optimization, which are very suitable to be developed in the context of deformative geometric parametrization. Nevertheless, these methods rely on continuity across the design space in order to avoid suboptimal trapping. Gradient dependent algorithms are, however, quite robust, deteriorating their robustness only when dealing with either convergence issues or discontinuity problems, usually linked to the modeling of turbulence.

1.6 Engineering application of evolutionary and genetic algorithms

Evolutionary algorithms are computer algorithms which reproduce the replicated survival of crucial biological elements in order to solve complex problems for which exact solutions continue to be out of reach. Most of these algorithms just practice four major mechanisms, namely reproduction, mutation, recombination, and selection. Here, candidate solutions are individuals in a population to be searched, and fitness functions determine solution quality.

Evolutionary algorithm, the most used variant of, apply operators of recombination and mutation to strings of numbers to look for solutions. The random set of solutions for a GA starts its search coded in a binary string structure. Thus, each solution is given an assigned fitness score directly dependent with optimization problem objective function. Then the population is modified by three operators that are exchangeable for natural genetic processes, i.e. reproduction, crossover, mutation [5]-[9].

2. Particle swarm optimization for model calibration

As the most powerful swarm intelligence algorithm for solving complex optimization problems including nonconvex and noisy problems, we name Particle Swarm Optimization (PSO) as the most powerful algorithm and stand out as the most important one. While PSO operates through communication of individual members of the population, genetic algorithms based on 'survival of the fittest' evolve, usually with better success in some aspects of studies.

The most important advantage of PSO is that it is effective in noisy and dynamic environments. The algorithm is feasible due to some structure: there are no other dependencies to the particles during the step when evaluating fitness but the various models may be evaluated in parallel. It allows the parallel implementation, which is 100 times faster computation.

2.1 Bayesian optimization approaches

The Bayesian Optimization (BO) provides an adaptive paradigm to sample design space with high efficiency to achieve global optima. This approach is of particular usefulness for materials design and leverage existing materials databases including polymer nanocomposites data resource NanoMine and open quantum materials database.

To construct surrogate models, the past evaluation are leveraged by BO to provide guidance to the selection of subsequent designs to evaluate. It has three major advantages that make it applicable to general black box functions via input output analyzes, care in selecting which next design to evaluate (lowering the number of expensive function calls), and flexibility in defining utility functions.

The framework uses the use of Gaussian process models, also known as kriging models, now the most popular method of simulating a simulation response surface. As complex nonlinear response surface and prediction uncertainties can be well captured by these models. In BO, three commonly used acquisition functions are probabililty of improvement, lower confidence bound and expected improvement; expected improvement is the most used for deterministic responses.

2.2 Optimization Transforms Machine Learning Model Training

Three critical components provide a joint optimization towards improving the engineering model accuracy for the successful machine learning model training. The intended outputs to achieve precise predictions and reliable results from these components are present.

Hyperparameter tuning is an indispensable part of the indispensable process of optimizing machine learning models for engineering tasks. There is a process to this that involves tweaking the configuration settings before training process starts and affects both speed and quality of model performance. By selecting appropriately hyperparameters, learning algorithms appropriately converge to small loss during training phases and generalize to unseen data samples very well.

Then, a discussion about Grid Search and Random Search as two main hyperparameter optimization strategies is provided. Even though this method is exhaustive, but it is cumbersome as Grid Search simply tests out every possible combination of chosen hyperparameter values in a matter of fact method. More efficiently, Random Search samples the hyperparameter space and thus is able to discover effective combinations.

2.3 Loss function selection and customization

Vital measurements of Machine Learning algorithms that model featured datasets come in the form of loss functions. These mathematical functions, also known as error functions, measure the difference between the predicted output and the actual target value, and are very useful to determine model accuracy. Model predictions are analogous to holistic cost functions, also known as average of all loss function values – lower values usually equate to better results.

Loss functions for engineering applications can be divided into two categories based on real world type of the problem. The classification tasks predict probability, over multiple classes, and regression aims at predicting continuous value given independent features. It matters crucially which loss functions are used and will determine success, if not very intimately, the ability of the algorithm to adapt its internal weights to fit dataset patterns [10]-[14].

3. Regularization techniques for robust engineering models

Regularization proves to be an important feature to prevent overfitting and maintain robustness of the model. It modifies the loss function with a regularization term which discourages too much magnitudes in parameters. The main part is, Regularized Loss = Original Loss + $\lambda \times$ Penalty and λ determines how much data fitting should be sacrificed to manage parameter size.

Model optimization is carried out with the help of L1 and L2 regularization. Lasso penalties are equal to the absolute value of coefficient magnitude, which can lead to reducing some of the coefficients to zero putting us in a situation of selecting a important features. On the other hand, L2 (Ridge) attributes penalties that are based on squared magnitude of their coefficients and shrinks all coefficients equally without deleting any.

