INTERNSHIP REPORT

FRAMEWORK FOR INVESTIGATING MACHINE LEARNING

MODEL RESPONSE TO INPUT PARAMETERS

MR. MANSOOR

PROGRAM: MASTER OF ENGINEERING IN ELECTRICAL & COMPUTER ENGINEERING

20ST DECEMBER, 2023

INTERNSHIP INFORMATION

Student name
Student ID
Program
Title
Internship period
Company
Company address:
Company Supervisor
TGGS Advisors

APPROVAL REPORT

Supervisor's signature	……..……..……..... (Professor Dr.) Date…
TGGS Advisor's signature	…….…… (Assistant Professor Dr.) Date…

Table of Contents

LIST OF FIGURE

1. INTRODUCTION

The term closely linked to the investigation of a model's response to input parameters is known as sensitivity analysis. This section provides a concise overview of the application and challenges associated with sensitivity analysis in the development and validation of computational models that replicate physical systems or scientific theories.

Sensitivity analysis plays role in mathematical modeling by pinpointing deficiencies and facilitating model calibration and verification. In complex systems prone to the 'uncertainty cascade effect,' sensitivity analysis identifies optimal points where uncertainty is minimized or maximized [1]. It also serves as a diagnostic tool in model validation and verification, ranking input features and revealing their interactions and dependencies [2]. Furthermore, sensitivity analysis is integral to data-driven machine learning and deep learning models, particularly in addressing transparency and explainability challenges [3]. It identifies the influence of individual or combined input parameters on model responses, enhancing the understanding of these complex models.

Sensitivity analysis techniques can be categorized mathematically as either deterministic or stochastic [4]. Deterministic techniques rely on real-world data to obtain input values and assess model response around a certain reference point or base case with few input parameters. On the other hand, stochastic techniques take into consideration the combined effect of all the input parameters and utilize random distribution data for model analysis. The current work adopts deterministic sensitivity analysis techniques to build and demonstrate the current framework.

Sensitivity analysis faces some common challenges due to similar characteristics of engineering models and their environment. Firstly, these systems have high input dimensionality space causing curse of dimensionality. Consequently, the computational cost of sensitivity analysis algorithms, which performs sampling or assess the interaction effect of input parameters, rises exponentially [6]. Secondly, engineering models are usually complex and computationally intensive which sometimes result in sensitivity analysis techniques based on surrogate models [7]. These secondary models are generally less complex offering better explainability and interpretability but require training and limitations.

2. SYSTEM FRAMEWORK

The current system framework, Figure 1, is designed to investigate a computational model response to input parameters using a broad range of sensitivity analysis techniques. The system is divided into three main components; data, model, and techniques. They operate almost independently allowing flexibility and scalability in the framework for the addition of new techniques and models. This section provides some functional and structural characteristic of the framework while avoiding any implementation details.

Current framework supports computational model, trainable or non-trainable, if it can be converted to the onnx format. The choice of onnx format is motivated by its portability and widespread support across various machine learning and deep learning frameworks and libraries [8]. This format ensures platform independence for the model and facilitates further experimentation through a common interface with any given onnx model.

While data may be used for parameter tuning of computational model (before converting them into onnx version) it can also be used during sensitivity analysis. In this framework, data is kept raw and each sensitivity analysis techniques draw statistics or samples from data with respect to their requirements. These techniques may use a different data form training one if it fulfill the required parameters, hence, synthetic data can be generated and integrated into the system.

The core module, sensitivity analysis, follows a similar structure for each technique. The closest possible implementation for each sensitivity analysis technique is shown in Figure 2. Using this template, the sensitivity techniques can be experimented with addition models, data, or sensitivity measure. New sensitivity techniques, following similar structure, can be easily integrated into the system. It can be inferred form Figure # that object-oriented design is adopted to implement each module but inheritance is avoided between modules to make the system flexible and scalable.

Evaluating model sensitivity can be tricky since there is no single evaluation metric or criterion for analyzing the model's response to each input parameter. Each sensitivity technique has a self-defined quantity or measure, whether scalar or vector, that offers relative or absolute information on the influence of each input parameter on the model's behavior.

Finally, to present the results of sensitivity technique, sensitivity measure, different visual representations, i.e. bar chart, have been placed inside a visual analytical module that possess parameters for customizing the plots and saving the results.

