Neural Optimization for Pharmaceutical Transportation Under Stressful Conditions

1st Georgia Fargetta
Department of Mathematics and Computer Science
University of Catania
Catania, Italy
georgia.fargetta@unict.it

2nd Alessandro Ortis
Department of Mathematics and Computer Science
University of Catania
Catania, Italy
alessandro.ortis@unict.it

3rd Sebastiano Battiato
Department of Mathematics and Computer Science
University of Catania
Catania, Italy
battiato@dmi.unict.it

***Abstract*—This study presents a refined model for supply chain competition in the transportation of pharmaceutical items, in stress conditions characterized by high demands of critical items and closures. Each shipper aims to minimize its costs by optimizing the parameters of the defined cost function by means of a supervised learning approach which exploits an Artificial Neural Network model. Given the challenges posed by emergency situations, we focus our attention on the imperative to optimize transportation costs from the shipper to the destination, while considering the mode of shipment. Specifically, our analysis leverages the Health dataset from *US Supply Chain Information for COVID-19* to investigate supply chain shipments during the COVID-19 pandemic, a period marked by significant logistical challenges in meeting demand and minimizing the cost across various destination countries. Finally, through this methodology, we present an illustrative example to observe the optimal supply chain solutions using a Neural Optimization Machine.**

***Index Terms*—Artificial Intelligence, Supervised Machine Learning, Health Optimization, Neural Network, Supply Chain, Neural Optimization Machine**

I. Introduction

Emergencies, whether triggered by man-made or natural events, significantly impact our social and economic fabric. Depending on the nature of the emergency, various hazards can arise in affected areas, underscoring the critical importance of emergency management. Businesses, in particular, must implement specialized measures to safeguard their operations from potential adverse effects of emergencies. Hence, proactive planning is essential to establish preparedness protocols before such events occur. In particular, when the emergency encompasses the entire world as with the COVID-19 pandemic, it brings heightened attention to the issues surrounding such emergency situations. Indeed, significant challenges during the pandemic revolved around the supply chain disruptions affecting various goods, including food, medical supplies, among others. Therefore, emergency management has become one of the most important and challenging issues. Moreover, emergency resource storage and distribution have led to strong competition for medical supplies among healthcare institutions. This study aims to underscore the importance of observing the distribution of medical supplies from multiple sources to numerous destinations during an emergency. Specifically, it focuses on the shipment of medical items and supplies to hospitals or pharmacies. In our model, we explicitly analyse how transportation costs of medical items affect the revenue of a shipper. In addition, the shipment costs, and its associated parameters are optimized using Computational Graph Neural Networks (CGNNs), which excel in regression tasks due to their flexibility, non-linear modelling capabilities, and adaptability to data. CGNNs, as universal approximators, can capture complex relationships [1], [2]. They automatically learn features from data, eliminating the need for manual feature engineering. Their adaptability allows them to adjust to the underlying patterns in the dataset during training. CGNNs are scalable, handling simple and complex regression problems, and regularization techniques prevent overfitting [3]. Therefore, after identifying the optimal parameters using a CGNN and comparing the outcomes with Support Vector Regression model [4], the article proceeds to analyze the constrained optimization problem. This issue will be addressed through a numerical example employing a Neural Network Machine [5], followed by a comparison of results with an exact optimization method.