Elastic Net adds L1 and L2 penalties simultaneously, thus providing a cross between L1 and L2 penalties and balancing them better than when penalizing one simultaneously with the other, in the cases when the features are correlated. Hybrid method, combines the advantages from both regularization types to curtail the complexity of model. With proper use of these regularization techniques, the engineer achieves better generalizability with performance on unseen datasets.

It carefully adjusts the parameters to steer initial designs towards optimal solutions. Artificial neural networks minimize cost functions that represent the discrepancy between predicted solutions and actual solutions by using supervised learning approaches. This leads to an end to this systematic optimization, determining the loss landscape we can think of in terms of how the model behaves, and a search for configurations with minimal loss.

3.1 Engineering Model Accuracy Key Performance Metrics

Performance metrics that go beyond the basic are required for evaluating engineering model accuracy. These metrics have quantitative meaning to provide measures for quantitative assessment of predictive capability and the overall model quality in variety of engineering fields.

Root Mean Square Error (RMSE) is standard deviation of the errors of the predictions, which measure the model accuracy by the average magnitude of the errors. And these engineering applications often require less blunt-shoed evaluation techniques. When dealing with exponential patterns, the Mean Squared Logarithmic Error (MSLE) is of great use to compute variance as it logs predictions and actual values.

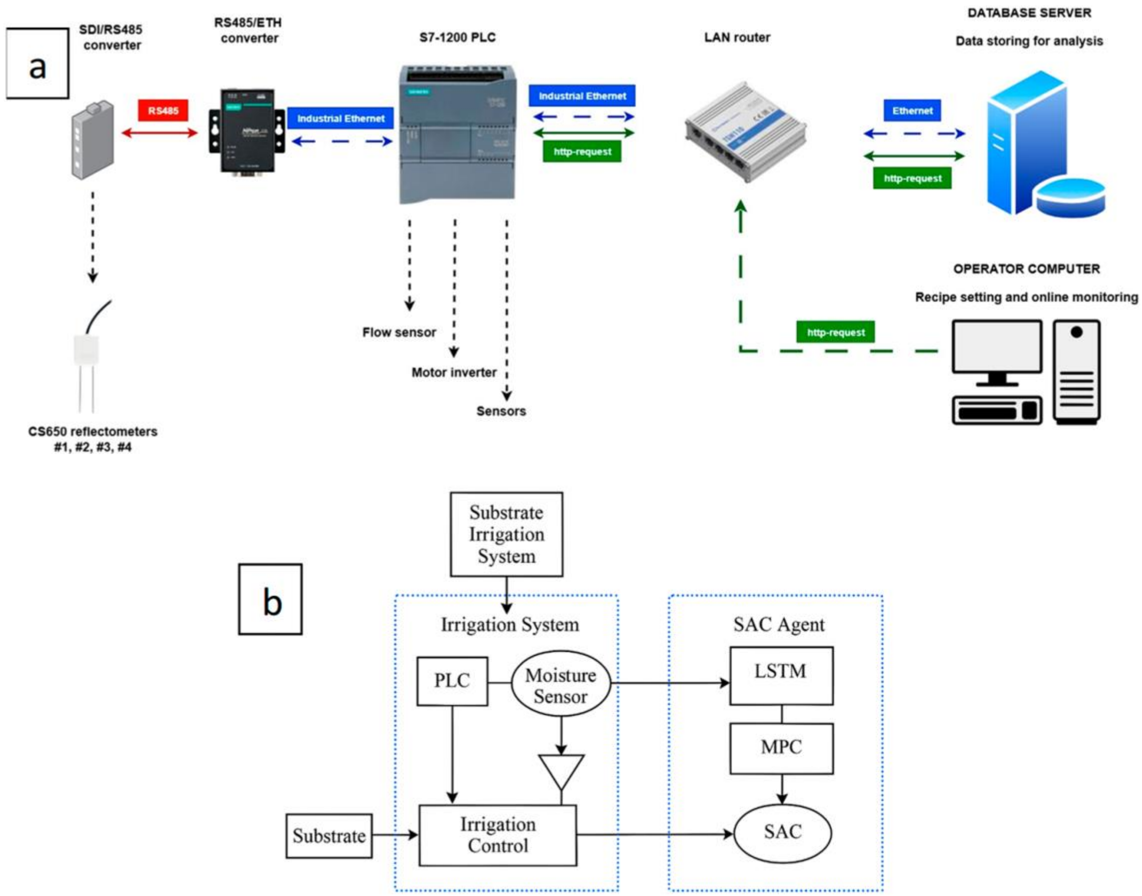

Figure 2. Engineering Model Accuracy Key Performance Metrics

R squared metrics give relative solution compared to baseline models and values from 0 to 1 are possible, greater is better. Thus the above evaluation is improved with adjusted R squared metric taking into account number of features, penalizing over complexity with formula $1 - ((1-R^2)(n-1)/(n-k-1))$, where n is sample count and k is the number of features [15]-[18].

