This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

Please cite this paper as:

Naser M.Z. (2022). ""CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members." *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.

CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members

M.Z. Naser, PhD, PE, M.ASCE
Glenn Department of Civil Engineering, Clemson University, Clemson, SC 29634, USA
AI Research Institute for Science and Engineering (AIRISE), Clemson University, Clemson, SC 29634, USA
E-mail: mznaser@clemson.edu, Website: www.mznaser.com

Abstract
This paper introduces CLEMSON, an automated machine learning (AutoML) virtual assistant (VA) that enables engineers to carry ac**C**e**LE**rated, si**M**ulation-free, tran**S**parent, reduced-**O**rder, and infere**N**ce-based fire resistance analysis with ease. This VA learns from physical observations taken from real fire tests to bypass bottlenecks and *ab initio* calculations associated with traditional structural fire engineering methods. CLEMSON leverages competitive ML algorithm search to identify those most suited for a given problem and then blend them into a cohesive ensemble to realize faster and reduced-order assimilation of predictions – thereby attaining higher accuracy and reliability. In addition, this VA is designed to be transparent and hence is supplemented with explainability measures to allow users to identify key factors driving its rationale and predictions. Once fully realized, CLEMSON augments its inner workings into a graphical user interface that can be used in a coding-free manner and with enriched visualization tools to allow users to directly harness the power of ML without the need for special software. To showcase the merit of the proposed VA, CLEMSON is applied to assess classification and regression problems by means of evaluating fire resistance rating, as well as temperature rise history and deformation history of concrete-filled steel tubular (CFST) columns via five algorithms, namely: Extreme Gradient Boosted Trees, Light gradient boosted trees, Neural Networks, Random Forest, and TensorFlow. Finally, this work also introduces three new and functional performance metrics that are explicitly derived for structural fire engineering applications and hence can be used to cross-check the validity of ML models.

Keywords: Machine learning (ML); Structural fire engineering; Ensemble; Columns; Explainable artificial intelligence (XAI).

Introduction
The advent of machine learning (ML) has launched exciting opportunities for engineers and scientists (Bishop 2007). One such opportunity that is of merit to explore is the development of ML-based virtual assistants (VA). These VAs can be thought of as tools with capabilities similar to those seen in their common counterparts of theoretical, analytical, or numerical (i.e., finite element (FE)) origins. In a way, ML-based VAs can be considered as a continuation of the natural evolution of existing models (Cuevas-Zuviría and Pacios 2020). At its basic form, one can think of VAs as means for accelerated and low-cost simulation-free tools to bypass the need for *ab initio* calculations (Botu and Ramprasad 2015).

This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

Please cite this paper as:

Naser M.Z. (2022). ""CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members." *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.

Adopting ML into the domain of structural fire engineering has been explored in recent years (Chaudhary et al. 2021; Fu 2020; Naser 2021a; Panev et al. 2021). The cited works, along with others (Chaudhary et al. 2021; Jiang et al. 2021; Naser 2019a; Ouache et al. 2021; Su et al. 2021), developed and applied different ML algorithms to examine a collection of structural engineering phenomena often triggered because of fires. A deep dive into the main findings of such works reveals two observations worthy of note: 1) a rise in the number of publications that leverage ML, and 2) ML seems to provide an attractive method of tackling structural fire engineering problems. Given the above two observations, together with those realized through ongoing discussions and conference proceedings (e.g., International Association for Fire Safety Science (McNamee et al. 2019) and the 2021 American Concrete Institute Spring Convention (ACI 2021)), the adoption of ML is expected to continue to soar in the coming years.

Despite the success of ML in this domain, as well as similar domains (Litman 2014; Tarassoli 2019), a few burning concerns continue to arise. For example, utilizing a particular algorithm to examine a phenomenon does not guarantee arriving at an ideal, let alone an optimal solution to a given problem. Building upon the *No Free Lunch Theorem*, which states that all optimization algorithms perform equally well when their performance is averaged across all possible problems, implies that there is not a single best algorithm (Wolpert and Macready 1997). Hence, it is of merit to explore multi-algorithmic search and ensemble learning to combine a series of algorithms into a ML model of higher accuracy, less vulnerability to overfitting, and better handling of missing or imbalanced data (Chou and Pham 2013; Polikar 2009).

Another concern that is tied to ML can be encompassed by the fact that many of the existing ML models are often labeled as "Blackboxes" with little insights into their inner mechanisms or the rationale/logic used to justify their predictions (Dosilovic et al. 2018). As such, ML models, if to be effectively and confidently utilized, are expected to be transparent to allow users to fully understand their inner workings and extract possible "true" causation they may be able to reveal. The notion of *transparency*, and by extension *explainability/interpretability*, is also necessary to establish liability and fairness since fire and engineering projects entail legal and human components. A dedicated discussion on explainable and interpretable ML models can be found elsewhere (Doran et al. 2018; Naser 2021b).

Scientists and engineers are expected to create intricate ML models with the growing need for larger and more accurate models. Such models are likely to be of complex topologies and in need of processing extensive databases. This translates into large energy consumption during training, development, storage, and deployment. Current works have estimated carbon emissions from training complex deep learning models to reach 150,000 Kg per model (Jackson 2019; Strubell et al. 2020). The reader is to note that 150,000 Kg of carbon emissions is equivalent to that emitted through the lifetime of five fuel-based vehicles (Strubell et al. 2020). Thus, and whenever possible, ongoing efforts are to favor energy-light algorithms or *reduced-order* techniques to allow the development of energy-friendly ML models (García-Martín et al. 2019).

In response to the lack of ML-based activities in existing curricula, fresh and senior scientists and engineers may not be well familiar with ML. This adds another layer of complication that may hinder the adoption and widespread of ML in this domain. Hence, it is thought of merit to develop ML tools that can be augmented into software, or applications (i.e., Apps), with easy-to-use graphical user interfaces (GUI) (Dudley and Kristensson 2018).

A robust ML model not only performs well during its training process but also continues to perform well beyond its training and into its deployment. Several challenges could arise during the deployment of a ML model. For instance, the model may face scenarios that were not part of its training (e.g., beyond the range of data used during its development), which may jeopardize its effectiveness. Furthermore, the quality of real-life data may be of lower caliber or lesser completeness, and hence such a model could struggle in attaining similar prediction capability to that obtained during its training. Thus, there is a need to continue to monitor model performance to improve its predictive capability. This enables a given ML model to attain and possibly improve its *inference* capability (Wu et al. 2019).

This work builds upon the above discussion and creates a blueprint for an AutoML-based VA (named CLEMSON) that can be used by scientists and engineers. Simply put, this paper introduces CLEMSON via a case study on CFST columns under fire conditions. In this case study, CLEMSON evaluates fire resistance rating, temperature rise, and deformation history of concrete-filled steel tubular (CFST) columns. In this pursuit, five algorithms, namely: Extreme Gradient Boosted Trees, Light gradient boosted trees, Neural Networks, Random Forest, and TensorFlow, are examined to enable accelerated, simulation-free, transparent, reduced-order, and inference-based analysis. The proposed framework is also supplemented with three new and functional performance metrics explicitly derived for fire resistance applications.

Framework and Technical Details of CLEMSON AutoML

This section articulates an overview of the big ideas behind CLEMSON, together with supplementary technical details.

Big Ideas Behind CLEMSON AutoML

In many scenarios, traditional (or simple) ML models have been reported to perform well when applied to structural fire engineering problems (Naser 2019b; Panev et al. 2021; Wu et al. 2021). Such models are not only easy to use but are also well-vetted. For the sake of this discussion, simple ML models can be thought of as 2D FE models that can perform well for general problems and hence negate the need for their more complex counterparts of 3D FE models. Adopting such models can come in handy as they nullify the need for developing overly complicated ML models.

On the other hand, simple models may not be able to solve the problem on hand or at least perhaps fail to attain good performance scores. In parallel, simple models may not correctly uncover the underlying mechanisms of a given problem. The same can also trigger a few issues related to the poor performance of ML models in the long run (i.e., overfitting, bias, etc.).

In some instances, the choice behind selecting a ML algorithm may stem from a personal preference (or simply due to the user's familiarity with such an algorithm/programing language; in a similar manner to selecting a finite element (FE) package (i.e., ANSYS vs. ABAQUS)). Such a practice is not always tied to realizing the best possible solution nor gives the user/stakeholder a chance to vet the performance of their algorithm against other algorithms. Thus, it is of merit for the interested user to explore a variety of ML algorithms (especially those incorporating a mixture of architectures, and topologies, etc.) (Schmidhuber 2015). In such an approach, a tournament between selected algorithms can be carried out to enable the ranking of ML models into a *"leaderboard"* according to pre-specified criteria (i.e., performance metrics).

Based on the above discussion, it may be inevitable to develop exotic ML models (e.g., complex deep neural networks), especially when simple models fail to perform adequately. These models require expensive resources. In fact, a recent work has noted that the development of exotic models cost can exceed $1.0 million per model and can take months/years to develop (Jackson 2019; Strubell et al. 2020). A more cynical look into such models is their need for user expertise to guide the development and deployment of these models, which may overwhelm users from our domain. Despite the higher costs of exotic models, they can be essential to overcoming specialized problems.

A workaround exotic models is to leverage ensemble learning. In this methodology, a collection of ML algorithms can be merged/blended into an ensemble that harnesses the positive advantages of each algorithm and minimizes their collective disadvantages. Nowadays, ensembles can be much easier and cost-friendlier to develop than exotic models. Ensembles have been noted to outperform sole and deep learning models in complex settings (Abdollahi-Arpanahi et al. 2020; Ganaie et al. 2021; Hamori et al. 2018).

Another dimension to ML adoption is the ability of the user (or engineer, for that matter) to understand the reasoning behind a model's prediction. Unlike other simulation techniques such as the FE method, which is guided by physics principles and follows intuitive/rule-based explanations, ML continues to be primarily driven by the supplied data (and hence the notion of data-driven analysis). Thus, there is a need to allow users to understand the reasons behind a model's or ensemble's predictions. Peeking into a model/ensemble behavior brings the missing "trust" component between the user and ML into the picture, a key issue that haunts Blackbox ML models. The capability to explain or interpret ML models can be carried out by adopting modern techniques such as feature importance (Altmann et al. 2010), partial dependence plots (Scikit 2021), Shapley values (Boehmke et al. 2020), among others.

Furthermore, ML models often lack a graphical user interface that allows a user (as opposed to the model's developer) to modify its architecture or to update its settings. This brings in a few issues; 1) lack of transparency, 2) poor implementation/distribution, and 3) reiterates the need for extensive coding knowledge. Therefore, it is of importance to develop accessible ML models that can be easily tweaked and applied.

This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

Please cite this paper as:

Naser M.Z. (2022). ““CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members.” *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.

To overcome the above challenges, CLEMSON builds upon the success of ML-based platforms that enables the automated development of ML models. In this platform, CLEMSON develops a series of ML algorithms to explore the phenomenon on hand. Then, this VA identifies the best scoring models. Once ranked, these models can then be used individually or can also be merged into an ensemble. In a way, CLEMSON fosters the above discussed principles that: 1) sole/simple models can be used to examine a problem if deemed satisfactory, 2) carrying out a comparative and ensemble ML analysis is potentially tied to realize improved performance over than any of the component models separately (Hindman 2015), 3) advocating for explainable ML, and 4) creating easy-to-use ML-based tools that are coding-less (e.g., with simple GUI). Figure 1 demonstrates the inner workings of CLEMSON.