A. Related Work

In recent years, researchers worldwide have analyzed emergency situations stemming from both natural phenomena such as earthquakes, tsunamis, etc., and as a consequence of the major epidemic of 2020, focusing on the transportation costs that as to be minimized. This has led to the analysis of various supply chain models tailored to uncertain emergency situations, where anomalous conditions are observed for transportation costs compared to everyday data. In [6] the authors present a supply chain evacuation model where a population has to be evacuated from crisis areas to shelters and propose an optimization formulation for minimizing a combination of the transportation cost and the transportation time. The authors in [7], [8] proposed a supply chain model to study the competition of healthcare institutions for medical supplies in emergencies caused by natural disasters. In particular, they develop a two-stage procurement planning model in a random environment, in which each healthcare institution seeks to minimize the purchasing cost of medical items and the transportation time solving their constraint optimization problem. Moreover, to solve an optimization model is important to choose an optimization algorithm to find the optimal solution. In [9], [10], the authors studied the multi-resolution finite element simulation for structures and materials and investigated the internal defects in additively manufacturing. The properties of multi-phase heterogeneous materials are optimized using a Neural Optimization Machine to minimize the stress in the simulation domain. In this paper, we show that the performance of the proposed model is superior to that of existing model SVR. In [11], the authors optimized the hyperparameters of SVR and the proposed model could predict patient flow and provide useful suggestions for hospital management. In [12], the authors presented a novel prediction model and compare their model to well-known prediction models such as Support Vector Regression.

B. Contribution

Our research makes several novel contributions to the field of supply chain optimization, distinctively advancing beyond existing literature in the following ways:

- **Integration of Computational Graph Neural Networks (CGNN):** Our study pioneers the application of CGNN to optimize the parameters of the cost function specific to each shipper within the supply chain. Unlike prior works such as Franco et al.'s conceptual modeling using causal loop diagrams [13], which focuses on internal pharmaceutical costs within hospitals, our approach extends to optimizing shipment costs from warehouse to warehouse across national borders.
- **Validation of Neural Optimization Machine (NOM) on Quadratic Functions:** While existing studies explore Neural Network Machines (NOMs) for various function types [5], [10], our research uniquely validates NOM on quadratic cost functions tailored to supply chain logistics. This validation is significant as it addresses a practical scenario of optimizing medical shipment costs during emergencies, which has not been extensively covered in the literature [14]
- **Application Scope and Methodology:** Unlike studies that primarily focus on vehicle routing or general logistics optimization during emergencies [15], [16], our work specifically targets the nuanced optimization of shipment costs using advanced neural network techniques. We introduce a dual neural network structure combining CGNN for parameter analysis and NOM for solving optimization problems, thus innovatively addressing the challenges of emergency supply chain logistics.
- **Distinctive Focus on Supply Chain Optimization:** Our study contributes by integrating theoretical advancements in neural network applications with practical supply chain management, emphasizing the adaptation and fine-tuning of cost function parameters to diverse shipping scenarios. This focused application fills a gap in current research by enhancing the efficiency and adaptability of supply chain operations during critical situations.

In summary, our research not only expands the theoretical foundations of CGNN and NOM applications but also provides practical insights into optimizing supply chain logistics under emergency conditions. By validating these methodologies on quadratic cost functions specific to medical supply chains, we contribute to advancing optimization techniques in supply chain management.

C. Organization

The remainder of the paper is organised as follows: Section I introduces the work; Section II describes the proposed methodology; Section III presents the Computational Graph Neural Network; Section IV describes the Neural Optimization Machine; Section V presents the experimental setup, results, and relative discussions; Section VI concludes the paper.

II. Proposed Model

In this section, we present a supply chain model for the medical supply competition, see also [7], [8]. Let $\mathcal{S}$ be the set of shippers, with typical shipper denoted by s; let $\mathcal{D}$ be the set of destinations, with typical destination denoted by d and let $\mathcal{M}$ be the set of transportation modes, with typical mode denoted by m, in Figure 1 denoted by different colored connections between origin and destination. We assume that N supply medicals, with their respective weight x_{sd}^m, are located in some shippers' warehouse and must be shipped to some destinations, using a specific transportation mode.