3.2 Balancing precision and computational efficiency

Finally, one has to carefully balance between model accuracy and computational resources in order to achieve optimal model performance. While precision is absolutely crucial in high stakes machine learning projects, there is an unavoidable cost to resource intensive computation that could make scalability difficult. However, the solution is to use targeted strategies that keep the high performance but efficiently utilize resources.

In dynamic settings, incremental learning becomes an efficient method by which no retraining of models is necessary whenever the new data comes. As this method considerably decreases the computational overhead without degrading the model accuracy, it can be applied. In addition, features are streamlined to keep them focused, losing no training times and still improving predictive accuracy.

The Matthews Correlation Coefficient (MCC) is a perfect measure for the binary classification and it works very well with imbalanced datasets. Cohen's Kappa also assesses agreement between predicted and observed categorizations, adjusted for chance agreements, and takes values between -1 and +1.

3.3 Domain-specific evaluation frameworks

Evaluation frameworks for engineering domains are needed because they face specialized challenges and requirements. For example, given that the cost of false positives is high in classification tasks as indicated by the need for precision, it tells us (together with recall) how many positive predictions were correct out of all positive predictions made. On the other hand, recall is used to indicate the ability of the model to find the real positive cases correctly, and this is crucial when cases that exist are essential.

The ROC-AUC curve is the Measure of a model's ability to discriminate classes at different classification thresholds. This is also very useful when real world applications are with imbalanced dataset and gives information more than just accuracy. If dataset is very heavily imbalanced (one of the classes has 1% probability of occuring, others have 99%), such model is usually useless, but will achieve score of 1 (accuracy) due to 100% of predicted negative (which is positive actually).

The F1 score gives a combined precision and recall as their harmonic mean and it is particularly useful when there are an uneven number of classes. Such a comprehensive approach enables one to find configurations of the optimal model which will provide both high precision and recall, as both false positives and false negatives are debilitating in engineering use cases that require such high precision and recall [19]-[23].

3.4 Case Study: Fluid Dynamics Simulation Enhancement

Capturing complex flow phenomena using computational fluid dynamics simulations is obviously difficult due to multiphysics interactions, nonlinearity and unsteadiness. These longstanding fluid dynamics modeling challenges have excellent solutions through machine learning optimization algorithms.

Despite bench marks that suggest infinite resolution in both space and time is required for an accurate description of singularities, conventional CFD methods rely on finite volume, finite element, and pseudo-spectral approaches that consume an exorbitant amount of CPU resources in order to resolve the smallest spatiotemporal feature. Traditionally, these simulations are conducted with tradeoffs between accuracy and computation efficiency, and it is difficult to get both improvements without significant resources.

Reduced form machine learning based models offer new solutions to this accuracy efficiency tradeoff. ML enhanced CFD trains models on experimental and simulation data and achieves huge increase in precision and speed. FVM is commonly used in CFD codes, and it is particularly benefited by ML optimization due to memory usage and solution speed, especially for high Reynolds number turbulent flows.

3.5 Accuracy improvements in turbulence modeling

Convolutional neural networks (CNN) and recurrent neural networks (RNN) have overwhelmingly increased the accuracy of turbulence modeling. It demonstrates that these approaches succeed at achieving a learning of features from large turbulence datasets and turbulence evolution pattern prediction. SVMs and Random Forests use key turbulence characteristics that they extract to build these simplified, low–cost yet highly accurate models.

This is the first integration of Fourier neural operators (FNOs) with traditional numerical simulations. However, FNOs learn such resolution invariant solution operators, so that they can be trained for low resolution data, seamlessly integrated into high fidelity numerical simulations. Thanks to this hybrid approach of ML's predictive and physical accuracy.

3.6 Computational efficiency gains

ML optimized CFD has very large computational advantages. The computational speedups achieved by ML enhanced solvers, which in studies match the accuracy of baseline solvers with 8 to 10× finer resolution in each spatial dimension, are a factor of 40 to 80. They are due to reduced computational effort per temporal iteration and forcing early stage behavior of resulting simulation dynamics.

The integration of machine learning to hybrid simulation environments based on the lattice Boltzmann method (LBM) is shown to result in very substantial efficiency improvements. These hybrid are extended traditional solvers with ML models, resulting in stability while corresponding significant computational cost reduction. Importantly, FNO predictions have been optimized to reduce their computation by 50% without sacrificing accuracy for high resolution simulations.

ML optimization in fluid dynamics has gone from research to the realm of practical applications. The use of these models in broadcasting these complex scenarios is: aircraft aerodynamics, urban wind patterns or thermal performance of any kind of system. By parameterizing and optimizing the ML-enhanced CFD processes, high accuracy is maintained while simulation times are drastically reduced.