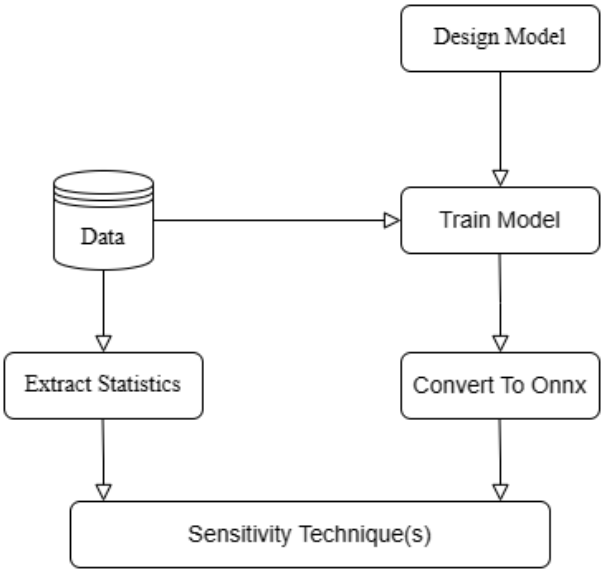

Figure 1: System Design

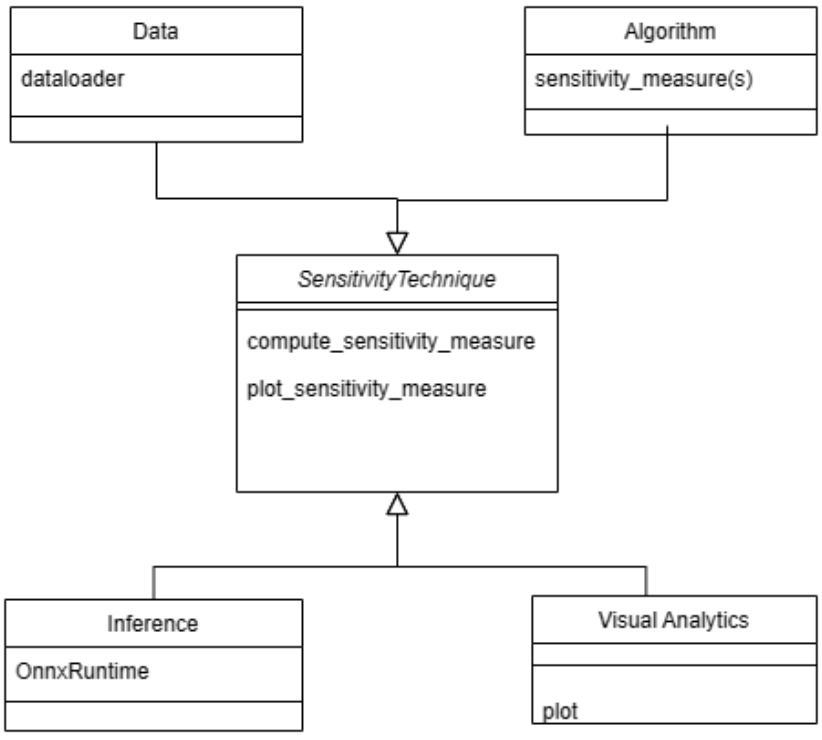

Figure 2: Generic Class Diagram for Sensitivity Techniques

3. DATA SOURCE

The dataset, California housing dataset, originate from 1990 U.S censes and contains demographic and housing information of each district in California [9]. Since, information is granular to district level therefore some statistics are only an average or median. Furthermore, dataset is open source and can be fetched using scikit library.

The dataset possess 20,640 instances with no missing values and numerical values as floating number. The target value is the median price of house in each district expressed in hundreds of thousands of dollars. Furthermore, after subsampling, few median house values are plotted with respect to latitude and longitude as scatter plot in Figure 3.

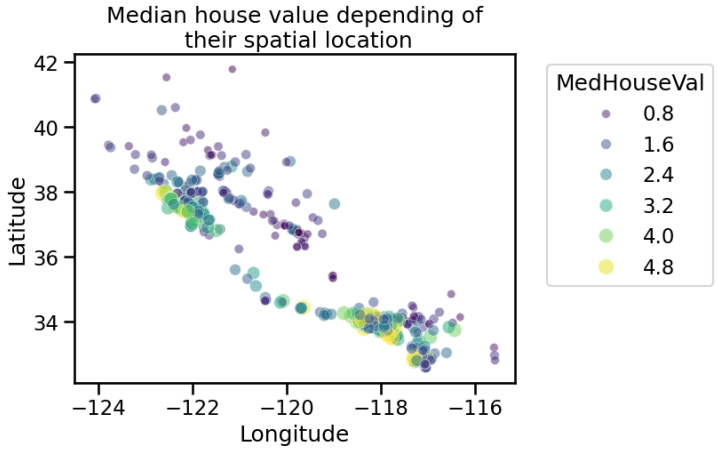

Figure 3: Median House Value w.r.t Spatial Location

4. CUSTOM MODEL

A custom model is designed to solve the regression problem of California housing price prediction. The model is a nonlinear neural network with two fully connected hidden layers and a single output node making it a scalar-value function $\mathbb{f} : \mathbb{R}^n \rightarrow \mathbb{R}$. The input layer has n nodes and output layer has single node whereas the hidden layers have 64 and 32 nodes respectively with activation function ReLU. Loss function is mean absolute error and evaluation metric is accuracy.

Model is trained using all 20,640 instances and validated using cross-validation. The batch size is kept 100 and model parameters are updated after every iteration using Adam optimizer. The trained model parameters are exported using torch library into .pth file extension.

Later, the torch model file (.pth) is converted to open neural network exchange model file format (.onnx). The onnx model is accessed through onnx library [8] which creates inference session and perform runtime model predictions on input data. This runtime session is used to investigate the model response to input parameter based on the given sensitivity analysis technique or model validation strategy.

5. APPLIED TECHNIQUES

To demonstrate the framework, some deterministic sensitivity analysis techniques are applied on the above mentioned model using the onnx runtime. Below are some methods with their algorithms, computational complexity, and results. Each technique is studied exclusive with its own merits and constraints, circumventing comparison or conclusion.