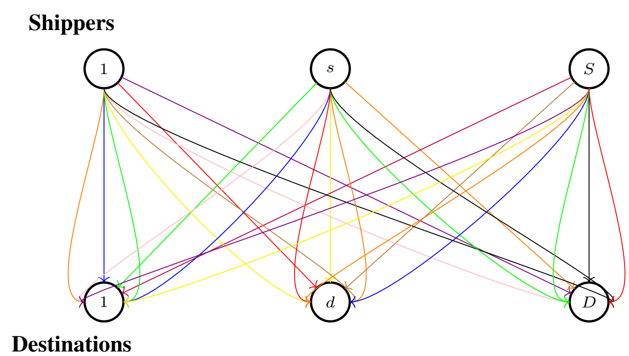

Fig. 1: Supply chain representation. The shipment from each shipper s to each destination d, using different type of modes m, indicated by different colors.

Let $C_s(x_{sd}^m)$ be the shipment cost for each shipper s:

$$C_s(x_{sd}^m) = a_s x_{sd}^m + b_s (x_{sd}^m)^2, \quad (1)$$

where $a_s \in \mathbb{R}$ and $b_s \in \mathbb{R}$ are the personal-type coefficients that capture the shippers' heterogeneity. This function represents the transportation costs that the shipper has to ship medical items from his warehouse s to the destination d. As remarked in [17], the quadratic form of the internal transportation costs does not only allow for tractable analysis but also serves as a good second-order approximation for a broad class of convex costs functions. In particular, a_s models the maximum internal demand willingness rate, and b_s models such willingness elasticity factor. The aim of each shipper is to solve his constraint optimization problem:

$$\text{Minimize} \quad \sum_{d\in D}\sum_{m\in M}\left(a_s x_{sd}^m + b_s (x_{sd}^m)^2\right) \tag{2}$$

$$\text{subject to} \quad x_{sd}^m \geq 0 \tag{3}$$

$$x_{sd}^m \leq W_d^m, \tag{4}$$

where W_d^m is a fixed maximum weight for each shipment and depend on the mode used and the destination chosen and (3) represents the non-negativity constraint for the weight that shippers has sent to each destination in the dataset considered.

III. Computational Graph Neural Network

A computational graph is a visual representation of a mathematical function using the principles of graph theory. Graph theory, at its core, revolves around the concept of nodes connected by edges, with everything in the graph being either a node or an edge. Within a computational graph, nodes represent either input values or functions that manipulate these values. Edges in the graph carry weights as data flows through it. Outbound edges from an input node are weighted with that input value; outbound nodes from a function node are weighted by combining the weights of the inbound edges using the specified function. For instance, we consider the relatively expression (1), which represents $\hat{y}_s$, the value to compare with the target y_s, i.e. the cost of the shipment in dollars. A deep neural network is like a framework for a mathematical function. When we define the architecture of a neural network we are laying out the series of sub-functions and specifying how they should be composed and how they are connected each other. During the training phase of the neural network, we manipulate the parameters of these smaller functions to adjust them on a complex instrument. Let be (1) our function, where a_s and b_s are scalar coefficients. The individual components within this function encompass various mathematical operations: one square, three multiplications and one addition. The tunable aspects of this function are the coefficients a_s and b_s, which in the realm of neural networks, are referred to as weights. The input to the function, represented by x_{sd}^m remain fixed as they are derived from the dataset and are beyond our control during the machine learning process. By modifying the values of our weights a_s and b_s, we can dramatically influence the output of the function. Nevertheless, irrespective of the specific values assigned to a_s and b_s, the function consistently contains terms involving $(x_{sd}^m)^2$. Consequently, our function maintains a distinct but bounded range of potential configurations. We consider a quadratic function as a consequence that in economic optimization models is used to consider this type of function as shipping cost or in general costs for manufacturers, retailers etc... (see [18]–[20]). The cost function for each shipper s to all destination d, using different type of modes m, is depicted as a Computational Graph Neural Network in Fig. (2).