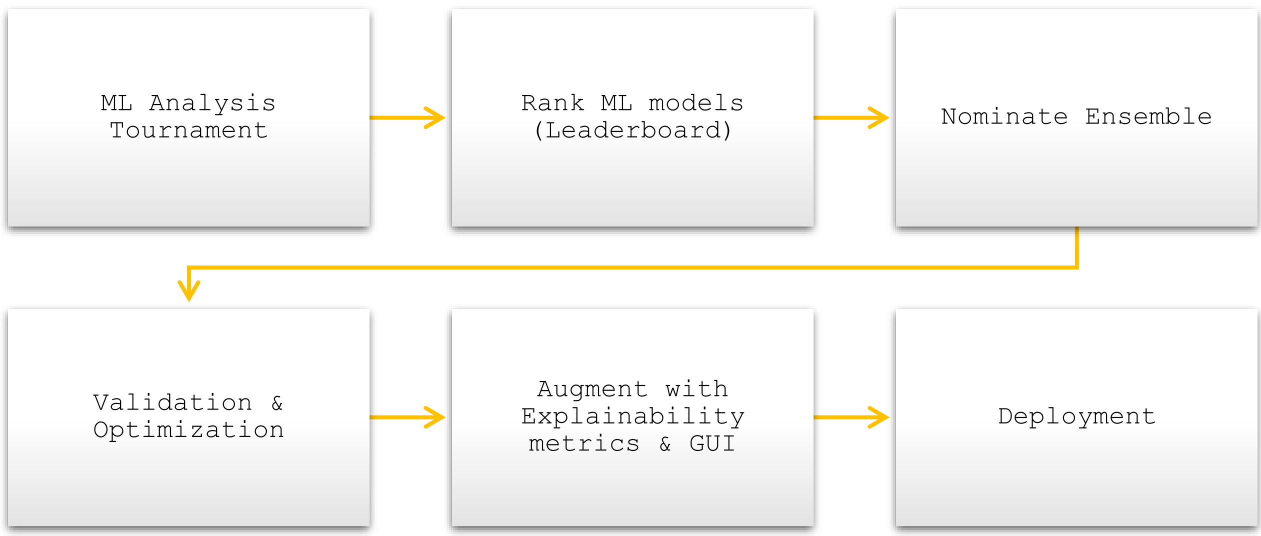


Fig. 1 Flowchart of CLEMSON

Selected Machine Learning Algorithms

Five algorithms are selected for showcasing the applicability of CLEMSON herein. These algorithms are: Extreme Gradient Boosted Trees (ExGBT), Light Gradient Boosted Trees (LGBT), Keras Residual Neural Network (KNN), Random Forest (RF), and TensorFlow Deep Learning (TFDL). The algorithms are briefly discussed herein, given that their full descriptions can be found in their respective references and in (Hastie et al. 2011; Ketkar and Ketkar 2017). The reader is encouraged to remember that CLEMSON can incorporate other algorithms as well. In a way, this study does not focus on a particular set of algorithms, but rather these are thought of as mechanisms to operate CLEMSON.

<u>Extreme Gradient Boosted Trees (ExGBT)</u>

The ExGBT algorithm arranges the collected data into a tree format. In this format, each tree examines a sampled subset in each iteration of analysis (Freund and Schapire 1997). In each iteration, this algorithm ties successive trees to residual errors to focus the analysis on the most challenging predictions. This algorithm was supplemented with the following settings: learning

This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

Please cite this paper as:

Naser M.Z. (2022). ""CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members." *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.

rate of 0.02, maximum tree depth of 5, subsample feature of 0.8, and 500 for the number of boosting stages. The ExGBT algorithm can be found online at (Scikit 2020a; XGBoost Python Package 2020).

Light Gradient Boosted Trees (LGBT)

Light gradient boosted trees (LGBT) is a newer algorithm to ExGBT and was developed by Microsoft. This algorithm requires little processing and has been noted to realize faster convergence and better handling of data (Freund and Schapire 1996). Like ExGBT, the LGBT successfully fits the residual errors from all previous iterations while incorporating two additional techniques to improve its performance. These techniques are known as gradient-based one-side sampling (to skips less informative points), and exclusive feature bundling (to group features in a near-lossless way) (Naser 2021a). The used algorithm can be found at (LightGBM 2020) with the following settings: learning rate = 0.1, maximum depth = "none", number of boosting stages = 500, with a *Sigmoid* activation function.

Keras Residual Neural Network (KNN)

Keras is an open-source library for developing neural network architectures (Li et al. 2018). In a residual architecture, a direct connection links data points to the target (response) variable to enable improved optimization by smoothing out the loss function. In the used KNN, a learning rate of 0.03 was used, along with a *Softmax* activation function, one layer containing 64 neurons. KNN can be readily found at (Keras 2020).

Random Forest (RF)

The Random Forest (RF) algorithm is a classical ensemble learner that creates a sequence of decision trees (Liaw and Wiener 2002). The RF applies the majority voting principle. In this principle, the average result of all trees is calculated to arrive at a final outcome. The RF algorithm can be found at (Scikit 2020b) with the following settings; number of trees = 500, Gini impurity to facilitate quality of a split, a maximum depth of "none", minimum leaf size, and maximum size for splits equals to 10 and "none", respectively.

TensorFlow Deep Learning (TFDL)

A TFDL is an open-source and free neural network-based model that uses Deep Learning and is developed by Google (Abadi 2016). The used algorithm has neurons in each layer = 100, number of training examples = 100, optimizer = *Conjugate gradient*, early stopping window = 10, adaptive learning rate, and activation function of *ReLu* – and can be found at (TensorFlow 2020).

Training and Evaluation Procedure

In a ML evaluation, the adopted training procedure can be elemental to the success of the developed analysis, and by extension, the well-being of its outcome (e.g., predictivity, etc.). In addition, it is during the training process that the hyperparameters of ML models can get tuned (i.e., via a random search method as currently implemented by CLEMSON, etc.) (Peskova and Neruda 2019). Hence, the developed model/ensemble is required to minimize flaws attributed to data handling and training. Proper data handling ensures that the collected data is cleansed of

This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

Please cite this paper as:

Naser M.Z. (2022). ""CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members." *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.

errors, outliers, and missing items and spreads across a wide range of practical applications, etc. (Witten et al. 2016).

Appropriate training of a ML model starts by randomly shuffling and splitting the database into three sets (T: training, V: validation, and S: testing) of preferably unequal proportions. The largest two sets are often reserved for training and validating the model. The model is then independently tested against the last set (since it was not involved in the training and validation procedure). On the other hand, a *k*-fold cross-validation procedure can also be used. In such a procedure, the collected dataset is randomly split up into test and training sets of *k* groups, wherein the model is trained using *k*-1 sets and then validated on the last *k* set. This procedure is repeated *k* times until each unique set has been used as the validation set. Finally, the model performance is evaluated on the test dataset (which was kept away during the training procedure).

Adopting the *k*-fold cross-validation method allows the model to train and to be validated on multiple datasets, which usually results in a more accurate model with better generalization abilities (less overfitting) than if the model is to be trained, validated, and tested on splits of a limited number of samples, and distributions. While CLEMSON can use any of the above two methods to train ML models, the results from a 10-fold cross-validation methodology is showcased in this study.

Once a model/ensemble completes the training and testing procedure, its predictive capability can be examined via performance metrics. These metrics quantify the variation between model predictions to actual measurements using mathematical constructs (Cremonesi et al. 2010; Laszczyk and Myszkowski 2019; Schmidt and Lipson 2010). Given the derivation and nature of such constructs, metrics often have advantages and disadvantages, which complicates the selection of metrics. Thus, it is best to examine the predictive capability of ML models via a series of comparisons pertaining to metrics of varying settings (see Table 1) (Botchkarev 2019; Naser and Alavi 2020). All in, such metrics are often applied at the global level of ML model; as in to compare the performance of the model across all, or a portion of, its predictions.

Table 1 List of selected performance metrics.

Metric	Expression
Classification	
Area under the ROC curve (AUC)	$AUC=\sum_{i=1}^{N-1}\frac{1}{2}(FP_{i+1}-FP_i)\,(TP_{i+1}-TP_i)$ *where, FP: number of false positives, TP: number of true positives.*
Balanced Accuracy (BA)	$BA=\frac{\frac{TP}{TP+FN}+\frac{TN}{FP+TN}}{2}$ *where, TN: number of true negatives.*
Log Loss Error (LLE)	$LLE=-\sum_{c=1}^{M}A_i logP$ *where, M: number of classes, c: class label, y: binary indicator (0 or 1) if c is the correct classification for a given observation.*

This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

Please cite this paper as:

Naser M.Z. (2022). ""CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members." *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.

Cumulative Clemson Metric (CCM)	$\dfrac{no. of\ obsevrations\ with\ (\frac{FR_{predicted} - FRR_{actual}}{60} = 0.0)}{total\ number\ of\ observations}$
Regression	
Mean Absolute Error (MAE)	$MAE = \frac{\sum_{i=1}^{n}\lvert E_i\rvert}{n}$
Symmetric Mean Absolute Percentage Error (SMAPE)	$SMAPE = \frac{100}{n}\sum_{i=1}^{n}\lvert E_i\rvert/(P_i + A_i)\ /2$
Root Mean Squared Error (RMSE)	$RMSE = \sqrt{\frac{\sum_{i=1}^{n} {E_i}^2}{n}}$
Coefficient of Determination (R^2)	$R^2 = 1 - \sum_{i=1}^{n}(P_i - A_i)^2 \ / \sum_{i=1}^{n}(A_i - A_{mean})^2$

A: actual measurements, *P*: predictions, *n*: number of data points, *E* = *A*-*P*.

In this work, three classification metrics and four regression metrics are used. The classification metrics are Balanced Accuracy (BACC), Area under the receiver operating characteristic curve (AUC), and Log Loss Error (LLE). The BACC is often used in binary and multiclass classification problems to deal with imbalanced datasets (such as that noted herein to identify fire resistance rating (FRR) of CFST columns such as 60 min, 120 min, 180 min, and 240 min). Hence, higher values of BACC are favorable as they imply good class predictivity. An area of unity for the AUC indicates a perfect score. The LLE penalizes for being confident in the wrong prediction, with a lower value for log loss being favorable.

In addition to the aforenoted traditional metrics, a new functional metric is developed herein. Unlike the other metrics, which give statistical insights to the predictivity of the ML model at the global level, the newly developed metric considers the characteristics of the fire problem undertaken herein and is applied to *individual observations*. This metric examines the predicted FRR obtained from a ML model to the observed FRR from fire tests – as seen in Eq. 1. For example, say that a ML model predicts FRR of a given RC column to be 120 min; however, this column has 60 min as FRR (i.e., failed within 60 min during a fire test). In this particular example, the new metric (conveniently named Clemson Metric (CM)) turns a non-zero value; implying poor fitness.

$$Clemson\ Metric\ (CM) = \frac{FRR_{predicted} - FRR_{actual}}{60} \qquad \text{Eq. 1}$$

where, CM is applicable for FRR ranging between 0.0-240 min, with a zero as a favorable performance that suggests a correct ML prediction. Negative scores of CM indicate conservative predictions, and positive scores indicate unconservative predictions.

A companion metric to CM is the Cumulative CM (CCM), and this metric extends CM from individual observations to the global level of the ML model. The CMM compares the percentage of all observations that pass the CM metric (i.e., return zero) to the total number of observations examined by the ML model. Hypothetical cut-offs of <25%, 25-50%, 50-75%, and >75% (i.e., the model predicts the correct FRR for all elements in the database with *XX*% accuracy or higher) are deemed of “*poor*”, “*fair*”, “*strong*”, and “*excellent*” fitness, respectively. In parallel to other traditional metrics, the CCM can be applied for training, validation, and testing stages, as well as to the whole database.

$$Cumulative\ Clemson\ Metric\ (CCM) = \frac{no.of\ obsevrations\ of\ (\frac{FRR_{predicted} - FRR_{actual}}{60} = zero)}{total\ number\ of\ observations} \quad \text{Eq. 2}$$

On the other hand, the regression metrics are Mean Absolute Error (MAE), Symmetric Mean Absolute Percentage Error (SMAPE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2). Briefly, MAE evaluates the arithmetic average of the absolute errors and can be used for different scales. SMAPE estimates the error as a percentage and has a range of 0-200%, while RMSE evaluates the standard deviation of residuals in a scale-independent order. Finally, R^2 is a scale-free metric that assesses the degree of association between measured and predicted values.