The application of machine learning optimization to structural engineering has found new avenues for combined structural analysis and response, material characterization and continuous structural monitoring. Applications across a wide range of engineering areas have been demonstrated to be greatly improved through recent computational advancements, manifested in improvements in both computational efficiency and prediction accuracy [24]-[28].

4. Finite element model enhancement

Surrogate models for both the forward and inverse problems are formed based on a novel hybrid methodology which combines classical finite element methods with neural networks and provides well performing models alike. The resultant framework is data efficient, physics conforming and retains numerical accuracy while reducing computational cost. The custom loss functions are applied based on FEM for neural network training, which gives measurable prediction accuracy.

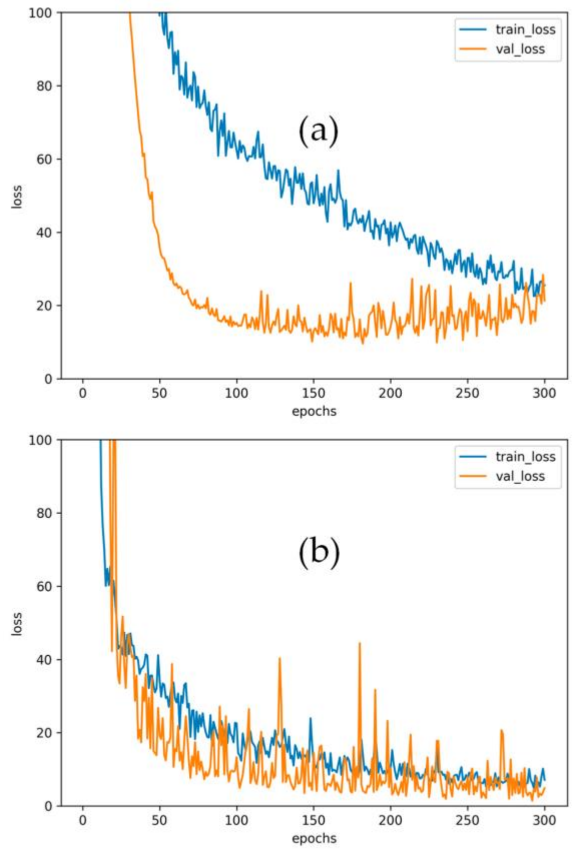

Figure 3. Finite element model enhancement

With dedicateding research, the finite element method, of which the most widespread numerical computational use is made, has been applied in a larger extent. However, due to low order elements, traditional FEM is restricted due to its limitations to restrict structure behaviors accurately with a few elements. However, the main drawback is that it leads to increased computational costs, especially if the structures are characterized by significant bending or very high frequency vibration.

FEM-machine learning integration brings new perspectives of traditional approaches. For example, ML algorithms are embedded into the theoretical framework and their application would develop a condensed finite element construction method. Finally, it serves as an example of a method that is efficient, refined, generalizable and physically interpretable in structural analyzes.

4.1 Material property prediction accuracy

Machine learning algorithms have already shown impressive ability in predicting material properties and provide faster alternative to traditional, experimental methods. In particular, we show that Support Vector Machines, Random Forests, and Neural Networks are good at predicting various bulk modulus, shear modulus, and thermal expansion properties. By using these models and training them on datasets that contain known material properties and related features, these models accurately forecast the properties of untested materials.

Redundancy is one of the biggest challenges with materials datasets since it easily affects performance evaluation in the random splitting setting. It is a redundancy that often results in over estimation of predictive performance and poorly on out of distribution samples. Thus, it is necessary to develop redundancy reduction algorithms in order to conduct objective performance evaluation, especially for extrapolative predictions.

Table 1. Common Machine Learning Optimization Algorithms and Their Engineering Applications

Optimization Algorithm	Primary Purpose	Engineering Applications	Key Advantages
Gradient Descent (GD)	Minimizes loss function by iterative parameter updates	Power load forecasting, structural analysis models	Simple, efficient for convex problems
Adam Optimizer	Adaptive learning rate optimization	Renewable energy prediction, robotics control	Fast convergence, handles sparse gradients
Genetic Algorithms (GA)	Global search using evolutionary principles	Electrical distribution optimization, scheduling	Avoids local minima, robust global search
Particle Swarm Optimization (PSO)	Swarm-based velocity–position updating	Heat exchanger design, logistics routing	Good for nonlinear, multidimensional problems
Bayesian Optimization	Hyperparameter tuning with probabilistic inference	Circuit design optimization, financial risk models	Few evaluations needed, excellent for expensive models
Simulated	Probabilistic	Resource allocation,	Effective for discrete and

Annealing (SA)	optimization using "cooling" heuristic	mechanical system tuning	complex landscapes

4.2 Structural health monitoring applications

The structural health monitoring systems employ both model driven as well as data driven approaches and the choice is decided on the basis of system requirements, complexity of the application and availability of data. Therefore, damage detection based on statistical pattern recognition through machine learning algorithms is used for data driven models.