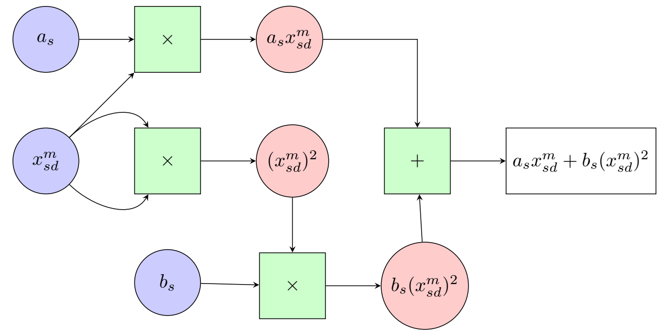

Fig. 2: Computational Graph Neural Network

IV. Neural Optimization Machine

Once the optimal parameters have been found, it is necessary to solve the optimization problem. The method we decided to use to solve our problem is the Neural Optimization Machine (NOM). In [5], [10], the authors demonstrated that this model is efficient for solving constrained, unconstrained and multi-objective optimization problem using neural networks. They compared the model with genetic optimization and swarm optimization algorithms, obtaining excellent performance. Briefly, a NOM solves an optimization problem in using the neural networks' built-in backpropagation algorithm by properly designing the Neural Network architecture. On the one hand, in the backpropagation algorithm, the basic method is stochastic gradient descent. It computes the gradients of the loss function with respect to the weights and biases. The weights and biases are then updated by the gradient information. At the end of the training, a set of local optimal weights and biases are obtained. Moreover, the gradient descent method can also be used to solve the optimization problem. It requires computing the gradient of the outputs of the Neural Network objective function with respect to its inputs. For further information see the following articles [5], [10]. In Section V, we present one of our experiments using NOM for constrained optimization problem, as a illustrative example.

V. Results and Discussion

A. Experimental Dataset

This study used a dataset from Kaggle, *US Supply Chain Information for COVID-19*. We considered the *ORIG-STATE* and the *DEST-STATE* as shippers and destinations, respectively. Each state is identified by its Federal Information Processing Standards (FIPS) code ranging from 01 to 56. In addition, we consider as transportation modes, the *MODE*, i.e. mode of transportation of each shipment from a shipper to a destination. Let be the input *SHIPMT_WGHT*, i.e. the weight of each shipment in pounds and finally let be the target value, y_s, of the shipment in dollars *SHIPMT_VALUE*, i.e. the cost of the shipment in dollars. The dataset contains 45 *SCTG*, Standard Classification of Transported Goods Codes, i.e. 2-digit SCTG Commodity Code of the shipment to identify the typology of the items. For our purpose, we consider the shipment with the

Code 21, i.e. Pharmaceutical Products. As a consequence some shipping origins were losing significance for the study because they contained limited data regarding this type of shipment. Indeed, the number of Shippers considered in our experiments are $|S| = 48$, see Table II.

B. Experimental Settings

1) Baseline SVR: In the SVR (Support Vector Regression) model, a nonlinear regression is performed by mapping the original feature space X onto a new feature space X' using a nonlinear mapping $x \rightarrow \phi(x)$. This transformation allows us to capture nonlinear relationships between the input features x^m_{sd} and the target variable y_s.

Mathematically, the relationship between x^m_{sd} and y_s in the SVR model can be expressed as follows:

$$Y_s = \sum_k \beta_k y_k K(x_k, x^m_{sd}) + b,$$

where Y_s is the predicted target variable for the generical input x^m_{sd}, β_k is the coefficient of the Lagrange multipliers, y_k is the target variable corresponding to the support vectors x_k, $K(x_k, x^m_{sd})$ is the kernel function, which calculates the similarity between the support vectors x_k and the input x^m_{sd}. This kernel function allows us to map the data into a higher-dimensional space where nonlinear relationships can be captured. The kernel helps SVR to capture different types of nonlinear relationships between input features and target variables.