Noting that the proposed ensemble will also be used to predict the *continuous response* of CFST columns (i.e., temperature-time and deformation-time history) as opposed to a specific value (say FRR), then predictions from the ML ensemble are recommended to be cross-checked via a new indicator named as the “maximum response deviation” (MRD). This new indicator evaluates ML predictions within a pre-specified range of real temperature-time and deformation-time observations. This indicator builds bounds based on the largest observation obtained (i.e., the temperature at core or displacement in the compression stage), then divides its value over equally spaced points in time of the duration of fire test (taken as 10 min herein). The outcome of this operation yields an upper and lower bound, wherein predictions from the model are to lay within. Model predictions are then compared across these bounds. If all predictions lay within the bound, then the model is said to have a perfect match. If <25%, 25-50%, 50-75%, and >75% of model predictions fall within these bounds, then the model is said to be of “poor”, “fair”, “strong”, and “excellent” match with observations. In a way, the MRD metric provides a dynamic indicator that relates to the observed performance of a single element and checks predictions against bounds derived for that specific element (vs. an arbitrary bound say at 5% error, or 10% error, etc.) to measure the quality of the model fitness. Examples of the newly developed metrics are provided in a later section.

In CLEMSON, ML models are ranked via their scores in each of the above metrics, and the highest-ranking model in all metrics gets to the top of the leaderboard. The best performing models are then blended into an ensemble. For regression problems, the ensembling method used herein averages the predictions of the best three ranked algorithms in CLEMSON’s Leaderboard. On the other hand, the same ensemble applies the majority vote mechanism for classification problems. For example, say that two algorithms predict that a column fails at 60 min, and one predicts that the same column fails at 120 min. In this instance, the ensemble would predict that the column fails at 60 min (i.e., 2 votes vs. 1 vote), and hence the use of an odd number of algorithms becomes handy. For completion, say that a user options to use an even number of algorithms, then a

This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

Please cite this paper as:

Naser M.Z. (2022). ""CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members." *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.

workaround would be to weigh predictions from the higher ranking algorithms with higher weights to allow for a weighted voting mechanism that would break any arising ties.

Deployment

Once the models and ensemble complete their evaluation procedure, the ensemble is then deployed into production. In this study, CLEMSON starts the analysis by classifying the anticipated fire resistance of a particular column to fall into either 60, 120, 180, or 240 min. Then, CLEMSON goes on to map/predict the temperature-time and deformation-time response of such a column via a supervised learning manner that can be viewed in a GUI. The GUI can be in terms of software-like or application-like (App) interface, or that of a simpler interface such as spreadsheets. The notion of GUI is set to allow cross-platform deployment of the developed ML ensemble. The use of classical spreadsheets can be of high merit since most practicing users already utilize such spreadsheet programs (Excel, etc.) and hence is showcased in this study. Finally, the ensemble is augmented with explainability tools during this stage.

Maintenance and Upkeep

A good practice is to keep an eye on the performance of the developed model/ensemble while being deployed. In the instance that model performance degrades with the addition of new observations, then the user may opt to retrain the model. In this work, an ensemble is expected to be periodically maintained throughout its lifecycle.

Future Features

Future versions of CLEMSON will be expected to support data handling and feature engineering, tools for model monitoring, and support for unsupervised learning problems.

Description of Database

This section describes the database to be used in this work as a case study. This database covers CFST columns that were exposed to standard fires. Further information on this database is outlined herein.

In total, 102 columns were collected from the open literature (Han et al. 2009, 2003; Lie and Chabot 1992; Wan et al. 2017). All selected columns were of hot-rolled tubing and filled with concrete materials of varying strength of 23-100 MPa. The temperature-time and deformation-time of all columns were obtained from their original resources (see Table 2 and Fig. 2). The frequency of all selected features is shown in Fig. 2, along with the correlation matrix and the scatter matrix of all features. The following features for each column were also collected;

<u>Geometric features:</u>
1) column diameter, D,
2) column slenderness, n,
3) tube thickness, t,

<u>Materials features:</u>
4) concrete compressive strength, f_c,

This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

Please cite this paper as:

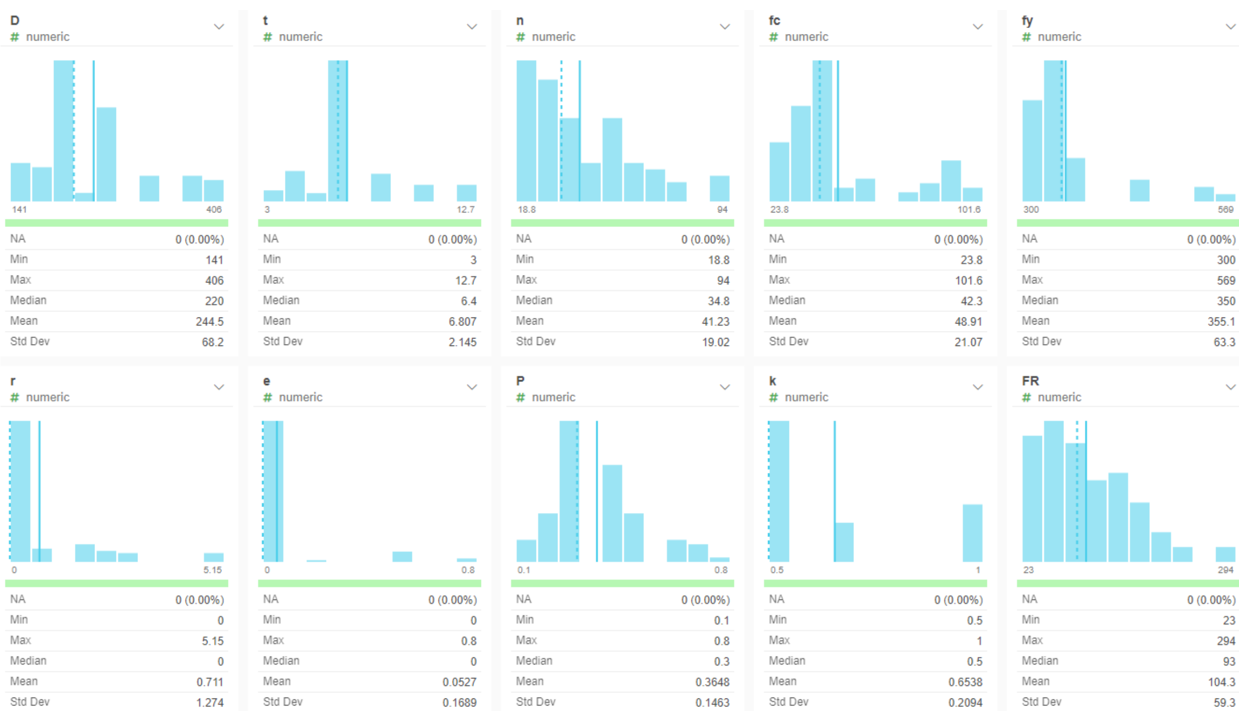
Naser M.Z. (2022). "“CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members.” *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.

5) yield strength of steel, f_y,
6) steel reinforcement ratio, r,

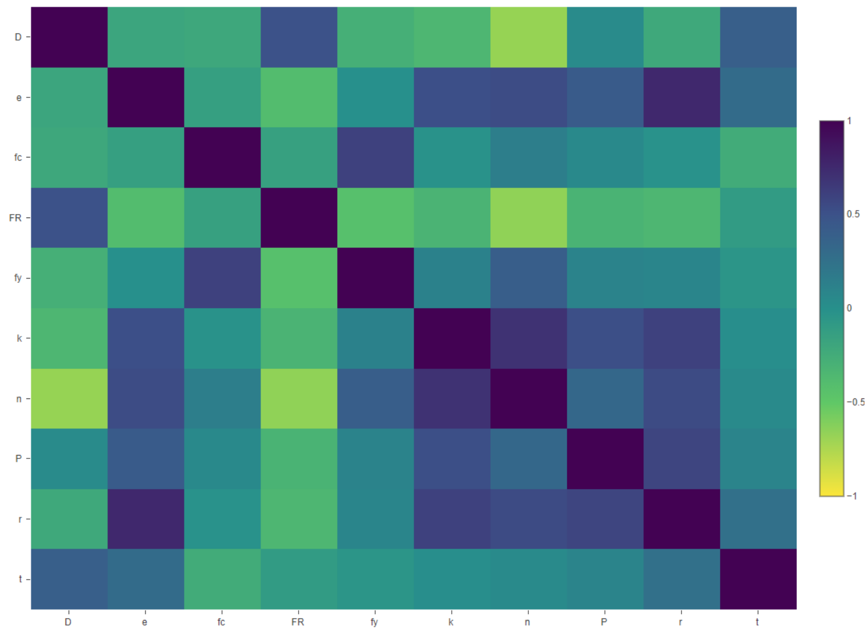
Loading features:
7) restraint conditions, K,
8) eccentricity, e,
9) level of applied loading, P,

Table 2 Statistics on collected database for CFST columns

	D (mm)	t (mm)	n	f_c (MPa)	f_y (MPa)	r (%)	e (%)	P (%)	FR (min)
Minimum	141.0	0.5	18.8	23.8	300.0	0.0	0.0	0.1	23.0
Maximum	406.0	1.0	94.0	35.5	569.0	5.2	0.8	0.8	294.0
Average	244.5	0.7	41.2	30.6	355.1	0.7	0.1	0.4	104.3
Standard deviation	67.8	0.2	18.9	3.3	62.9	1.3	0.2	0.1	59.0
Skewness	0.7	0.9	1.0	-0.4	1.9	2.0	3.2	0.6	0.8



(a) Frequency of features



(b) Correlation matrix

(c) Scatter matrix

Fig. 2 Insights into the selected database

Model Performance

As discussed in an earlier section, the performance of the developed ensemble is examined on two fronts. The first is against performance metrics that are listed in Table 1, and the second is by applying two sets of visual comparisons (via goodness of agreement between predicted and observed histories and through "maximum response deviation" (MRD) indicator) to examine predictions from the ensemble across the full temperature-time and deformation-time history of a collection of CFST columns.

As one can see from the results listed in Table 3, the developed ensemble outperforms that of the sole ML models in the majority of comparisons and for most of the selected performance metrics in the cases of predicting FRR and deformation-history of fire exposed CSFT columns. In parallel, the ensemble scores a close second to the LGBT model in capturing the temperature-time history of columns as a function of fire exposure. This behavior is due to the ensemble averaging the outcome of the three highest ranking models applied herein ExGBT, RF, and LGBT. For the sake of this discussion, a user could have opted to use the LGBT model instead of the ensemble since each examined phenomenon is independent of the others; however, this practice was not followed herein to maintain a more coherent comparison using one ensemble.

In addition to the above examination listed in Table 3, which primarily targets the overall predictivity of the ensemble, an additional layer of validation is applied in Fig. 3. This additional validation applies a visual perspective to the performance of the predicted temperature-time and deformation-time when compared to observations from fire tests. Figure 3 shows that the comparison between actual observations from fire tests and those predicted by ML are in good agreement. Overall, predictions from the ML ensemble match the trends of temperature rise and deformation progression during the fire.

Special attention can be paid to ensemble predictions towards the point of failure of each column. It is at this point that ensemble predictions can deviate a bit from those observed. This is due to the fact that CLEMSON, at this stage, ties FRR (i.e., 60, 120, 180, and 240 min) as a failure criterion associated with each column – as opposed to the actual failure of the column. Tying the failure to fire rating continues to be the practice of choice thus far, and hence is adopted here. A look into ensemble predictions at the point of FRR shows a much closer accuracy than that at the point of failure – implying that CLEMSON satisfies its pre-set settings for most of the comparisons. Future versions of CLEMSON can be tweaked to follow a more performance-based approach.