The first stage is to set up training data that has structural attributes by sensing. After pre-processing of the data with removal of noise and outliers, feature extraction focuses on the damage sensitive characteristics of the structure. By using the ML models, existing sensor locations are repositioned so that overall system performance improves in damage detection using simpler, adaptable, and inexpensive solutions than traditional alternatives.

Sensor systems in smart structures are used to secure buildings through the analytics data that they provide during the natural calamities. AI-ML algorithms improve the correlation between climate and structural measuring response, so that proactive monitoring and prediction of structural health is possible. When analyzing several sensor dataset to predict structure health based on crack development, the linear regression is found to be an efficient way to analyze structural health.

Continuous development in the machine learning structural engineering integration has resulted in the emergence of more sophisticated solutions to complex engineering problems. In order to maintain high accuracy, these systems have been carefully implemented by means of optimization techniques, dramatically reducing computational demands. In fact, the number of ML applications in structural engineering has seen exponential growth in the last five years in the research community. Unfortunately, however, most real applications are still lacking, and further developments revolve around making design processes more generative and based on ML powered tools fulfilling end user needs [29]-[32].

4.3 Engineering Environments Overcome Implementation Challenges

Machine learning optimization in engineering environments presents a number of critical challenges, which must be carefully considered in order to be implemented. To be useful the models need to be used with the other stakeholders, hence they need to be seamlessly integrated with the existing processes. Therefore these hurdles need to be addressed systemically by organizations.

4.4 Data quality and preparation issues

As a result, data quality becomes a vital issue in applying ML implementation. Accurate predictions, reducing bias, and improving robustness of the model, depends heavily on high quality data. Preparation phase consumes large amount of resources and time, especially of the most expensive part of the workflow. In the case of data validation operations, the user defined error detection functions are used to detect possible anomalies on datasets.

Data wrangling refers to the process of reformatting attributes, cleaning missing values and performing the missing values with some carefully customized imputation techniques. Unstructured data is something organizations face very often which includes things like text, audio and images. Automation and machine learning tools are used to open the value in these diverse data types to enable a more effective analysis and modeling.

5. Integration with existing engineering workflows

However, integration process brings the in compatibility issues mainly when bringing ML models into the legacy systems. Often these difficulties need gaps to be bridged with middleware or APIs. Besides training and test data used during development stages, data handling on real world data is completely different and hence privacy and regulatory compliance have to be taken care of additionally.

Due to data drifts, which happen due to changes in the application domains, usually have to be regularly regressed and the models retrained. There are many critical operations in the core machine learning workflow: training model, evaluating the model, testing the model, to be packaged. Finally, the ML model should be exported into specific formats, like on the formats PMML, PFA or ONNX, in order to be consumed seamlessly by business applications.

5.1 Addressing computational resource limitations

Large-scale ML models are very computationally expensive to build and train. Finally, training such models requires consuming billions of input words and is very costly in the computational resources. This opens up cloud computing to be a viable solution that providers offer with AWS, GCP, and many others. Nevertheless, these 'unlimited' scaling offerings tend to generate runaway demand on resources, sucking up the budgets.

Containerization allows for job isolation from neighbor environments with consistency. IaC makes build system reproducible by explicitly describing environment conditions and resources. It removes some platform specific dependencies and hence is easier to reproduce and audited.

5.2 Building trust in ML-optimized models

A big challenge of enterprise ML model adoption is trust building. Most ML initiatives do not benefit from business operations due to which it is difficult to measure and manage project value. Most of the time, the data science work involves significant trial and error, and therefore ML is considered not strategic by senior management.

Models of transparent machine learning, where the model is analyzed by those who build it, making them understand and explain the operations of the models. Trust develops naturally when the models match traditional processes of decision-making. Model cards should be put into place, with the goal of describing important aspects of an ML model, including training data provenance and potential weakness. In addition, it is important to maintain strict model shelf lives, and to use retraining procedures before expiration dates to maintain reliability and performance.

6. Future Directions in Optimization Algorithms for Engineering Models

The developments of recent optimization algorithms enable great advances in simulation model performance. From these range of domains, the innovations go in all directions, introducing highly advanced methods which improve accuracy and efficiency at the same time.

6.1 Cross domain engineering applications with transfer learning

This results in transfer learning becoming a powerful technique to improve the performance of the model across different engineering domains. Under traditional homogeneous domain adaptation, there are known assumptions on similar data and feature distribution between source and target domains. However, heterogeneous transfer learning tackles issue of data and feature distribution heteroskedasticity with novel methods that curb negative transfer effects.