2) Prepocessing: The distribution for training and testing datas is 70% and 30%, respectively. The performance of the model was evaluated through MSE (5). We consider 100 and 1000 epochs of training with a learning rate of 0.001 and batch size 16.

3) Prediction Parameters: The performance of the proposed method is evaluated using the metrics given in Table I : Mean Square Error (MSE) for our model and for the Support Vector Regression (SVR), see [21]. The metric is defined as:

$$MSE = \frac{1}{|S|} \sum_{s=1}^{S} (y_s - \hat{y}_s)^2 \tag{5}$$

C. Result Analysis

For training and evaluating a Support Vector Machine Regression model (SVR), a fundamental component in predictive analytics and machine learning, is used. The dataset undergoes a pivotal split into training and testing subsets. This partitioning strategy ensures that the model is trained on a distinct portion of the data and subsequently evaluated on unseen instances, safeguarding against overfitting and providing an accurate estimation of the model's generalization performance. Then, a crucial preprocessing step ensues, wherein the features are standardized using the StandardScaler. Standardization is imperative for enhancing the model's convergence properties, as it scales the features to have a mean of zero and a standard deviation of one. By normalizing the feature space, we mitigate the influence of outliers and discrepancies in feature magnitudes, fostering a more robust and stable model. We construct an Support Vector Regression Model, i.e. the creation of an SVR model tailored to the dataset's characteristics. Employing a linear kernel, the class instantiates an SVR capable of capturing complex relationships between the input features and target variable. SVRs excel in both linear and nonlinear regression tasks, leveraging the principles of margin maximization to delineate optimal decision boundaries and discern underlying patterns in the data. For the training, the SVR undergoes rigorous training on the standardized training data, assimilating the underlying patterns and intricacies

TABLE I: Optimal Parameters obtained for 100 epochs and relative metrics, for each shipper to each destination using a transportation mode.

Shipper	a_s_100	b_s_100	**best_mse_100**	svr_mse
1	-0.371839	0.000225	**5.197391e+08**	5.256551e+08
2	-0.119827	0.720953	**1.523970e+09**	2.545136e+10
4	-0.400317	0.000067	**8.517217e+08**	1.052328e+10
5	1.366001	0.344382	**1.076113e+07**	1.086703e+07
8	-1.135799	0.002056	**2.629670e+09**	2.967939e+09
9	0.713142	0.000120	**3.858925e+08**	4.146675e+08
10	0.282592	0.023349	**5.934452e+13**	1.243443e+14
12	-0.757118	0.000022	**2.733869e+09**	3.550352e+09
13	0.969651	-0.000039	**2.340329e+11**	2.354039e+11
15	1.267865	0.000140	**1.272984e+08**	1.898105e+08
16	-1.025665	0.042876	**5.958752e+06**	1.776799e+09
17	1.596166	0.000119	**7.591839e+10**	1.165793e+11
18	-0.928571	0.000631	**2.601719e+10**	1.153366e+11
19	-0.264430	0.000199	**8.744938e+07**	1.078784e+08
20	0.728520	0.000372	**2.154563e+09**	3.217343e+09
21	-1.550962	0.000142	**1.925286e+08**	2.104150e+08
22	1.110282	0.000022	**4.405172e+08**	1.966950e+09
23	-0.257233	0.000161	**3.841563e+08**	5.959191e+08
24	-0.103659	0.000015	**6.153357e+09**	5.830063e+10
25	-0.896780	0.000984	**4.422943e+10**	4.762152e+10
26	-0.030042	0.000261	**8.740391e+10**	9.491336e+10
27	-1.424691	0.000558	**1.193523e+09**	1.631018e+09
28	0.577734	0.000079	**2.012408e+07**	1.083249e+08
29	0.241570	0.000047	**4.197499e+11**	1.011049e+12
30	0.817556	0.000895	**1.652810e+07**	3.991144e+08
31	0.888650	0.000008	**2.950662e+08**	3.594547e+08
32	-0.693383	0.614393	**9.822186e+08**	3.491325e+10
33	1.398881	0.007820	**1.428259e+07**	4.192016e+07
34	-0.228766	0.000216	**7.484862e+10**	7.719484e+10
35	0.140957	0.000093	**4.713419e+08**	4.978047e+08
36	-1.152758	0.000202	**2.646952e+10**	3.615392e+10
37	-1.181616	0.000955	**1.578757e+12**	8.541090e+12
38	0.174052	1.306873	**1.259062e+03**	3.098663e+03
39	0.010496	0.000002	**2.130807e+11**	3.669545e+11
40	0.810885	0.000125	**2.731858e+07**	1.500541e+08
41	0.873363	0.000248	**1.646164e+08**	1.787679e+08
42	-0.216764	0.000291	**9.131278e+10**	9.192584e+10
44	-0.965053	0.000161	**4.344464e+08**	7.617139e+08
45	0.941932	0.000065	**2.055724e+10**	3.294369e+10
46	0.257064	0.000172	**5.072158e+06**	2.710161e+08
47	-0.880924	0.004779	**1.052767e+13**	1.279735e+13
48	-1.428957	0.000077	**4.788906e+09**	7.709044e+09
49	1.252321	-0.000018	**7.381890e+08**	1.277091e+09
50	-0.230021	0.003578	**2.957562e+08**	5.419364e+09
51	0.185936	0.001444	**8.406393e+10**	2.554171e+11
53	-0.046740	0.000309	**3.602948e+07**	5.928175e+07
54	-0.599415	0.001017	**5.333128e+09**	4.330959e+10
55	-0.477663	0.006914	**4.365712e+09**	1.955903e+10