In addition, Fig. 3 also provides a new lens to another set of visual comparisons by applying the "maximum response deviation" (MRD) indicator to examine predictions from the ensemble across the full temperature-time and deformation-time history of a sample of CFST columns. As one can see, this figure shows how the predictions from ML fall within the MRD bounds as an additional means to assess ML predictions at the micro-level (specific intervals of 10 min each). Figure 3 also displays a comparison between the different MRD scores and how such scores quantify the degree of agreement between predictions and real observations. Finally, the reader may notice that the bounds for MRD tend to be narrower with longer fire exposures. This is natural as it reflects the derivation process of this dynamic metric discussed earlier, which subjects scrutiny to columns surviving long fires since accurately capturing fire response to such columns is often complex as they tend to undergo convoluted conditions such as creep etc. (Buchanan and Abu 2017).

This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

Please cite this paper as:

Naser M.Z. (2022). ""CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members." *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.

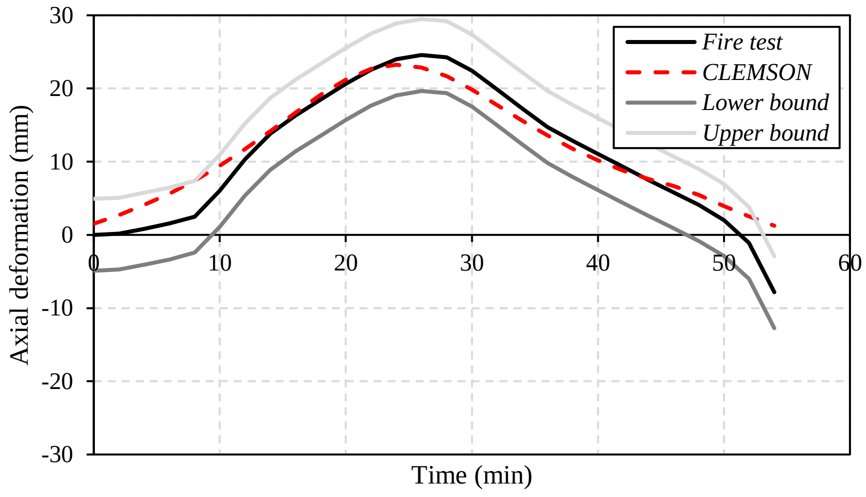
Table 3 List of selected performance metrics (Note – *T*: Training sample, *V*: Validation sample, *S*: Testing sample).

Fire Resistance Rating																			
Metric		***Ensemble***			***ExGBT***			***LGBT***			***KNN***			***RF***			***TFDL***		
		T	***V***	***S***	***T***	***V***	***S***	***T***	***V***	***S***	***T***	***V***	***S***	***T***	***V***	***S***	***T***	***V***	***S***
AUC	$AUC = \sum_{i=1}^{N-1} \frac{1}{2}(FP_{i+1} - FP_i)(TP_{i+1} - TP_i)$	*0.906*	*0.781*	*0.817*	0.867	0.771	0.810	0.860	0.758	0.746	0.835	0.767	0.813	0.797	0.751	0.658	0.895	0.766	0.752
Balanced Accuracy	$BA = \frac{\frac{TP}{TP+FN} + \frac{TN}{FP+TN}}{2}$	*0.505*	*0.421*	*0.427*	0.447	0.350	0.300	0.443	0.374	0.360	0.426	0.412	0.427	0.378	0.355	0.260	0.493	0.405	0.300
Log Loss (LL) Error	$LLE = -\sum_{c=1}^{M} A_i log P$	*0.967*	*1.132*	*1.186*	0.995	1.165	1.213	0.957	1.184	1.266	1.003	1.179	1.192	1.143	1.241	1.431	1.140	1.317	1.514
Cumulative Clemson Metric (CCM)	$\frac{no.of\ obsevrations\ with\ (\frac{FR_{predicted} - FRR_{actual}}{60} = 0.0)}{total\ number\ of\ observations}$	0.732	*0.800*	*0.671*	0.711	0.701	0.621	0.683	0.651	0.476	0.651	0.600	0.471	0.516	0.556	0.333	*0.783*	0.700	0.571
Deformation-time history																			
Metric		***Ensemble***			***ExGBT***			***LGBT***			***KNN***			***RF***			***TFDL***		
		T	***V***	***S***	***T***	***V***	***S***	***T***	***V***	***S***	***T***	***V***	***S***	***T***	***V***	***S***	***T***	***V***	***S***
Mean Absolute Error (MAE)	$MAE = \frac{\sum_{i=1}^{n}\lvert E_i\rvert}{n}$	*1.184*	*1.793*	*2.543*	1.509	1.878	2.857	1.857	2.479	3.267	4.752	5.667	6.428	3.698	4.038	5.149	2.802	3.337	4.653
Symmetric Mean Absolute Percentage Error (SMAPE)	$SMAPE = \frac{100}{n}\sum_{i=1}^{n}\lvert E_i\rvert/(P_i + A_i)/2$	36.247	*35.37*	*44.60*	*34.77*	38.30	49.64	47.23	46.34	55.45	73.94	76.67	92.25	68.94	61.85	69.52	60.00	56.08	75.93
Root Mean Squared Error (RMSE)	$RMSE = \sqrt{\frac{\sum_{i=1}^{n} E_i^2}{n}}$	*2.008*	3.603	8.182	2.619	*3.589*	*8.097*	3.106	4.607	8.188	5.732	8.057	11.078	4.638	6.210	10.612	3.673	5.173	9.325
Coefficient of Determination (R^2)	$R^2 = 1 - \sum_{i=1}^{n}(P_i - A_i)^2 / \sum_{i=1}^{n}(A_i - A_{mean})^2$	*0.964*	*0.903*	*0.712*	0.938	0.898	0.711	0.913	0.850	0.712	0.704	0.571	0.472	0.806	0.736	0.516	0.878	0.812	0.626
Temperature-time history																			
Metric		***Ensemble***			***ExGBT***			***LGBT***			***KNN***			***RF***			***TFDL***		
		T	***V***	***S***	***T***	***V***	***S***	***T***	***V***	***S***	***T***	***V***	***S***	***T***	***V***	***S***	***T***	***V***	***S***
Mean Absolute Error (MAE)	$MAE = \frac{\sum_{i=1}^{n}\lvert E_i\rvert}{n}$	17.563	16.296	17.498	18.129	16.127	18.256	*14.446*	*14.967*	*15.646*	27.449	27.145	30.747	24.392	22.148	22.299	82.871	73.253	69.560
Symmetric Mean Absolute Percentage Error (SMAPE)	$SMAPE = \frac{100}{n}\sum_{i=1}^{n}\lvert E_i\rvert/(P_i + A_i)/2$	14.547	13.510	12.705	13.594	12.983	12.834	*12.840*	*12.802*	*12.321*	23.334	23.095	22.037	19.756	18.411	15.837	50.067	47.663	42.015
Root Mean Squared Error (RMSE)	$RMSE = \sqrt{\frac{\sum_{i=1}^{n} E_i^2}{n}}$	27.271	23.567	25.593	27.709	23.259	28.196	*24.601*	*22.463*	*23.254*	36.488	37.553	42.835	36.491	32.112	32.711	94.179	85.993	83.146
Coefficient of Determination (R^2)	$R^2 = 1 - \sum_{i=1}^{n}(P_i - A_i)^2 / \sum_{i=1}^{n}(A_i - A_{mean})^2$	0.954	0.958	0.957	0.953	0.959	0.948	*0.963*	*0.962*	*0.965*	0.918	0.892	0.881	0.918	0.922	0.931	0.453	0.439	0.553

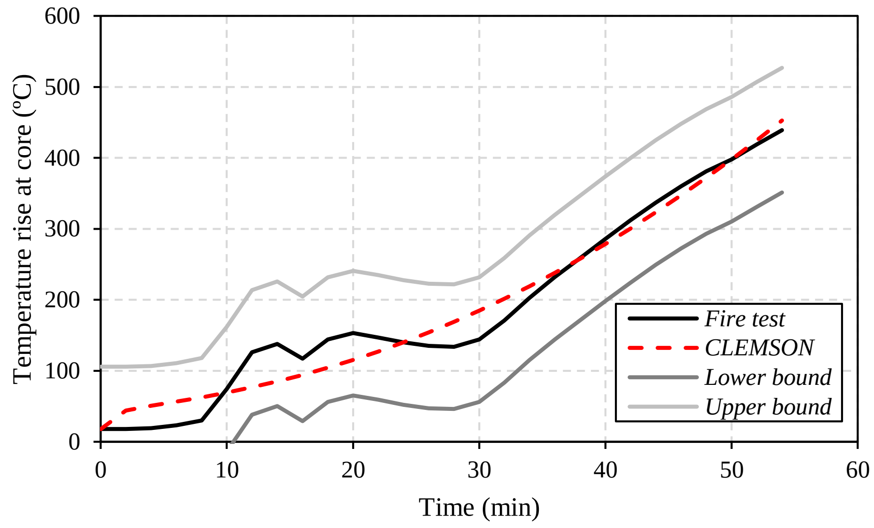
This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

Please cite this paper as:

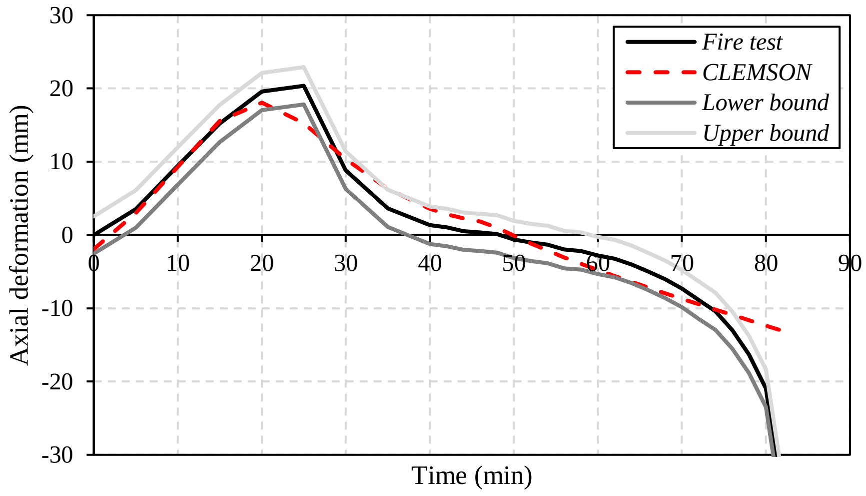
Naser M.Z. (2022). ""CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members." *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.



