A new approach to detecting damage in structures that uses global search techniques and machine learning with vectorized data

T. Le-Xuan[a], T. Nguyen-Chi[a], H. Tran-Ngoc[a], T. Bui-Tien[a] and M. Abdel Wahab[b]

a Department of Bridge and Tunnel Engineering, Faculty of Civil Engineering, University of Transport and Communications, Hanoi, Vietnam;
b Soete Laboratory, Faculty of Engineering and Architecture, Ghent University, Technologiepark Zwijnaarde 903, B-9052 Zwijnaarde, Belgium

Abstract

The paper proposes a new approach to address the local minimum problem in machine learning (ML) using an evolutionary algorithm called Cuckoo search (CS). While ML is widely used, it can be limited by getting trapped in local minima, especially in complex problems such as structural health monitoring. The proposed method combines GD technique for fast convergence with CS for global search capability to solve the problem. The vectorization technique is also applied to reduce computational costs. The approach is evaluated on numerical and experimental models and outperforms CS, ML, and other hybrid ML in terms of accuracy while reducing computational costs.

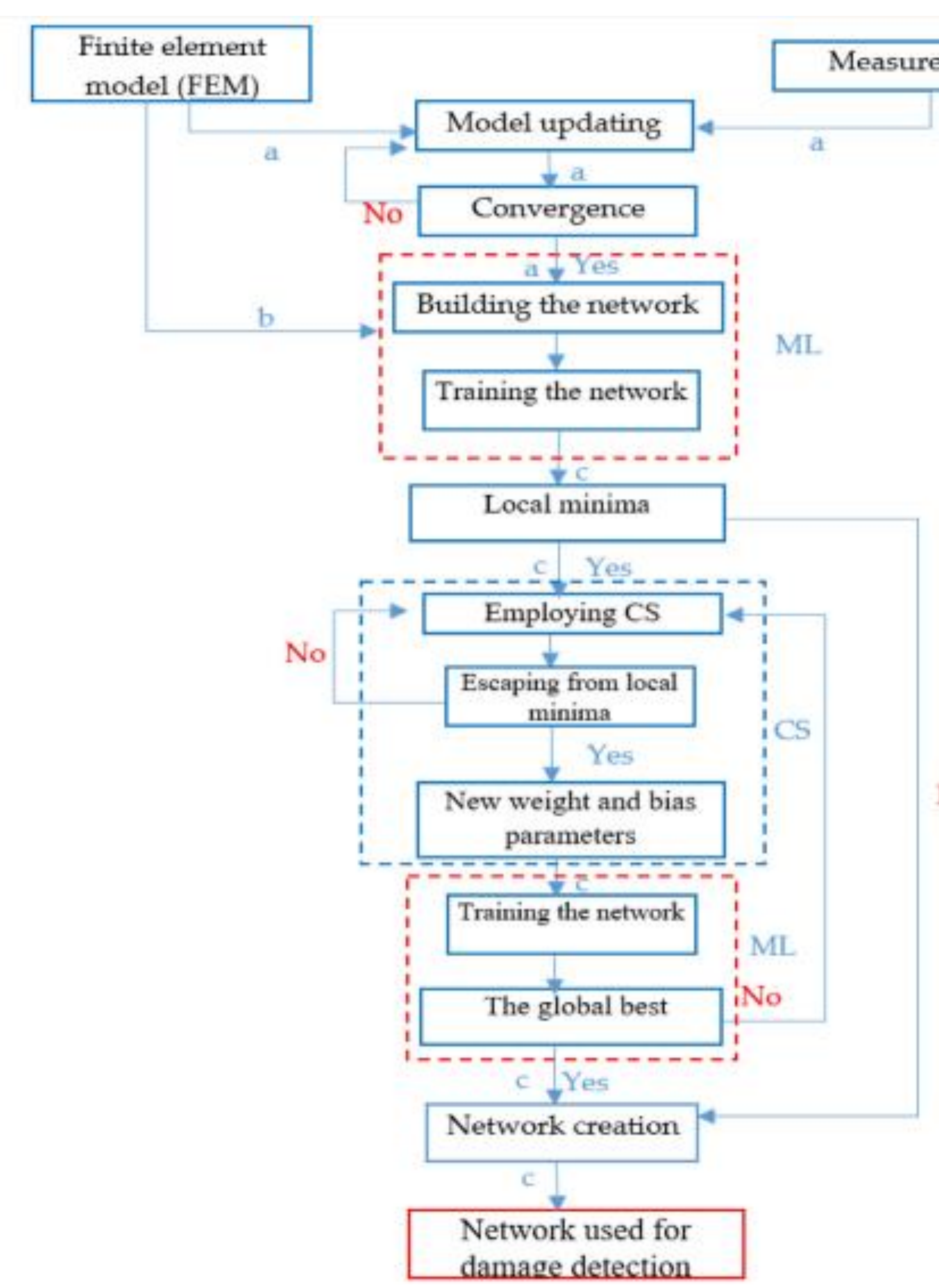
Keywords: Machine Learning, Cuckoo Search, Structural Health Monitoring, Local Minima, Damage Detection, Vectorized data.