inherent in the dataset. Through an iterative optimization process, the model adjusts its parameters to minimize the disparity between predicted and actual target values, iteratively refining its predictive capabilities. Following model training, the predictive prowess of the SVR is put to the test, as it makes informed predictions on the unseen test data. These predictions are subsequently evaluated using the Mean Squared Error (MSE), a quintessential metric for quantifying the model's predictive accuracy. The MSE quantifies the average squared difference between predicted and actual target values, providing valuable insights into the model's performance across the entire test dataset. Finally, we made a comparison between the performance of the current SVR model and the best-performing of our model. By scrutinizing the MSE values, we ascertain whether our CGNN model surpasses the SVR in predictive accuracy. If our CGNN model exhibits superior performance, as evidenced by a lower MSE, the model chosen to optimize the parameters is better. Furthermore, as can be seen from the Table I, the CGNN model is always better than the SVR, i.e. it has a lower MSE. For reasons of space we have decided to report in the Table I only the optimal parameters obtained for 100 epochs, the best_mse for 100 and the svr_mse for 100 epochs. In conclusion, the Computational Graph Neural Network (CGNN) method yields superior results compared to SVR, but its strengths extend beyond mere performance enhancement. This approach enables the modeling and construction of both linear and nonlinear mathematical functions, allowing for the selection of the best-fitting curve and polynomial expression to approximate our data. Thus, depending on the selected problem, the optimization function can be tailored to be more characteristic and specific, taking into account the problem at hand and the available data.

D. Constrained Neural Network Machine

Once the optimal parameters a_s and b_s are found for each shipper, it is interesting to observe their respective optimization problems. Due to space limitations, we have decided to solve an optimization problem for a general shippers. We studied the problem of minimizing shipping costs for second shipper. We consider the Neural Network Machine, in which we obtain the optimal value of the function considering 1000 epochs, batch size 16 and learning rate 0.001. In the numerical example, we solve the following optimization problem for shipper 2.