5.1. BOUNDARY SENSITIVITIES

Boundary sensitivities, a subset of One-Factor-At-a-Time sensitivities, are the most basic and simple sensitivity method to implement and analyze. It treats the model as a black box and a scalar-value function. The model $f: R^n \rightarrow R$ is explored around a base case or reference case, x^0 such that $x^0 = (x_1^0,\ x_2^0, \dots, x_n^0)$ which in this example is the mean values of each input parameter obtained from training data. The two sensitivity cases in this technique are x^+ (maxima) and x^- (minima) where $x^+ = (x_1^+,\ x_2^+, \dots, x_n^+)$ and $x^- = (x_1^-,\ x_2^-, \dots, x_n^-)$. These extrema are the upper and lower limits of each input parameter obtained from the available training data. However, local sensitivity is performed by perturbing one input at a time to its extrema with respect to the sensitivity case. Hence, all input values remain at their base case except the input, i, which is under observation. For instance, consider the case of maxima for input parameter i, then $x = \left(x_1^0, x_2^0, \dots, x_{i-1}^0,\ x_i^+, x_{i+1}^0, \dots, x_n^0\right) = \left(x_i^+, x_{\sim i}^0\right)$ and the output obtained is $y_i^+ = f\left(x_i^+, x_{\sim i}^0\right)$. The output variation calculated as sensitivity measure is $\Delta_i^+ y = y_i^+ - y^0$ where $y^0 = f(x^0)$. Same notational and formula is adopted in case of minima except that x_i is shifted to x_i^- and the output sensitivity measure is $\Delta_i^- y = y_i^- - y^0$. Figure 5 presents the algorithmic implementation of the obtained boundary sensitivities.

The most suitable visual diagram to analyze the boundary sensitivities is the tornado diagram [10]. The boundary sensitivity measure appears as horizontal bars in a tornado diagram and can be called upper and lower tornado sensitivity measure. The y-axis labels the input parameters while the x-axis label the value of sensitivity measure. The upper and lower boundary sensitivity measures calculated using the above mentioned model and California housing data can be seen below in Figure 6 & 7.

Since there is no computational involved in preparing the use cases, x_i^+ and x_i^-, therefore the computational cost is equivalent to the number of model evaluations. Henceforth, for n input parameters the cost of computing boundary sensitivities is $2n + 1$. Furthermore, batch computation is achieved by storing the use cases in a matrix and using it a placeholder for input data to model.

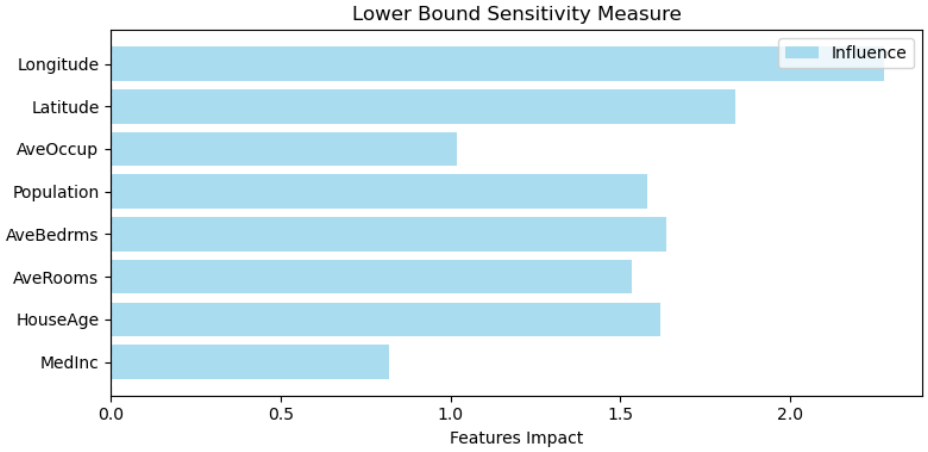

Figure 4: Results of Upper Bound Sensitivity Measure

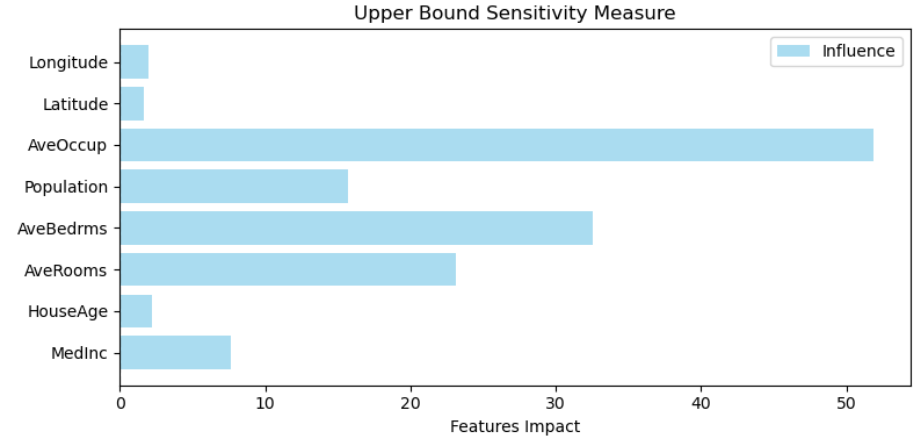

Figure 5: Results of Lower Bound Sensitivity Measure

ALGORITHM 1: PSEUDOCODE FOR BOUNDARY SENSITIVITIES

Input: $X \leftarrow data$, $f \leftarrow model$

Output: Δy^- , Δy^+

Procedure:

1 calculate $x^0 \leftarrow mean(X)$

2 compute $y^0 \leftarrow f(x^0)$

3 n ← number of input parameters

4 ***for*** $i \leftarrow 1$ ***to*** n **do**

5 find $x_i^+ \leftarrow max(X_i)$

6 compute $y_i^+ \leftarrow f(x_i^+, x_{\sim i}^0)$

7 calculate $\Delta y_i^+ \leftarrow (\, y_i^+ - y^0 \,)$

8 find $x_i^- \leftarrow min(X_i)$

9 compute $y_i^- \leftarrow f(x_i^-, x_{\sim i}^0)$

10 calculate $\Delta y_i^- \leftarrow (\, y_i^- - y^0)$

11 ***end***

12 *return* Δy^- , Δy^+

Figure 6: Pseudocode for Boundary Sensitivities

The boundary sensitivities together with tornado diagram have certain benefits and limitation. The sensitivity measure and plots represent both magnitude and direction of change caused by perturbing any single input at a time. Secondly, tornado diagram is able to present the input parameters influence in descending order with respect to magnitude; ranking input parameters.

However, boundary sensitivities don't present the complete picture. For example, the influence of input parameter, i, is unknown for any value other than extrema. Secondly, tornado sensitivity measures are not symmetric and should only be applied to monotonic function which is generally not the case. Third, tornado sensitivity measure never measures the interaction effect of input parameters against a model response. Thus, boundary sensitivities with their graphical representation as tornado diagram could be used at investigate a very basic model.

1.1. ONE-WAY SENSITIVITY FUNCTION

One-way sensitivity function [4] is another local sensitivity analysis technique and a subset of One-Factor-At-a-Time sensitivities. For model $f: R^n \rightarrow R$ there exist n input parameters and equivalent number of one-way sensitivity functions. Each one-way sensitivity function $h_i : X_i \rightarrow R$ performs one-to-one mapping of input parameter i. Here, $X_i \subset Z_i$ where X_i is the derived domain of parameter i from the actual domain Z_i. A more accurate formulation of this function is $h_i(x_i) = f(x_i, x_{\sim i}^0)$ where the base case or reference case $x^0 = (x_1^0, x_2^0, \dots, x_n^0)$. In this example, x^0 is the mean value of each input parameter obtained from training data. The predetermined range of input parameter spans from minimum to maximum value of each parameter such that $x_i : \{x_i^1 = x_i^-, x_i^2, x_i^3, \dots x_i^N = x_i^+\}$. Later, the one-way sensitivity measure $h_i^*(x_i) = h_i(x_i) - f(x^0)$, is calculated for each input parameter i. Table # shows the algorithmic implementation of this method.

The most suitable graphical representation of one-way sensitivity function is spider plot [10]. Furthermore, to make the plots smooth, interpolating techniques may be applied on the model output against each parameter. The x-axis labels the uniform distribution of values from 0 to 1 while the y-axis labels the one-way sensitivity measures. The number of plots is equivalent to the number of input parameters. The resulting sensitivities of the current model against the California housing data can be viewed through spider plot in Figure #.

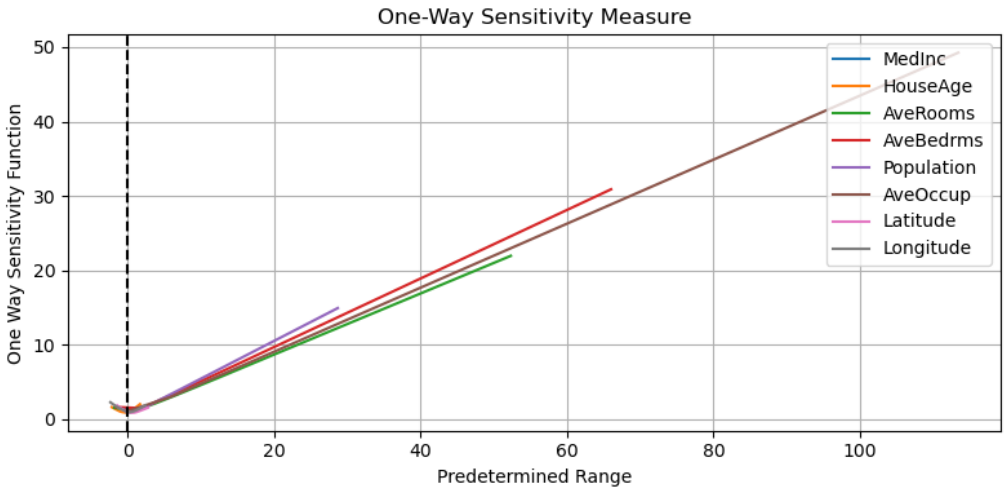

Figure 7: Results of One-Way Sensitivity Function

ALGORITHM 2: PSEUDOCODE FOR ONE-WAY SENSITIVITY FUNCTION

Input: $X \leftarrow data$, $f \leftarrow model$
Output: Δy
Procedure:
calculate $x^0 \leftarrow mean(X)$
compute $y^0 \leftarrow f(x^0)$
n ← number of input parameters
m ← numberof samples
for $i \leftarrow 1$ ***to*** n **do**
    extract samples $x_i : \{ x_i^1 , x_i^2, \dots , x_i^m\}$
    for $j \leftarrow 1$ ***to*** m ***do***
        compute $h_i^j(x_i^j) \leftarrow f(x_i^j,\ x_{\sim i}^0)$
        compute $\Delta y_i^j \leftarrow h_i^j(x_i^j) - f(x^0)$
    end
end
return Δy