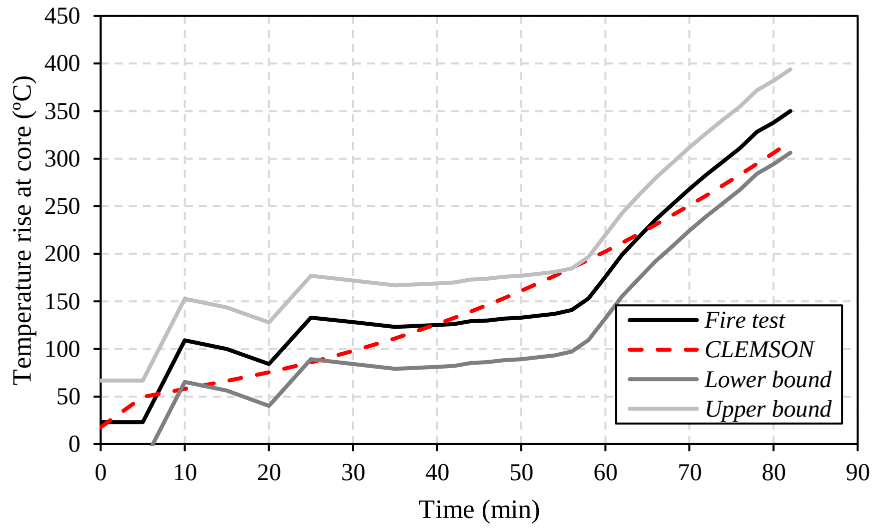
(a) C2 [MRD 93% (strong agreement)] [FRR less than 60 min]



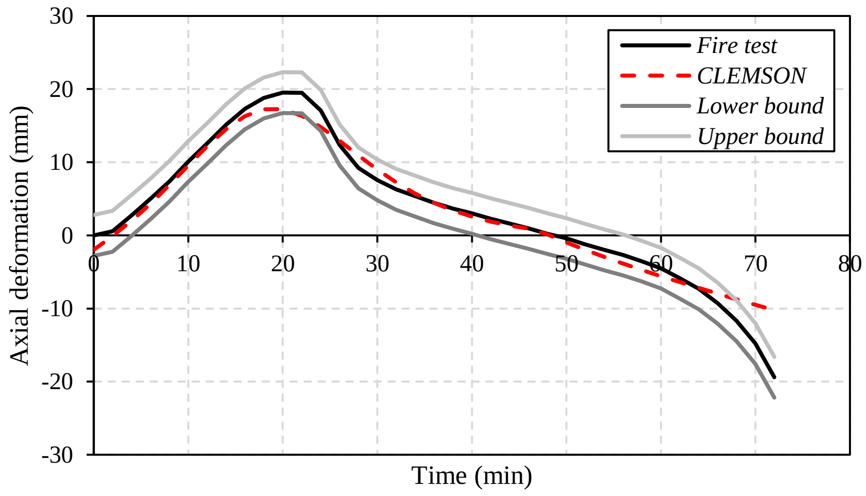
(b) C2 [MRD 100% (strong agreement)] [FRR less than 60 min]



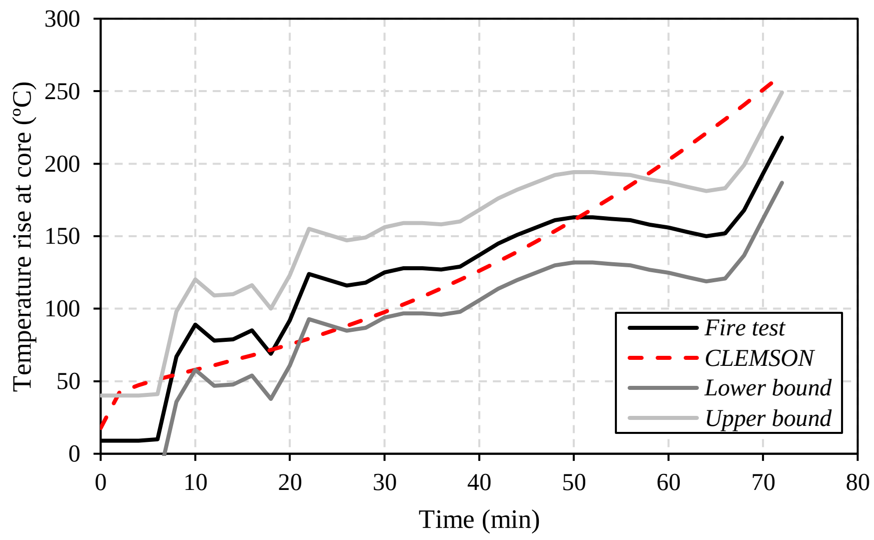
(c) C17 [MRD 90% (excellent agreement)] [FRR at 60 min]



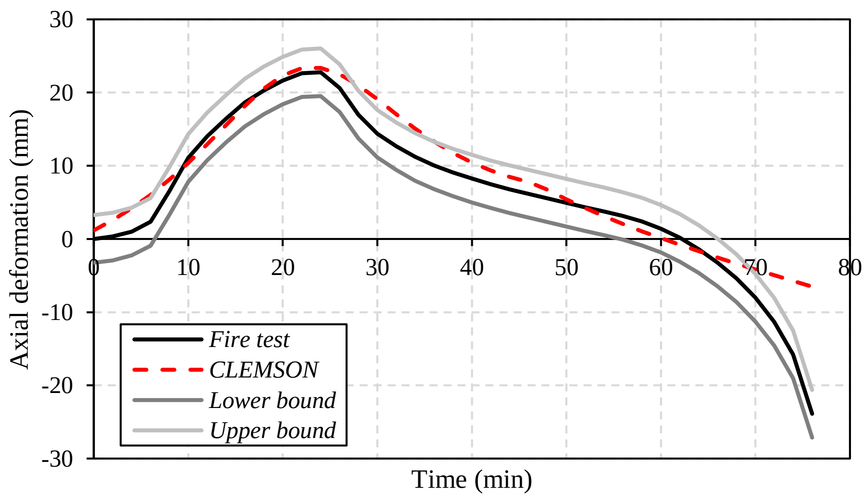
(d) C17 [MRD 89% (excellent agreement)] [FRR at 60 min]



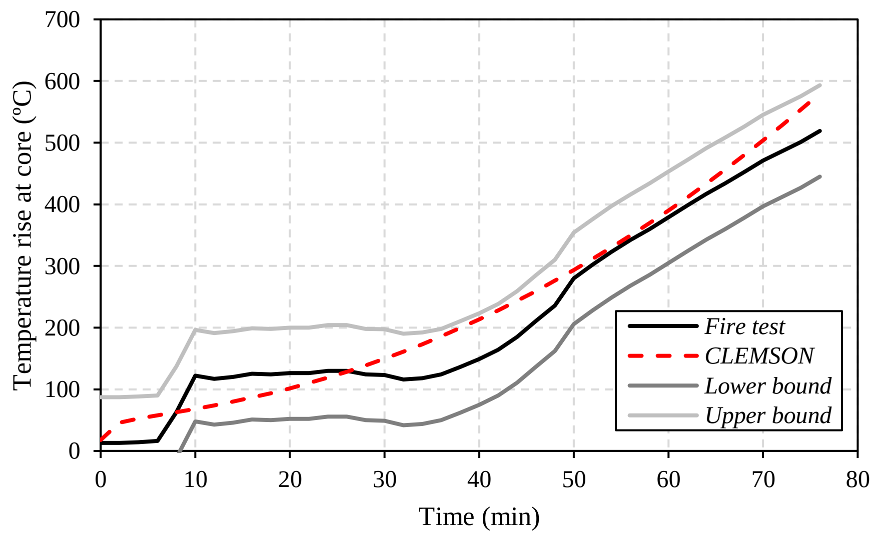
(e) C15 [MRD 89% (excellent agreement)] [FRR at 60 min]



(f) C15 [MRD 62% (strong agreement)] [FRR at 60 min]



(g) C5 [MRD 75% (excellent agreement)] [FRR at 60 min]

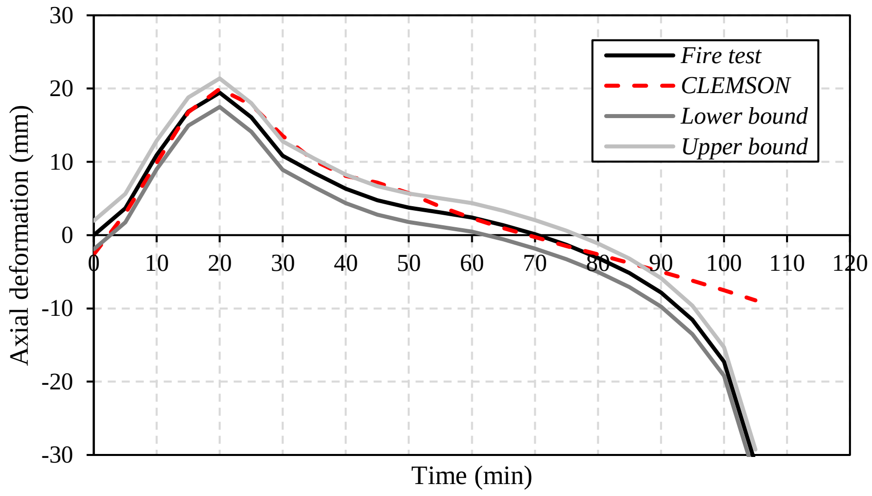


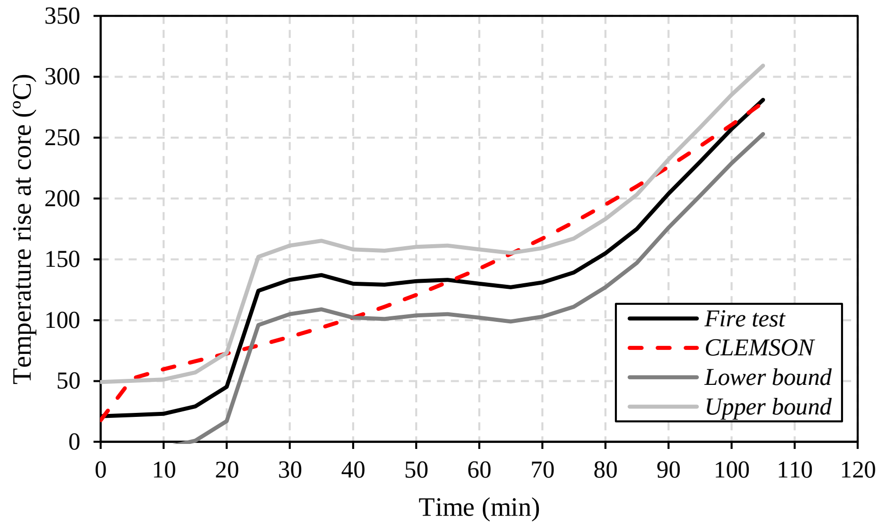
(h) C5 [MRD 100% (excellent agreement)] [FRR at 60 min]

This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

Please cite this paper as:

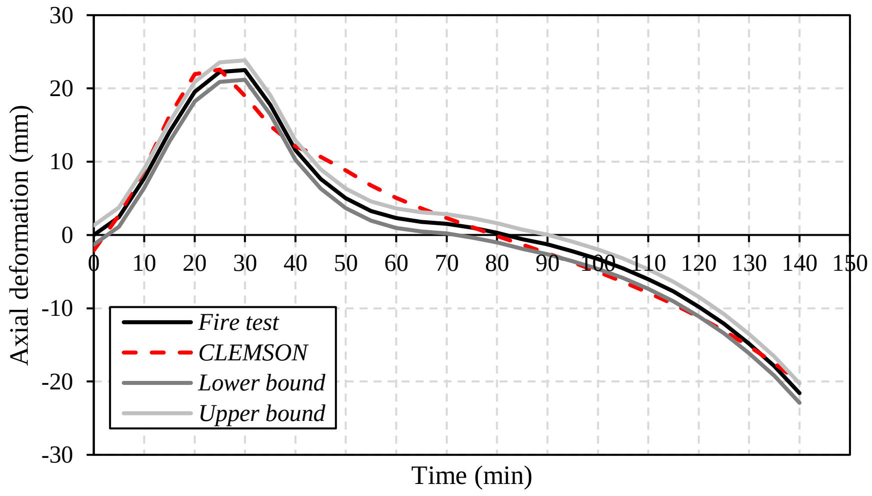
Naser M.Z. (2022). ""CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members." *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.