Introduction

- Nowadays, the economy is experiencing rapid growth, which results in overloading of the transportation infrastructure. Consequently, bridge damage increases. For this reason, structural health monitoring (SHM) is crucial to detecting structural damage.
- With recent notable advances, ML algorithms have been commonly employed. However, because of the application of backpropagation algorithms based on gradient descent (GD) techniques, the network of ML may be trapped in local minima. This drawback may reduce the accuracy and effectiveness of ML. To remedy these shortcomings, numerous researchers have employed algorithms based on global search techniques to eliminate initial local. Nevertheless, those solutions are only valid under certain circumstances.
- Therefore, this paper proposes a novel machine learning based – on an evolutionary algorithm, name Cuckoo search (CS) to solve the problem which working parallel with Ml during the process of training the network.
- To consider the performance of the proposed method, a large-scale truss bridge and a laboratory measurement with different damage scenarios are employed. Additionally, in oder to deal with a large amount of data is generated, we apply vectorization for the data of the object function. To compare with the proposed method, ML, CS, and the methods of hybrid ML are also used for SHM of the considered structures

Methodology

MLCS2



Numerical model

Single damages

The input data includes the natural frequencies of the first 15 modes at different damage scenarios. The damage level is assigned from 0% to 50% with an interval of 1% for each element. Only truss members on one plane are employed for damage detection.

$$N_e = n_e * n_s \qquad (1)$$

Where: n_e is the number of considered elements, n_s is the number of damage scenatios occurring at one element.

Multiple damages

Damages are generated at 2 any element at the same time. The damage level is assigned from 0% to 50% with an interval of 1% for each element.

$$N_{s2} = n_s \frac{n_e!}{(n_e-2)! * 2!} \qquad (2)$$

Damages are generated at 3 any element at the same time. The damage level is assigned from 0% to 50% with an interval of 1% for each element.

$$N_{s3} = n_s \frac{n_e!}{(n_e-3)! * 3!} \qquad (3)$$

With ! is factorial

Results

For providing further validation, a laboratory beam is employed to evaluate the effectiveness and robustness of MLCS2. To minimize the effect off bearing stiffness, a beam with a free-free boundary condition is considered.

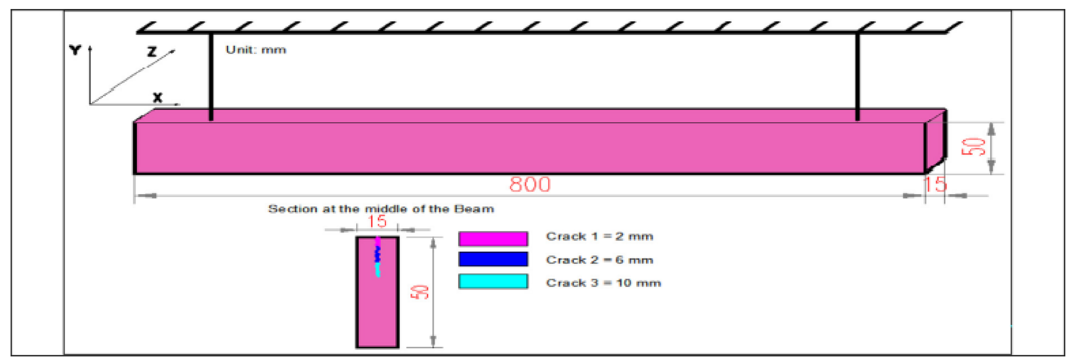


Figure 2. Free-free beam

- The beam is divided into 31 elements, using 3-dimensional (2D) elements with 2 DOFs at each node including translational and rotational displacement. To create datasets tot train the network, the FE model needs to be updated.

Single Damage

Algorithms	MSE-values	R- Values	Total CPU time (second -s)
CS			41245
ML	2.9475	0.995	38
MLCS1	0.8675	0.997	239
MLCS2	0.5016	0.998	112

- MLCS2 is superior to MLCS1, ML, and CS in terms of MSE-values and ***R***-values
- In terms of computational time, ML, MLCS1 and MLCS2 spend 38 s, 239 s, and 112 s to find the optimal solution, whereas CS spends too much time on this process (41245 s)

2 Damage Elements

Algorithms	MSE-values	R- Values	Total CPU time (second-s)
CS			42539
ML	7.0405	0.979	2150
MLCS1	6.1677	0.981	3704
MLCS2	5.8540	0.983	3011

- MLCS2 surpasses CS, ML, and MLCS1 in terms of accuracy (MSE – values and R – values).
- In terms of computational time, ML, MLCS1, and MLCS2 spend 2150 s, 3704 s, and 3011 s in turn to determine the best solution, whereas CS expends 42539s for this process

3 Damage Elements

Algorithms	MSE-values	R- Values	Total CPU time (second -s)
CS			42568
ML	15.4821	0.951	9602
MLCS1	14.5996	0.954	16218
MLCS2	14.546	0.955	10513

- MSE-values obtained from MLCS2 is lowest, at 14.54, whereas those calculated by ML, and MLCS1 are 15.48, and 14.59, respectively.
- Regarding the CPU time (computational cost), CS2 spend 42568 s to look for the global best, compared to only 9602, 16218 s, and 10513 s for ML, MLCS1, and MLCS2, respectively.

Conclusion

- ML significantly reduces computational time and can correctly identify damage cases belonging to the datasets. But, because of the application of backpropagation algorithms based on gradient descent (GD) techniques, the network of ML may be trapped in local minima, leading to reduce the accuracy and effectiveness of ML.
- The application of CS is time-consuming. Besides, Cs may only demonstrate its capacities if the objective function can be arbitrarily selected despite the great benefits of CS.
- MLCS1 assists the network in seeking beneficial starting points and eliminate some initial minima, but the network may still get stuck in other local minima during the process of training the network.
- With the same input data, network architecture and object function, MLCS2 is superior to CS, ML, MLCS1 in terms of accuracy and considerably reduces calculational costs compared to CS

Acknowledgements