Figure 8: Pseudocode for One-Way Sensitivity Function

The computation cost of one-way sensitivity measure is $n * N$ where n is the number of input parameters and N is the number of samples extracted for each input parameter. Thus, if the plot is not smooth or the predetermined range is large $(N \gg n)$ then total cost for calculating the sensitivity measure could be large enough as impractical.

One-way sensitivity function comes with its advantages and constraints. One benefit of its sensitivity measure is its ability to provide insight on direction of change since it spans from minimum to maximum value of each parameter. Secondly, one-way sensitivity function can be extended to multilinear functions that can provide greater insight using linear coefficients but with additional computational cost. Third, boundary sensitivities can be easily derived from one-ways sensitivity measures and plotted on spider plots with vertical bars.

Few risks associated with one-way sensitivity plots is the x-axis scale. Practically, each input parameter has a different unit and henceforth a separate range and spacing. Scaling all input parameters to [0, 1] and extracting samples with equal spacing might not justify the relative influence. Another drawback of one-way sensitivity measure is its inability to find the dependence of inputs parameters on each other, interaction effect.

1.2. SCENARIO DECOMPOSITION

The term 'scenario' has wide meaning and applications based on problem and context. Scenario may be hypothetical or deterministic state of a system that helps predict model behavior in case of uncertainty. The quantitative modeling of model response to given scenario is known as scenario analysis which can be linked to sensitivity analysis by finite change decomposition. In current context, the aim of scenario decomposition is to analyze the interaction of input parameters, up to 2nd order, using predefined scenarios and decomposition strategy.

The model response to a scenario is analyzed with respect to a base case or reference case $x^0 = (x_1^0,\ x_2^0, \dots, x_n^0)$ and the output is calculated as $\Delta y = f(x^1) - f(x^0)$ where x^1 is the scenario. The Δy can be decomposed to any nth order using the below notation [11];

$$\Delta y = \sum_{i=1}^{n} \phi_i + \sum_{i<j} \phi_{i,j} + \phi_{1,2,\dots,n} \quad (1)$$

where

$$\phi_i = f\left(x_i^1\ ;\ x_{\sim i}^0\right) - f(x^0) \quad (2)$$

$$\phi_{i,j} = f\left(x_i^1, x_j^1\ ;\ x_{\sim i,j}^0\right) - \phi_i - \phi_j - f(x^0) \quad (3)$$

$$\phi_{i,j,k} = f\left(x_i^1, x_j^1, x_k^1\ ;\ x_{\sim i,j,k}^0\right) - \phi_{i,j} - \phi_{i,k} - \phi_{j,k} - \phi_i - \phi_j - \phi_k - f(x^0) \quad (4)$$

…

Eq. 1 results in 2^n terms [12] and generates multiple sensitivity indices to understand model behavior. Below are the three main sensitivity indices based on Eq. 2, 3, 4.

First, individual effect or first order finite change sensitivity indices (using Eq. 2) is $\phi_i^1 = \Delta_i y$, where i is the input parameter under observation. Second, the interaction effect, residual interaction or second order finite change sensitivity indices of a parameter i (based on Eq. 3, 2) is $\phi_i^I = \sum_{j=1, i\neq j}^{n} \phi_{i,j}$, where i and j are parameters under investigation. Third, total effect or total finite change sensitivity indices of an input parameter i (based on Eq. 2, 3, 4) is $\phi_i^T = \phi_i + \sum_{j=1}^{n} \phi_{i,j} + \sum_{k=1}^{n}\sum_{j=1}^{n} \phi_{i,j,k} + \cdots + \phi_{1,2,\dots,n}$.

For given example, two scenarios are selected and used to calculate the above sensitivities. These include x^+ (maxima) and x^- (minima) where $x^+ = (x_1^+,\ x_2^+, \dots, x_n^+)$ and $x^- = (x_1^-,\ x_2^-, \dots, x_n^-)$. The change is calculated with reference to base case x^0 which is the mean value of each input parameter obtained from training data. Note; the interaction sensitivity measure is restricted to second order due to computational complexity discussed later. Again, One-Factor-At-a-Time approach is adopted to measure individual sensitivities of input parameter and later their interaction and total effect. However, using matrix notation allows batch processing of

data during model runtime. In addition, a binary mask is generated that possess all the combination of input parameters. Nevertheless, below Table # describe the closest possible algorithmic implementation to compute the three sensitivity measures.