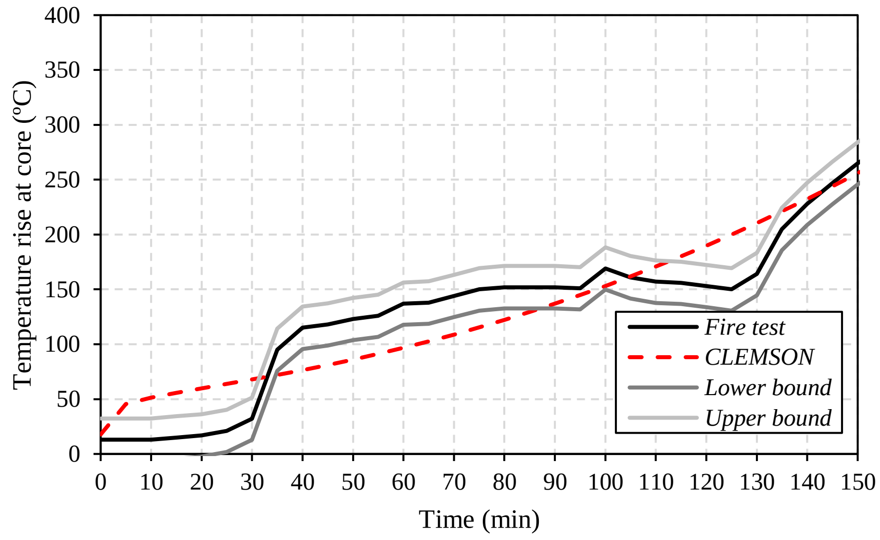




(i) C20 [MRD 55% (strong agreement)] [FRR at 60 min]

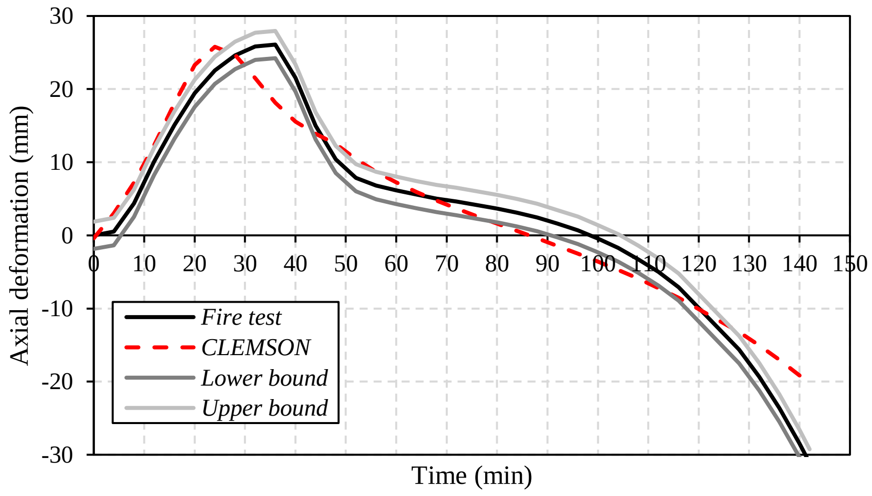
(j) C20 [MRD 45% (fair agreement)] [FRR at 60 min]

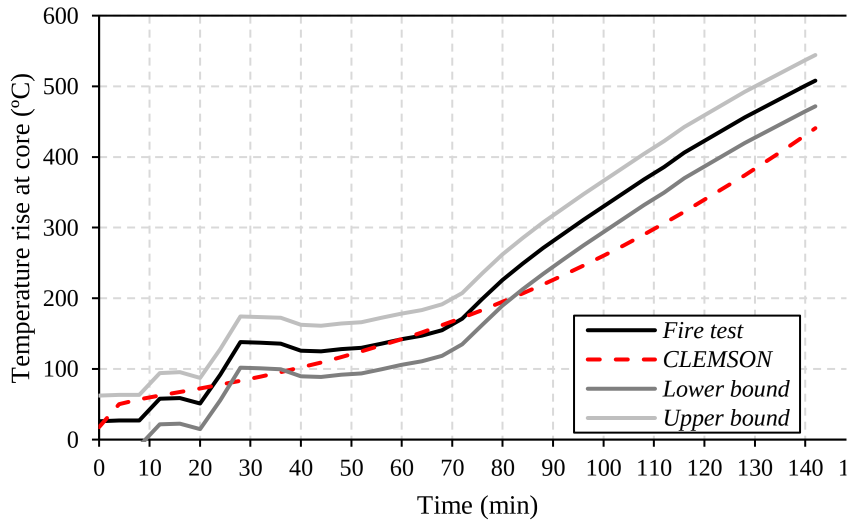




(k) C29 [MRD 40% (fair agreement)] [FRR at 120 min]

(l) C29 [MRD 39% (fair agreement)] [FRR at 120 min]





(m) C23 [MRD 23% (poor) agreement] [FRR at 120 min]

(n) C23 [MRD 54% (fair agreement)] [FRR at 120 min]

Fig. 3 Additional validation of CFST columns response to fire [taken from the works of Lie and Chabot (Lie and Chabot 1992)]

Insights into Ensemble's Explainability

Once the developed ensemble is deemed acceptable, as noted in the previous section, then the ensemble is further examined herein. Special regard is given to the explainability of the ensemble by means of exploring feature importance and partial dependence plots of key features governing FRR, as well as those associated with reconstructing temperature-time, and deformation-time history of CFST columns.

Feature Importance

In a ML analysis, a model (or ensemble) consists of a number of features, each of which is expected to make a unique contribution towards the outcome of model prediction (Altmann et al. 2010). Simply, feature importance presents the extent to which its features influence predictions from the developed ensemble. Such importance can be measured by evaluating the increase of a model's prediction error after systematically permuting all of its features (Altmann et al. 2010). In this evaluation, a feature may score a high value if permuting its values increases the model error – thereby deeming such feature as "important" and vice versa. Hence, by understanding the influence of each feature, one can interpret model's predictions.

As one can see from Fig. 4a, the ensemble developed to predict FRR of CFST columns is dependent upon seven features, namely P, f_c, D, f_y, k, n, t, r, and e, which they scored: 100, 92, 90, 82, 74, 60, 40, 29, and 12%, respectively. One should note that the same figure also shows some variation in the importance of each feature importance as compared by the ensemble and other ML models. Despite such variation, notably that faced by the RF model, there seems to be an overall agreement in the magnitude of feature importance values; therefore, implying consistency across the different models. This also shows the merit in exploring a series of algorithms as opposed to favoring a sole algorithm.

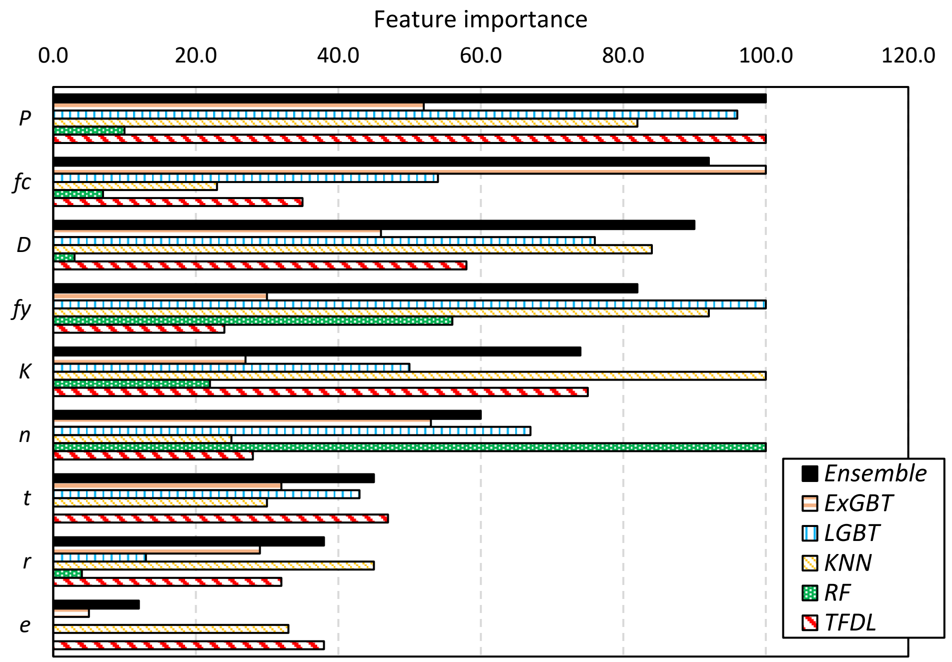
Figure 4b also shows importance scores for features responsible for reconstructing temperature-time history of CFST columns. The thermal response of CFST columns is seen to be primarily governed by the exposure time to fire, diameter, and thickness of columns (in this order), with other features having minimal impact. This analysis meshes with the fact that the thermal response of CFST columns is indeed a function of fire exposure time and size of the column, as noted by (Han et al. 2009; Kodur and Naser 2020; Lu et al. 2009). Since all selected columns were exposed to standard fires, made of hot-rolled tubes and plain normal strength concrete, then the effect of concrete type, boundary conditions, etc., can be normalized. However, the reader is to note that the same methodology could be extended to other structural elements exposed to other heating conditions.

On the other hand, Fig. 4c displays feature importance for deformation-time history of CFST columns. Unlike the relatively smaller number of features governing the thermal response of columns, the deformation-time history is seen to be governed by exposure time, P, D, t, and k. these additional features reflect the naturally complex nature of the mechanical response of CFST columns under fire conditions (Han et al. 2013; Wan et al. 2017). The feature importance analysis also shows consistency among the ensemble and other algorithms.

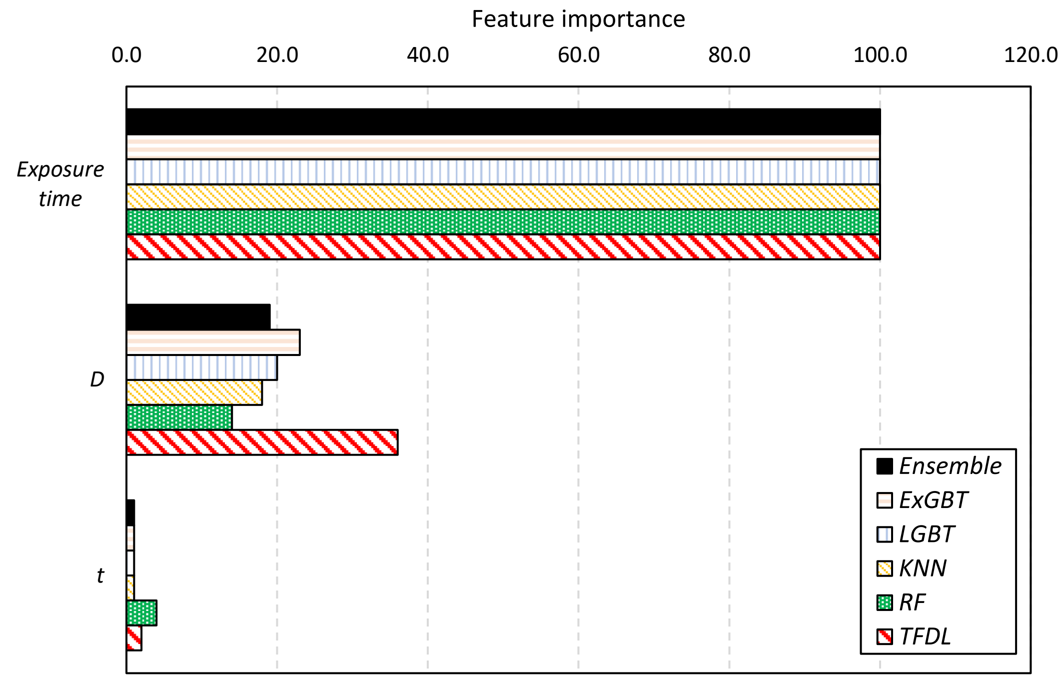
This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

Please cite this paper as:

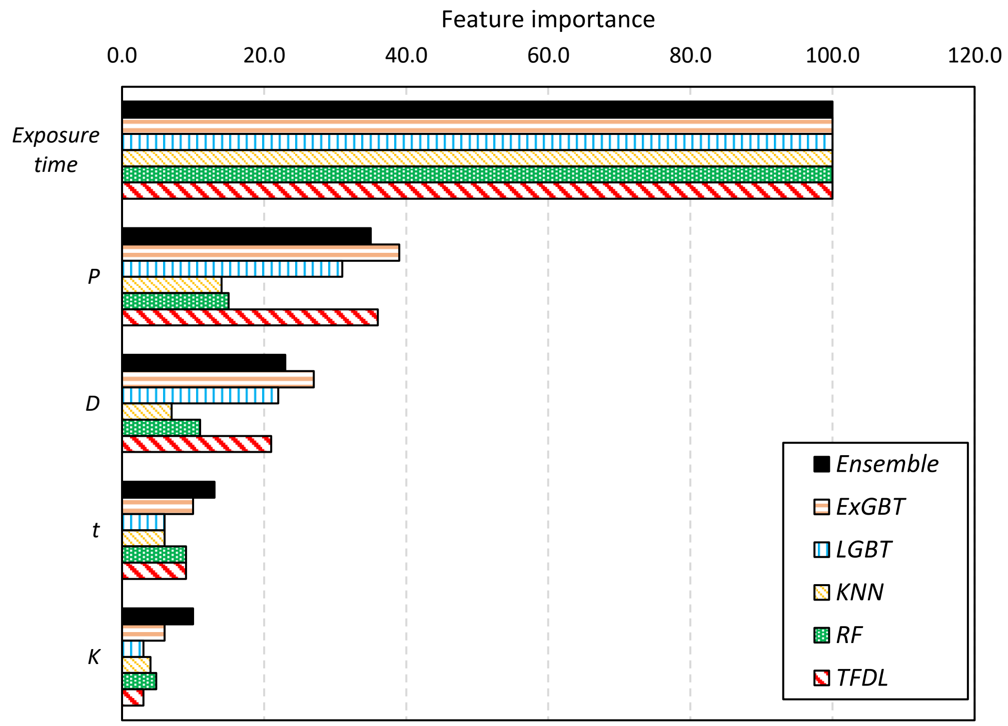
Naser M.Z. (2022). ""CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members." *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.



(a) Fire resistance rating (FRR)



(b) Temperature-time history of CFST columns



(c) Deformation-time history of CFST columns

444 Fig. 4 Insights into feature importance

This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

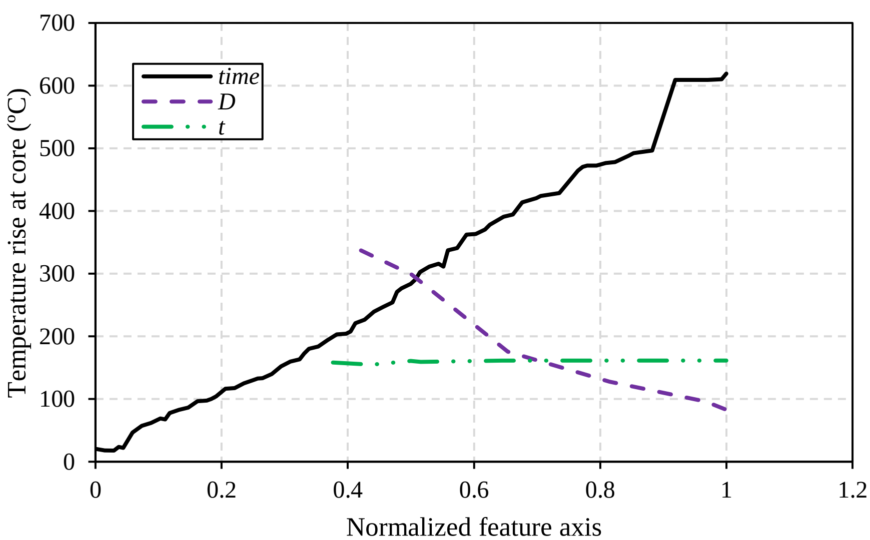
Please cite this paper as:

Naser M.Z. (2022). ""CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members." *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.

Partial Dependence Plots (PDP)

The partial dependence plot (PDP) is another tool that can shed insights into a ML model's explainability. A PDP portrays an individual feature's marginal effect on model predictions while holding the other features constant (Friedman 2001). The outcome of a PDP can also be used to reveal the type of relationship a feature has on model predictions (e.g., linear, nonlinear, etc.). A PDP helps determine the transition in a model's predictive performance to the change in the feature(s) (Friedman 2001). Figure 5 shows PDPs for all features responsible for reconstructing temperature-time and deformation-time history of CFST columns (Scikit 2021)).

As expected, Fig. 5a shows a positive linear relationship between exposure time and temperature rise in CFST columns. The same figure also shows how increasing tube diameter size tends to lead to a reduction in temperature rise at the core of CFST columns. This is in response to the fact that bigger columns can hold a larger mass of concrete. Such columns tend to require higher thermal energy to increase core temperature in response to the high thermal capacity of concrete. Acquiring higher thermal energy is positively tied to a more prolonged fire exposure. A clear transition occurs at 65% of the normalized diameter size (which corresponds to a diameter of 219 mm). This cut-off point shows that temperature rise at the core seems to stagger in columns of diameters larger than 219 mm. Finally, given the small thickness of the tube, the partial dependence of this feature seems to have a minor effect on temperature rise in CFST columns. A look into this PDP shows a good match between the ensemble's rationale and guiding physics principles.

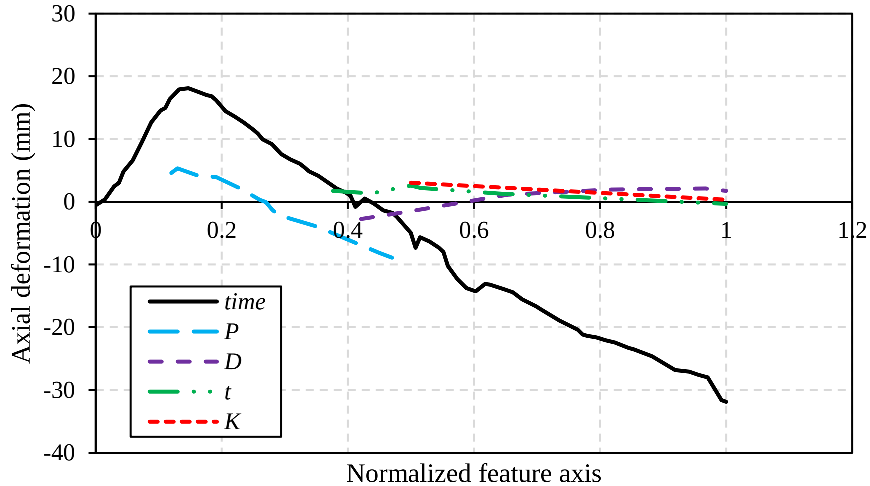


(a) Temperature-time history

This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

Please cite this paper as:

Naser M.Z. (2022). ""CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members." *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.



(b) Deformation-time history

Fig. 5 Insights from partial dependence plots

On the deformation front, Fig. 5b shows that the axial deformation of plain CFST columns is a function of exposure time, *P*, *D*, *t*, and *k* (as noted in the previous section). In addition, this figure depicts the complex relationship between exposure time and axial deformation. Up to about 40% of a typical column's exposure under standard fires is associated with positive deformation (i.e., expansion), with a peak taking place within the first 10%-20% range of the exposure, followed by a contraction stage that accelerates failure. This also means that once a CFST reaches its maximum expansion, one can confidently estimate the failure point of such a column. In addition, smaller magnitudes of loading are also associated with an expansion of columns, while larger load levels tend to induce faster contraction of columns under fire in response to accelerated creep effects (Kodur et al. 2020). The influence of tube diameter and thickness, as well as boundary conditions, is steady and not as influenceable as the aforenoted features.

Ensemble's Graphical User Interface (GUI)

Finally, the developed ensemble was integrated into a spreadsheet (tool) to be used through the Excel program. This tool operates by inputting the "inputs" variables, which then are used to evaluate FRR, and corresponding thermal and mechanical response histories. This spreadsheet is presented in Fig. 6 and can provide users with a means to deploy this ensemble. Given that the developed ensemble does not require carrying out a thermo-mechanical coupled analysis, meshing or otherwise, one can appreciate the attractiveness of the developed ensemble, especially when compared to other methods of analysis such as those based on the finite difference (FD) or FE methods, or those of iterative nature. For completeness, the developed spreadsheet will be attached to this publication and can be freely downloaded from this research group's website.

This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

Please cite this paper as:

Naser M.Z. (2022). ""CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members." *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.

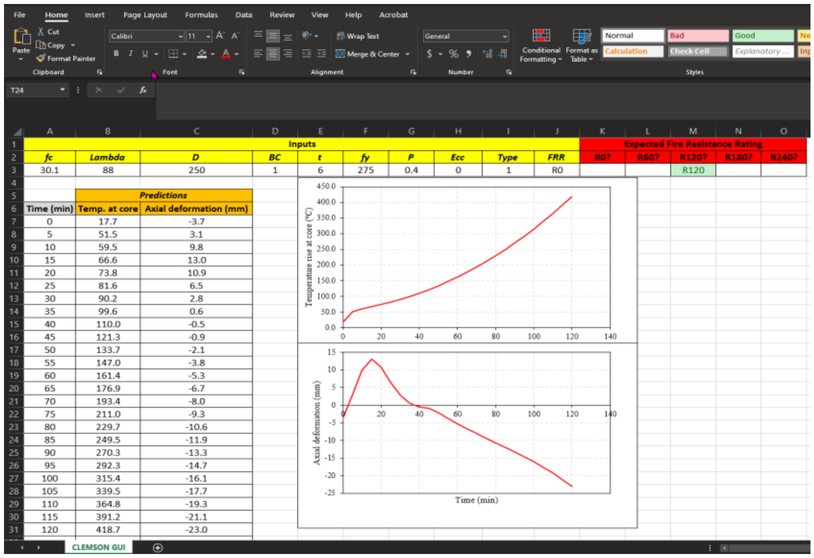


Fig. 6 GUI of the developed spreadsheet

Conclusions

This paper presents CLEMSON – an AutoML virtual assistant that enables engineers to carry out ac**C**e**LE**rated, si**M**ulation-free, tran**S**parent, reduced-**O**rder, and infere**N**ce-based fire resistance analysis. CLEMSON leverages competitive, and ensemble ML algorithm search and is supplemented with explainability measures, as well as GUI capabilities. For the purpose of this work, CLEMSON is applied to evaluate the fire resistance rating, temperature-time, and deformation-time history of CFST. The results of this analysis infer the suitability and applicability of CLEMSON as cross-checked against observations from real fire tests and traditional and new functional performance metrics. The following list of inferences can also be drawn from the findings of this study:

- AutoML methods, when properly applied, present a new opportunity that facilitates the acceptance and widespread of ML into the structural fire engineering domain.
- This study shows the merit of adopting multi-algorithm/multi-metric ML analysis.
- Utilizing ensemble learning is shown to yield favorably improved performance as compared to individual ML models.
- Applying the newly derived functional performance metrics can aid in adding a new layer of verification to ML predictions.
- Fire resistance of CFST columns is seen to be governed by exposure time, load level, tube diameter, and thickness, as well as boundary conditions.
- Under standard fires, up to about 40% of a typical CFST column's is associated with expansion, with a peak taking place within the first 10%-20% range of the exposure, followed by a contraction stage towards failure.

This is a preprint draft. The published article can be found at: https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399

Please cite this paper as:

Naser M.Z. (2022). ""CLEMSON: An Automated Machine Learning (AutoML) Virtual Assistant for Accelerated, Simulation-free, Transparent, Reduced-order and Inference-based Reconstruction of Fire Response of Structural Members." *ASCE Journal of Structural Engineering*. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003399.

Acknowledgment

The author would like to thank the Editor and Reviewers for their support of this work and constructive comments that enhanced the quality of this manuscript.