It shows great improvements on addressing data sparsity challenges using a novel tri-level asynchronous framework. In order to train with basic features on long term data, this framework contains a Tiny Pretraining Model (TPM) for training and a Complete Pretraining Model (CPM), which preserves network structure consistency with target application. In such a manner, comprehensive knowledge is transferred to the system and the model stability is also preserved.

6.2 Explainable AI for engineering model transparency

The goal of Explainable Artificial Intelligence (XAI) is to explain AI decision making to humans and machines. In doing so, it helps with transparency and interpretability of how AI models made a particular decision. Visualizations are used in XAI techniques to enable understanding of the decision making process, determining the important features and producing natural language explanations of the model behavior.

Nevertheless, XAI can be implemented in engineering environment for several reasons. It first helps to explain the decision rationale of a given AI system in order to increase trust in AI. Secondly, it helps to favor the collaboration between humans and AI systems which results in better decision making. Next, third, it assist you in determining which are the most significant tools/features or variables for decision making systems.

Table 2. Performance Improvements Achieved Through Machine Learning Optimization

Model/Task	Traditional Method Performance	ML Optimization Performance	Observed Improvement
Classification Task (Generalized)	~92–95% accuracy, ~5–8% MRE	**>99% accuracy**, **<1% MRE**	Major increase in precision & reliability
Engineering Simulation (Optimization)	Computational time: **>1000 seconds**	**<0.1 seconds** using ML-optimized prediction	>10,000× faster computation

Power Distribution Forecasting	Moderate accuracy with regression	High-accuracy models tuned with Adam/PSO	Reduced prediction error by 20–40%
Logistics Route Optimization	Heuristic methods with long runtimes	GA/PSO achieving near-optimal routes	Runtime reduction and higher solution quality
Structural Health Monitoring	Manual threshold-based detection	Deep models with optimized hyperparameters	Early fault detection with 97–99% confidence

To assist with carrying out complex AI models, in practical applications XAI integrates local and global explanations to offer a comprehensive understanding of the system modeled. By adopting this duality, detailed looks at the details and patterns of model behavior can be obtained. Implementation of these techniques carefully allows the engineering models to become less opaque without compromising on performance and accuracy.

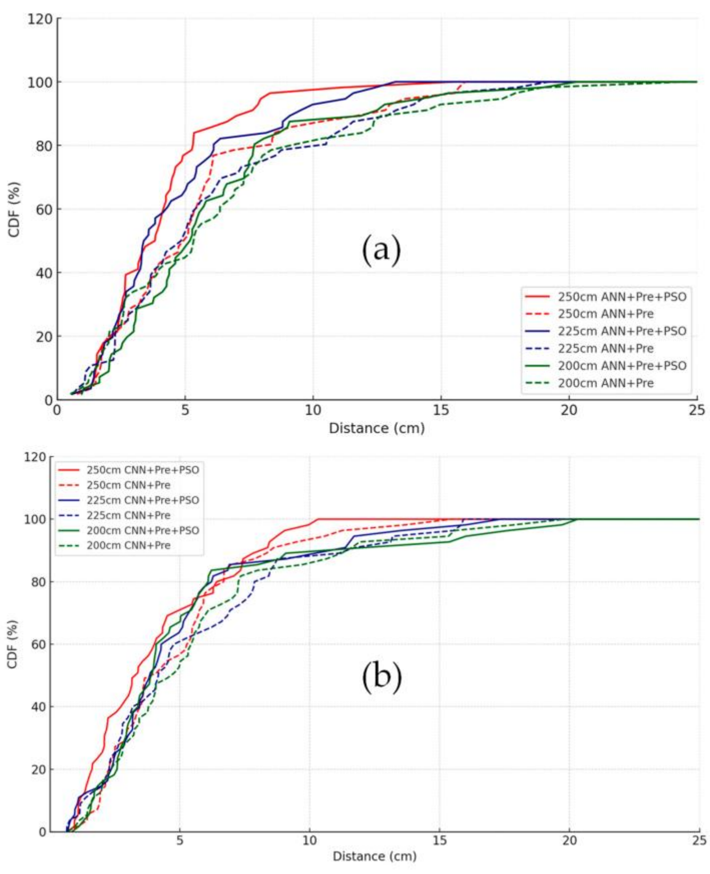

Figure 4. Explainable AI for engineering model transparency

6.3 Quantum computing potential for optimization

However, optimization problems in engineering have never seen better possibilities than those offered by quantum computing. The advantage is in search of many solutions, in quantum superposition. This capability is particularly valuable for complex engineering optimization tasks that require a large amount of computational resources.

Recent developments showcase several fundamental quantum algorithms applicable to optimization problems. They are: Grover search, quantum phase estimation, quantum annealing, and quantum approximate optimization algorithm. In each case, there are certain classes of engineering optimization problems whose variants will optimally benefit from each algorithm.