The most suitable representation of the above three sensitivity measures is the generalized tornado diagram [6]. Each sensitivity measure is plotted as a horizontal bar and the y-axis label each input parameter whereas the x-axis labels the values of sensitivity measures. However, the diagram can only plot a single scenario at a given instance for convenient analysis. Figure #, display the three sensitivity measures with interaction effect of second order only.

The computational cost of scenario decomposition is the main constraint in its application. Essentially cost comes from complete finite change decomposition that require $2^n - 1$ model evaluations [12] which is practically not possible to compute if n is even slight greater. However, if order of interaction is reduced, partial decomposition, to k then the number of model evaluations would $\frac{n\,!}{k\,!\,(n-k)\,!}$ [12]. In the current example, partial decomposition is performed and the value of k is set to two. The order of interactions directly affects total sensitivity measure which is the sum of individual and interaction sensitivity measures. Nevertheless, the current implementation uses a binary mask (of all input parameters combination) and sum of its column to identify the type and order of each sensitivity measure.

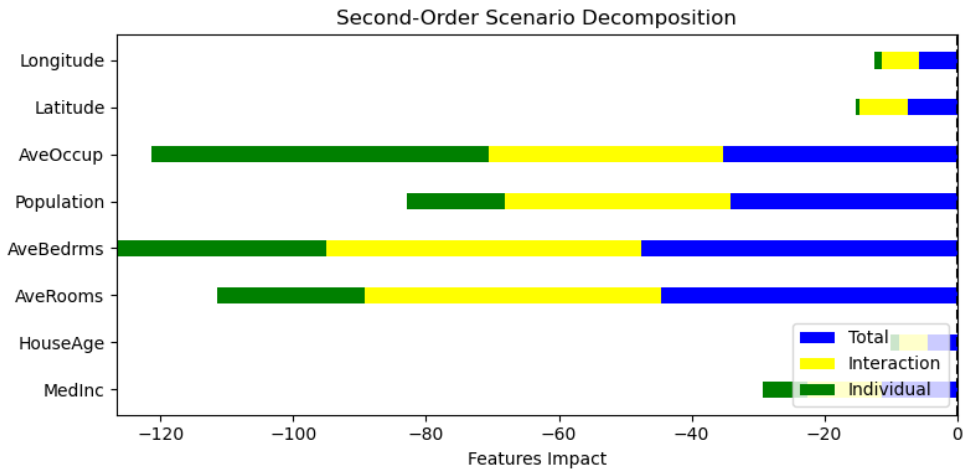

Figure 9: Results of Second Order Scenario Decomposition

ALGORITHM 3: PSEUDOCODE FOR SECOND-ORDER SCENARIO DECOMPOSITION

Input: $X \leftarrow data$, $f \leftarrow model$

Output: ϕ^1 , ϕ^I , ϕ^T

Procedure:

$x^0 \leftarrow mean(X)$

n ← number of input parameters

c ← $\frac{n!}{2! * (n-2)!} + n$

mask ← generate_binary_mask(c)

$x^1 \leftarrow \max(X)$ ***or*** $\min(X)$

w ← $(x^1 * mask) + (x^0 * (1 - mask))$

$y^0 \leftarrow f(x^0)$

for $i \leftarrow 1$ ***to*** c ***do***

  if $sum(mask_i) == 1$ ***then***

    $\phi_i^1 = f(w_i\ ;\ x_{\sim i}^0) - f(x^0)$

  end

  if $sum(mask_i) == 2$ ***then***

    $\phi_i^I \leftarrow f(w_i)$

    $m \leftarrow nonzero\ element\ indices\ (mask_i)$

    for j ***in*** m ***do***

      $\phi_i^I \leftarrow \phi_i^I - \phi_j^1$

    end

  end

end

for $i \leftarrow 1$ ***to*** n ***do***

  $\phi_i^T \leftarrow \phi_i^1 + \phi_i^I$

end

return ϕ^1 , ϕ^I , ϕ^T

Figure 10: Pseudocode for Scenario decomposition

Scenario decomposition has widespread applications due to strong mathematical roots (leading to differential importance measure) and detailed analysis it provides on the influence of input parameters on model output. However, due to computational cost, it is essential to identify the critical scenarios for uncertainty quantification which is a subjective task and may lead to bias or incomplete analysis. [*more]

1.3. FINITE DIFFERENCES

Finite difference is one of basic and fundamental differentiation based sensitivity analysis techniques [13] that belong to class of local sensitivity analysis techniques. The method originates from Taylor series expansion where the change in model response, $y^0 \rightarrow y$ is estimated using the decomposition, Eq. 5, if the model is at least twice differentiable at x^0 .

$$\nabla y = f(x) - f(x^0) =$$

$$\sum_{i=1}^{n} \frac{\partial g(x^0)(x_i - x_i^0)}{\partial x_i} + \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \frac{\partial^2 f(x^0)(x_i - x_i^0)(x_k - x_k^0)}{\partial x_i \partial x_k} + o\left(||(x - x^0)^2||\right) \qquad (5)$$

However, if the second-order term is ignored, assuming infinitesimal change, and model response with respect to single input is consider, that results in Eq.6 below [14].