Data Availability

Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

Conflict of Interest

The author declares no conflict of interest.

References

Abadi, M. (2016). "TensorFlow: learning functions at scale." *ACM SIGPLAN Notices*.

Abdollahi-Arpanahi, R., Gianola, D., and Peñagaricano, F. (2020). "Deep learning versus parametric and ensemble methods for genomic prediction of complex phenotypes." *Genetics Selection Evolution*.

ACI. (2021). "The Concrete Industry in the Era of Artificial Intelligence." <https://www.concrete.org/news/newsdetail.aspx?f=51732604> (May 18, 2021).

Altmann, A., Toloşi, L., Sander, O., and Lengauer, T. (2010). "Permutation importance: A corrected feature importance measure." *Bioinformatics*.

Bishop, C. (2007). "Pattern Recognition and Machine Learning." *Technometrics*.

Boehmke, B., Greenwell, B., Boehmke, B., and Greenwell, B. (2020). "Interpretable Machine Learning." *Hands-On Machine Learning with R*.

Botchkarev, A. (2019). "A new typology design of performance metrics to measure errors in machine learning regression algorithms." *Interdisciplinary Journal of Information, Knowledge, and Management*.

Botu, V., and Ramprasad, R. (2015). "Adaptive machine learning framework to accelerate ab initio molecular dynamics." *International Journal of Quantum Chemistry*.

Buchanan, A., and Abu, A. (2017). *Structural design for fire safety*.

Chaudhary, R. K., Van Coile, R., and Gernay, T. (2021). "Potential of Surrogate Modelling for Probabilistic Fire Analysis of Structures." *Fire Technology*, Springer, 1–27.

Chou, J. S., and Pham, A. D. (2013). "Enhanced artificial intelligence for ensemble approach to predicting high performance concrete compressive strength." *Construction and Building Materials*.

Cremonesi, P., Koren, Y., and Turrin, R. (2010). "Performance of Recommender Algorithms on Top-N Recommendation Tasks Categories and Subject Descriptors." *RecSys*.

Cuevas-Zuviría, B., and Pacios, L. F. (2020). "Analytical Model of Electron Density and Its Machine Learning Inference." *Journal of Chemical Information and Modeling*.

Doran, D., Schulz, S., and Besold, T. R. (2018). "What does explainable AI really mean? A new conceptualization of perspectives." *CEUR Workshop Proceedings*.

Dosilovic, F. K., Brcic, M., and Hlupic, N. (2018). "Explainable artificial intelligence: A survey." *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings*.

Dudley, J. J., and Kristensson, P. O. (2018). "A review of user interface design for interactive machine learning." *ACM Transactions on Interactive Intelligent Systems*.

Freund, Y., and Schapire, R. E. (1996). "Experiments with a New Boosting Algorithm." *Proceedings of the 13th International Conference on Machine Learning*.

Freund, Y., and Schapire, R. E. (1997). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting." *Journal of Computer and System Sciences*.

Friedman, J. H. (2001). "Greedy function approximation: A gradient boosting machine." *Annals of Statistics*.

Fu, F. (2020). "Fire induced progressive collapse potential assessment of steel framed buildings using machine learning." *Journal of Constructional Steel Research*.

Ganaie, M. A., Hu, M., Tanveer, M., and Suganthan, P. N. (2021). *Ensemble deep learning: A review*.

García-Martín, E., Rodrigues, C. F., Riley, G., and Grahn, H. (2019). "Estimation of energy consumption in machine learning." *Journal of Parallel and Distributed Computing*.

Hamori, S., Kawai, M., Kume, T., Murakami, Y., and Watanabe, C. (2018). "Ensemble Learning or Deep Learning? Application to Default Risk Analysis." *Journal of Risk and Financial Management*.

Han, J., Dong, Y., and Xu, Z. (2009). "Study on fire-resistance performance of reinforced concrete filled steel tubular columns." *Jianzhu Jiegou Xuebao/Journal of Building Structures*.

Han, L.-H., Zhao, X.-L., Yang, Y.-F., and Feng, J.-B. (2003). "Experimental Study and Calculation of Fire Resistance of Concrete-Filled Hollow Steel Columns." *Journal of Structural Engineering*.

Han, L. H., Chen, F., Liao, F. Y., Tao, Z., and Uy, B. (2013). "Fire performance of concrete filled stainless steel tubular columns." *Engineering Structures*.

Hastie, T., Tibshirani, R., and Friedman, J. (2011). "The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)

(9780387848570): Trevor Hastie, Robert Tibshirani, Jerome Friedman: Books." *The elements of statistical learning: dta mining, inference, and prediction*.

Hindman, M. (2015). "Building Better Models: Prediction, Replication, and Machine Learning in the Social Sciences." *Annals of the American Academy of Political and Social Science*.

Jackson, R. (2019). *A roadmap to reducing greenhouse gas emissions 50 percent by 2030*.

Jiang, L., Jiang, Y., Zhang, Z., and Usmani, A. (2021). "Thermal Analysis Infrastructure in OpenSees for Fire and its Smart Application Interface Towards Natural Fire Modelling." *Fire Technology*.

Keras. (2020). "GitHub - keras-team/keras: Deep Learning for humans." <https://github.com/keras-team/keras> (Feb. 9, 2021).

Ketkar, N., and Ketkar, N. (2017). "Introduction to Keras." *Deep Learning with Python*.

Kodur, V., Alogla, S. M., and Venkatachari, S. (2020). "Guidance for Treatment of High-Temperature Creep in Fire Resistance Analysis of Concrete Structures." *Fire Technology*.

Kodur, V., and Naser, M. Z. (2020). *Structural fire engineering*. McGraw Hill Professional.

Laszczyk, M., and Myszkowski, P. B. (2019). "Survey of quality measures for multi-objective optimization: Construction of complementary set of multi-objective quality measures." *Swarm and Evolutionary Computation*, Elsevier, 48, 109–133.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018). "Visualizing the loss landscape of neural nets." *Advances in Neural Information Processing Systems*.

Liaw, A., and Wiener, M. (2002). *Classification and Regression by RandomForest*.

Lie, T. T., and Chabot, M. (1992). "Experimental Studies on the Fire Resistance of Hollow Steel Columns Filled with Plain Concrete." *National Research Council Canada*.

LightGBM. (2020). "Welcome to LightGBM's documentation! — LightGBM 3.1.1.99 documentation." <https://lightgbm.readthedocs.io/en/latest/> (Feb. 9, 2021).

Litman, T. (2014). "Autonomous Vehicle Implementation Predictions: Implications for Transport Planning." *Transportation Research Board Annual Meeting*.

Lu, H., Zhao, X. L., and Han, L. H. (2009). "Fire behaviour of high strength self-consolidating concrete filled steel tubular stub columns." *Journal of Constructional Steel Research*.

McNamee, M., Meacham, B., van Hees, P., Bisby, L., Chow, W. K., Coppalle, A., Dobashi, R., Dlugogorski, B., Fahy, R., Fleischmann, C., Floyd, J., Galea, E. R., Gollner, M., Hakkarainen, T., Hamins, A., Hu, L., Johnson, P., Karlsson, B., Merci, B., Ohmiya, Y., Rein, G., Trouvé, A., Wang, Y., and Weckman, B. (2019). "IAFSS agenda 2030 for a fire safe world." *Fire Safety Journal*.

Naser, M. Z. (2019a). "AI-based cognitive framework for evaluating response of concrete structures in extreme conditions." *Engineering Applications of Artificial Intelligence*, 81.

Naser, M. Z. (2019b). "Fire Resistance Evaluation through Artificial Intelligence - A Case for Timber Structures." *Fire Safety Journal*, 105, 1–18.

Naser, M. Z. (2021a). "Mechanistically Informed Machine Learning and Artificial Intelligence in Fire Engineering and Sciences." *Fire Technology*, Springer, 1–44.

Naser, M. Z. (2021b). "An engineer's guide to eXplainable Artificial Intelligence and Interpretable Machine Learning: Navigating causality, forced goodness, and the false perception of inference." *Automation in Construction*, Elsevier, 129, 103821.

Naser, M. Z., and Alavi, A. (2020). "Insights into Performance Fitness and Error Metrics for Machine Learning."

Ouache, R., Nahiduzzaman, K. M., Hewage, K., and Sadiq, R. (2021). "Performance Investigation of Fire Protection and Intervention Strategies: Artificial Neural Network-based Assessment Framework." *Journal of Building Engineering*.

Panev, Y., Kotsovinos, P., Deeny, S., and Flint, G. (2021). "The Use of Machine Learning for the Prediction of fire Resistance of Composite Shallow Floor Systems." *Fire Technology*.

Peskova, K., and Neruda, R. (2019). "Hyperparameters search methods for machine learning linear workflows." *Proceedings - 18th IEEE International Conference on Machine Learning and Applications, ICMLA 2019*.

Polikar, R. (2009). "Ensemble learning." *Scholarpedia*.

Schmidhuber, J. (2015). "Deep learning in neural networks: An overview." *Neural Networks*, Springer, 61, 85–117.

Schmidt, M. D., and Lipson, H. (2010). "Age-fitness pareto optimization."

Scikit. (2020a). "sklearn.ensemble.GradientBoostingRegressor — scikit-learn 0.24.1 documentation." <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html> (Feb. 9, 2021).

Scikit. (2020b). "sklearn.ensemble.RandomForestClassifier — scikit-learn 0.24.1 documentation." <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html> (Feb. 9, 2021).

Scikit. (2021). "4.1. Partial Dependence and Individual Conditional Expectation plots — scikit-learn 0.24.1 documentation." <https://scikit-learn.org/stable/modules/partial_dependence.html> (Feb. 20, 2021).

Strubell, E., Ganesh, A., and McCallum, A. (2020). "Energy and Policy Considerations for Modern Deep Learning Research." *Proceedings of the AAAI Conference on Artificial Intelligence*.

Su, L., Wu, X., Zhang, X., and Huang, X. (2021). "Smart performance-based design for building fire safety: Prediction of smoke motion via AI." *Journal of Building Engineering*, Elsevier, 43, 102529.

Tarassoli, S. P. (2019). "Artificial intelligence, regenerative surgery, robotics? What is realistic for the future of surgery?" *Annals of Medicine and Surgery*.

TensorFlow. (2020). "GitHub - tensorflow/tensorflow: An Open Source Machine Learning Framework for Everyone." <https://github.com/tensorflow/tensorflow> (Feb. 9, 2021).

Wan, C. Y., Zha, X. X., and Dassekpo, J. B. M. (2017). "Analysis of axially loaded concrete filled circular hollow double steel tubular columns exposed to fire." *Fire Safety Journal*.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. *Data Mining: Practical Machine Learning Tools and Techniques*.

Wolpert, D. H., and Macready, W. G. (1997). "No free lunch theorems for optimization." *IEEE Transactions on Evolutionary Computation*.

Wu, C. J., Brooks, D., Chen, K., Chen, D., Choudhury, S., Dukhan, M., Hazelwood, K., Isaac, E., Jia, Y., Jia, B., Leyvand, T., Lu, H., Lu, Y., Qiao, L., Reagen, B., Spisak, J., Sun, F., Tulloch, A., Vajda, P., Wang, X., Wang, Y., Wasti, B., Wu, Y., Xian, R., Yoo, S., and Zhang, P. (2019). "Machine learning at facebook: Understanding inference at the edge." *Proceedings - 25th IEEE International Symposium on High Performance Computer Architecture, HPCA 2019*.

Wu, X., Zhang, X., Huang, X., Xiao, F., and Usmani, A. (2021). "A real-time forecast of tunnel fire based on numerical database and artificial intelligence." *Building Simulation*.

XGBoost Python Package. (2020). "Python Package Introduction — xgboost 1.4.0-SNAPSHOT documentation." <https://xgboost.readthedocs.io/en/latest/python/python_intro.html#early-stopping> (Feb. 10, 2021).