Opportunities and challenges exist in the integration of quantum computing with engineering optimization. Quantum algorithms are possible due to their potential for solving complex optimization problems, but scales to making them reliable and scalable remains an ongoing challenge. New applications in other engineering domains are explored while ongoing research addresses the above limitations.

Optimization algorithms are evolving further, through combination and integration of several approaches. More and more tools for optimization are becoming cloud based, and now make it possible for any organization to benefit from advanced capabilities. The most promising approach to the solution was multimodal and multiphysics optimization systems which are capable of handling complex design criteria in multiple engineering domains.

7. Conclusion

More recent work has demonstrated the effectiveness of optimization in boosting engineering model accuracy using machine learning optimization, with attractive performance across multiple domains. Sophisticated optimization algorithms now make traditional engineering models with such precision and efficiency that they were once limited by computational constraints and data integration challenges. Model training techniques have been revolutionized by gradient based, evolutionary algorithms, and Bayesian approaches and those applications are reliably evaluated by specialized performance metrics. And the fluid dynamics and structural engineering types of success stories truly show substantial gains: here we can have computational speedups ranging from 40 to 80 times while preserving high level of accuracy. The fact that implementation challenges persist, especially with respect to data quality and workflow integration, however, is met by increased adoption of ML optimized solutions. Transfer learning, explainable AI, and applications to quantum computing are expected to become promising future developments. In this opening, these advancements will further propel boundaries in the engineering model optimization and produce more accurate, efficient and transparent solutions to the complex engineering problems. This is a big step forward in tackling the difficult analytical problems and solving them by the combination of traditional engineering expertise with machine learning optimization. Ultimately, this synergy facilitates engineers in tackling previously intractable problems while maintaining the required standards for engineering applications.

References:

[1]. Awang, Z.; Lim, S.H.; Awang, N.F. Pendekatan Mudah SEM: Structural Equation Modelling; MPWS Rich Publication: Selangor, Malaysia, 2018.

[2]. Hair, J.F.J.; Hult, G.T.M.; Ringle, C.M.; Sarstedt, M.; Danks, N.P.; Soumya, R. Review of Partial Least Squares Structural Equation Modeling (PLS-SEM) Using R: A Workbook; Springer Nature: Cham, Switzerland, 2021.

[3]. Nikolopoulou, K. Mobile learning usage and acceptance: Perceptions of secondary school students. J. Comput. Educ. 2018, 5, 499–519.

[4]. Brinda, T.; Napierala, S.; Behler, G.A. What do secondary school students associate with the digital world? In Proceedings of the WiPSCE' 18: Proceedings of the 13th Workshop in Primary and Secondary Computing Education, Potsdam, Germany, 4–6 October 2018; pp. 25–34.

[5]. Hostovecky, M.; Prokop, P. The relationship between internet addiction and personality traits in Slovak secondary schools students. J. Appl. Math. Stat. Inform. 2018, 14, 83–101.

[6]. McPhee, A. Problem-based learning in initial teacher education: Taking the agenda forward. J. Educ. Enq. 2002, 3, 60–78.

[7]. Woods, D.R. Problem-based learning for large classes in chemical engineering. In Bringing Problem-Based Learning to Higher Education: Theory and Practice; Wilkerson, L., Gijselaers, W.H., Eds.; Jossey-Bass: San Francisco, CA, USA, 1996; pp. 91–99.

[8]. Stinson, J.E.; Milter, R.G. Problem-based learning in business administration education: Curriculum design and implementation issues. In Bringing Problem-Based Learning to Higher Education: Theory and Practice; Wilkerson, L., Gijselaers, W.H., Eds.; Jossey-Bass: San Francisco, CA, USA, 1996; pp. 33–42.

[9]. Pittala, Chandra Shaker, et al. "Realization and comparative analysis of thermometer code based 4-bit encoder using 18 nm FinFET technology for analog to digital converters," *Soft Computing and Signal Processing: Proceedings of 3rd ICSCSP 2020, Volume 1*. Singapore: Springer Singapore, 2021. 557-566.

[10]. Walker, A.E.; Leary, H. A problem-based learning meta-analysis: Differences across problem types, implementation types, disciplines, and assessment levels. Interdiscip. J. Probl.-Based Learn. 2009, 3, 12–43.

[11]. Hmelo-Silver, C.E.; Derry, S.J.; Bitterman, A.H.; Hatrak, N. Targeting transfer in a STELLAR PBL course for pre-service teachers. Interdiscip. J. Probl.-Based Learn. 2009, 3, 24–42.

[12]. Goodnough, K. Issues in modified problem-based learning: A self-study in pre-service science teacher education. Can. J. Sci. Math. Technol. Educ. 2005, 5, 289–306.

[13]. Liu, Y.; Gayle, A.A.; Wilder-Smith, A.; Rocklöv, J. The reproductive number of COVID-19 is higher compared to SARS coronavirus. J. Travel Med. 2020, 5, 1–4.