$$\Delta y = \sum_{i=1}^{n} \frac{\partial f(x^0)}{\partial x_i} dx_i \ , \ dx_i = x_i - x_i^0 \qquad (6)$$

Eq.6 is a general equation for both uniform (absolute) and proportional (relative) perturbation of input parameter [12]. A uniform perturbation in input parameter leads to Eq. 7 whereas proportional perturbation leads to Eq.8 below.

$$\Delta y = \sum_{i=1}^{n} \frac{\partial f(x^0)}{\partial x_i} \qquad (7)$$

$$\Delta y = \sum_{i=1}^{n} \frac{\partial f(x^0)}{\partial x_i} x_i^0 \qquad (8)$$

Considering uniform perturbation in Eq. 7 and using Newton approach, the partial derivative can be calculated as in Eq.9 below.

$$\lim_{\Delta x_i \rightarrow 0} \frac{f(x_i^1, x_{\sim i}^0) - f(x^0)}{\Delta x_i} = \lim_{\Delta x_i \rightarrow 0} \frac{\Delta y^1}{\Delta x_i} = \frac{\partial f(x^0)}{\partial x_i} \qquad (9)$$

Since, practically input parameters don't have same units; therefore, the current results are evaluated using proportional perturbation in Eq.8 using same approach as in Eq.9. Furthermore, partial derivatives solely don't help in ranking the influence of input parameters; hence, fraction of model-output change with respect to the input variable x_i is calculated as differential importance measure, D_i, [15] equated in below in Eq. 10.

$$D_i = \frac{\frac{\partial f(x^0)}{\partial x_i} x_i^0}{\sum_{i=1}^{n} \frac{\partial f(x^0)}{\partial x_i} x_i^0} \qquad (10)$$

In addition, since D_i possess additive property therefore it leads to joint differential importance measure of two or more variables which tells there combined fractional model-output change [15]. Though, it can be performed without additional computational cost and current implementation can be extended to the interaction effect, but this is currently not presented in this work.

The current implementation assumes model to be a scalar function, f, such that $f : R^n \leftarrow R$. This results in a Jacobian row vector J^{1*n} rather than a matrix. However, the current implementation could be scaled to vector function, $f: R^n \leftarrow R^m$, with Jacobian matrix J^{m*n} that results in differential matrix measuring the differential importance of model input x_i with respect to model output y_j. For the given example, x^0 is the normalized means of each input parameter and used as the reference/local point to calculate differential importance measure. The step size is machine epsilon, ϵ, which is scaled with respect to each input parameter using the formula $\epsilon . (1 + |x_i|)$. Other possible variants of step size includes, $(|x_i|) . \sqrt{\epsilon}$, $\epsilon . (|x_i|)$ or simply ϵ. However, through experimentation best suitable step size has been used that avoids vanishing gradient problem and on the same time provides differential importance measure irrespective of parameter units and range. Below in Figure # is the pseudocode for finding D_i using forward finite difference technique.

Computational cost of calculating differential sensitivity measure, of a scalar value function with n variables, using forward finite difference is n model evaluation. Similarly, a vector value function with m outputs and n input variables costs $m * n$ model evaluations. Furthermore, joint differential importance measure of any order can be estimated using the generated Jacobean matrix without any addition model evaluation.

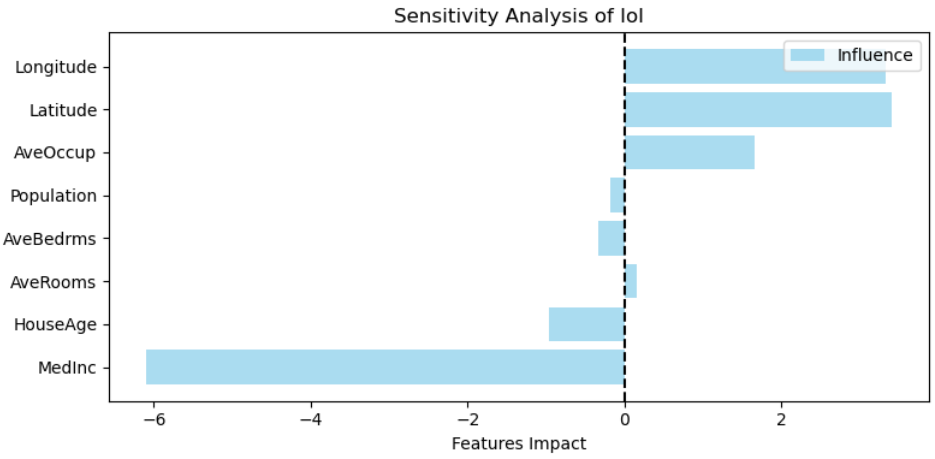

Figure 11: Results of Finite Differece Differential Sensitivity Measure

ALGORITHM 4: PSEUDOCODE FOR FINITE DIFFERENTIAL SENSITIVITY MEASURE

Input: $x^0 \leftarrow reference\ point,\ \ f \leftarrow model$

Output: D

Procedure:

compute $y^0 \leftarrow f(x^0)$

n $\leftarrow$ number of input parameters

$\epsilon \leftarrow$ step size

for $i \leftarrow 1$ ***to*** n ***do***

$\Delta x = \epsilon \,.\, (1 + |x_i|)$

$x_i = x_i + \Delta x$

$y \leftarrow f(x)$

$$J_i = \frac{y - y^0}{\Delta x_i}\, x_i^0$$

$x_i = x_i - \Delta x$

end

for $i \leftarrow 1$ ***to*** n ***do***

$$D_i \leftarrow \frac{J_i}{\sum_{j=1}^{n} J_j}$$

end

return D

Figure 12: Pseudocode for Finite Differences

Finite difference technique has its benefits like any other differentiation technique. It provides magnitude as well as directional information of change in model response with respect to any given input. Secondly, it can be applied to black-box model whose source code is not available and consequently no analytical solution [13]. Third, finite difference performs well if function is non-smooth at point of interest by adapting the step size for computing gradient [16]. Lastly, it can be applied to any complex model with any level of nesting while maintaining efficiency.

However, finite difference has some key limitations. First, the selection of perturbation size or step size which if kept large would lead to inaccurate result and very low would result in zero gradients. Precisely, the truncation error of finite differences is equivalent to the perturbation size [11]. Secondly, differential sensitivity measure relies on the differentiability of the model output which may not be the case in every system.

6. FUTURE WORK

Based on the current work and problem domain, the most suitable approach is differentiation based. Therefore in future differentiation based techniques, i.e. algorithmic differentiation [17], will be applied to perform the sensitivity analysis of input parameters. The new method will be integrated into the current system framework using the same onnx model used for the application of other deterministic techniques. The earlier sensitivity measure, differential sensitivity measure, will be used compare the algorithmic differentiation with existing finite differences method. Finally, some additional sensitivity measure would be introduced into the system for better analysis.

7. BIBLOGRAPHY

[1] J. C. Helton, "Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal," *Reliability Engineering and System Safety*, vol. 42, no. 2–3, 1993, doi: 10.1016/0951-8320(93)90097-I.

[2] P. Wei, Z. Lu, and J. Song, "Variable importance analysis: A comprehensive review," *Reliability Engineering and System Safety*, vol. 142, 2015, doi: 10.1016/j.ress.2015.05.018.

[3] E. Borgonovo and E. Plischke, "Sensitivity analysis: A review of recent advances," *European Journal of Operational Research*, vol. 248, no. 3, 2016, doi: 10.1016/j.ejor.2015.06.032.

[4] A. Saltelli *et al.*, *Global sensitivity analysis: The primer*. in Global Sensitivity Analysis: The Primer. 2008. doi: 10.1002/9780470725184.

[5] S. Kucherenko and B. Iooss, "Derivative based global sensitivity measures," 2014, doi: 10.48550/ARXIV.1412.2619.

[6] E. Borgonovo and C. L. Smith, "A Study of Interactions in the Risk Assessment of Complex Engineering Systems: An Application to Space PSA," *Operations Research*, vol. 59, no. 6, pp. 1461–1476, Dec. 2011, doi: 10.1287/opre.1110.0973.

[7] S. Razavi *et al.*, "The Future of Sensitivity Analysis: An essential discipline for systems modeling and policy support," *Environmental Modelling and Software*, vol. 137, 2021, doi: 10.1016/j.envsoft.2020.104954.

[8] "ONNX: Open Neural Network Exchange." [Online]. Available: https://onnx.ai/

[9] "California Housing Data." https://www.kaggle.com/datasets/dhirajnirne/california-housing-data, 1997. [Online]. Available: https://www.kaggle.com/datasets/dhirajnirne/california-housing-data

[10] T. G. Eschenbach, "Spiderplots versus Tornado Diagrams for Sensitivity Analysis," *Interfaces*, vol. 22, no. 6, 1992, doi: 10.1287/inte.22.6.40.

[11] E. Borgonovo, "Sensitivity analysis with finite changes: An application to modified EOQ models," *European Journal of Operational Research*, vol. 200, no. 1, pp. 127–138, Jan. 2010, doi: 10.1016/j.ejor.2008.12.025.

[12] E. Borgonovo, *Sensitivity Analysis*, vol. 251. in International Series in Operations Research & Management Science, vol. 251. Cham: Springer International Publishing, 2017. doi: 10.1007/978-3-319-52259-3.

[13] J. R. R. A. Martins and J. T. Hwang, "Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models," *AIAA Journal*, vol. 51, no. 11, pp. 2582–2599, Nov. 2013, doi: 10.2514/1.J052184.

[14] J. C. Helton, "Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal," *Reliability Engineering & System Safety*, vol. 42, no. 2–3, pp. 327–367, Jan. 1993, doi: 10.1016/0951-8320(93)90097-I.

[15] E. Borgonovo and G. E. Apostolakis, "A new importance measure for risk-informed decision making," *Reliability Engineering and System Safety*, vol. 72, no. 2, 2001, doi: 10.1016/S0951-8320(00)00108-3.

[16] F. Van Keulen, R. T. Haftka, and N. H. Kim, "Review of options for structural design sensitivity analysis. Part 1: Linear systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 30–33, pp. 3213–3243, Aug. 2005, doi: 10.1016/j.cma.2005.02.002.

[17] J. R. R. A. Martins and A. Ning, *Engineering Design Optimization*, 1st ed. Cambridge University Press, 2021. doi: 10.1017/9781108980647.