[14]. Mizumoto, K.; Chowell, G. Transmission potential of the novel coronavirus (COVID-19) onboard the diamond Princess Cruises Ship, 2020. Infect. Dis. Model. 2020, 5, 264–270.

[15]. Ndaïrou, F.; Area, I.; Nieto, J.J.; Torres, D.F. Mathematical modeling of COVID-19 transmission dynamics with a case study of Wuhan. Chaos Solitons Fractals 2020, 135, 109846.

[16]. Ludwig, M.; Jesberg, J. Using Mobile Technology to Provide Outdoor Modelling Tasks—The MathCityMap-Project. Procedia—Soc. Behav. Sci. 2015, 191, 2776–2781.

[17]. Rajeev Ratna Vallabhuni, et al., "Comparative Validation of SRAM Cells Designed using 18nm FinFET for Memory Storing Applications," Proceedings of the 2nd International Conference on IoT, Social, Mobile, Analytics & Cloud in Computational Vision & Bio-Engineering (ISMAC-CVB 2020), 2020, pp. 1-10.

[18]. Zender, J.; Ludwig, M. The long-term effects of MathCityMap on the performance of German 15 year old students concerning cylindric tasks. In Proceedings of the Eleventh Congress of the European Society for Research in Mathematics Education; Jankvist, U.T., Van den Heuvel-Panhuizen, M., Veldhuis, M., Eds.; Freudenthal Group & Freudenthal Institute, Utrecht University and ERME: Utrecht, The Netherlands, 2019; pp. 3003–3010.

[19]. Gurjanow, I.; Oliveira, M.; Zender, J.; Santos, P.A.; Ludwig, M. Mathematics Trails: Shallow and Deep Gamification. Int. J. Serious Games 2019, 6, 65–79.

[20]. Pais, A.; Valero, P. Researching research: Mathematics education in the Political. Educ. Stud. Math. 2012, 80, 9–24.

[21]. Radford, L. On theories in mathematics education and their conceptual differences. In Proceedings of the International Congress of Mathematicians; Sirakov, B., de Souza, P., Viana, M., Eds.; World Scientific Publishing Co Roos: Singapore, 2018; Volume 4, pp. 4055–4074.

[22]. Radford, L.; Radford, L. Play and the production of subjectivities in preschool. In Mathematics Education in the Early Years. Results from the POEM4 Conference 2018; Carlsen, M., Erfjord, I., Hundeland, P.S., Eds.; Springer: Cham, Switzerland, 2020; pp. 43–60.

[23]. Nikolopoulou, K.; Akriotou, D.; Gialamas, V. Early Reading Skills in English as a Foreign Language Via ICT in Greece: Early Childhood Student Teachers' Perceptions. Early Child. Educ. J. 2019, 47, 597–606.

[24]. Hinojo-Lucena, F.J.; Aznar-Díaz, I.; Cáceres-Reche, M.P.; Trujillo-Torres, J.M.; Romero-Rodríguez, J.M. Factors Influencing the Development of Digital Competence in Teachers: Analysis of the Teaching Staff of Permanent Education Centres. IEEE Access 2019, 7, 178744–178752.

[25]. Kerrigan, J. Active learning strategies for the mathematics classroom. Coll. Teach. 2018, 66, 35–36.

[26]. Rajeev Ratna Vallabhuni, et al., "Comparative Analysis of 8-Bit Manchester Carry Chain Adder Using FinFET at 18nm Technology," 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), Thoothukudi, India, 2020, pp. 1579-1583.

[27]. Aznar, P.; Ull, M.Á.; Martínez, M.P.; Piñero, A. Competencias Básicas para la Sostenibilidad: Un Análisis desde el Diálogo Disciplinar. Bordón. Rev. Pedag. 2014, 66, 13–27.

[28]. Wiek, A.; Withycombe, L.; Redman, C.L. Key Competencies in Sustainability: A Reference Framework for Academic Program Development. Sustain. Sci. 2011, 6, 203–218.

[29]. Skovsmose, O. Towards a Philosophy of Critical Mathematics Education; Kluwer: Dordrecht, The Netherlands, 1994.

[30]. Black, S.; Spreen, C.; Vally, S. Education, COVID-19 and care: Social inequality and social relations of value in South Africa and the United States. South. Afr. Rev. Educ. 2020, 26, 40–61.

[31]. Statistics South Africa. Inequality Trends in South Africa: A Multidimensional Diagnostic of Inequality; Statistics South Africa: Pretoria, South Africa, 2019.

[32]. Bell, L. Theoretical foundations for social justice education. In Teaching for Diversity and Social Justice: A Sourcebook; Adams, M., Griffin, P., Bell, L., Eds.; Routledge: New York, NY, USA, 1997; pp. 3–26.