$$\begin{aligned} \text{Minimize:} \quad C_1(x^1_{2d}) = & -0.119827x^1_{21} + 0.720953(x^1_{21})^2 \\ & -0.119827x^1_{22} + 0.720953(x^1_{22})^2; \\ \text{subject to} \quad & 0 \le x^1_{21} \le 1, \\ & 0 \le x^1_{22} \le 1, \end{aligned}$$

where each constraint correspond to a layer in the Neural Optimization Machine (see [10]). In this example, we define the variable constraints lie within the interval $[0,1]$, with the aim to consider the percentage of shipment weight relative to the maximum weight value in the dataset. This approach ensures the standardization of shipment weights across all shippers, destinations, and modes of transportation. In Table II, we summarize the results obtained using the exact method and the Neural Optimization Machine (NOM). In this case, there is the possibility to use an exact method due to the lower number of variable used for the illustrative example. Moreover, in the case of a complete study, it will not be possible to solve the optimization problem, using an exact method. Therefore, we would be compelled to resort to a non-exact method, such as a classical genetic algorithm, a swarm intelligence algorithm, or indeed utilize the algorithm developed for a neural network that solves optimization problems, such as the NOM. In [5], the authors demonstrate that the NOM outperforms or matches the performance of other non-exact algorithms for constrained optimization and approaches the solution found even with the exact method, which is naturally faster than any non-exact algorithm.

TABLE II: Results of the optimization algorithms

		Optimization Methods	
		Exact Method	Neural Optimization Machine
Results	x^1_{21}	0.0831032	0.08227842
	x^1_{22}	0.0831032	0.08427878

VI. Conclusion and Future Works

This study introduces a pioneering approach to tackle the complexities of supply chain optimization, particularly under emergency conditions such as the transportation of pharmaceutical items during the COVID-19 pandemic. By framing the problem as a regression task and employing a Computational Graph Neural Network (CGNN), we have successfully minimized transportation costs while considering critical factors such as shipment modes and destinations. Our method surpasses the baseline Support Vector Machine Regressor (SVR) across all evaluated shippers, demonstrating its efficacy in real-world scenarios using data from the US Supply Chain Information for COVID-19 dataset.

The utilization of neural network models in optimizing supply chain operations during crises offers substantial insights and practical solutions for mitigating logistical challenges and reducing costs. Our findings underscore the capability of interdisciplinary approaches, integrating advanced machine learning techniques with supply chain management, to enhance resilience and adaptability in the face of unprecedented disruptions. This framework not only addresses immediate operational challenges but also lays the groundwork for future advancements in supply chain optimization methodologies.

Moving forward, future research directions include expanding the scope of our model to compare its performance against other existing models and regression techniques. Further investigation will focus on identifying all optimal solutions for individual shippers and conducting comparative analyses with traditional optimization algorithms. Additionally, ongoing efforts will explore the scalability and applicability of our approach across different emergency scenarios and global supply chain networks.

In conclusion, the methodology presented in this study represents a significant advancement in the field of supply chain optimization, leveraging state-of-the-art neural network models to achieve efficient and adaptable solutions. As we continue to navigate through future emergencies, this research sets a promising precedent for enhancing supply chain resilience and optimizing operations amidst evolving global challenges. In the future, we aim to validate our approach with different datasets, potentially expanding beyond pharmaceuticals to encompass broader supply chain contexts, and exploring alternative optimization models tailored to specific operational constraints and objectives.

Acknowledgment

The authors are supported by PNRR MUR project PE0000013-FAIR.

References

[1] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[2] A. Sharma, S. Singh, and S. Ratna, "Graph neural network operators: A review," *Multimedia Tools and Applications*, vol. 83, no. 8, pp. 23 413–23 436, 2024.

[3] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning, volume 1*, 2016.

[4] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.

[5] J. Chen and Y. Liu, "Neural optimization machine: A neural network approach for optimization," *arXiv preprint arXiv:2208.03897*, 2022.

[6] G. Fargetta and L. Scrimali, "Optimal emergency evacuation with uncertainty," in *Mathematical Analysis in Interdisciplinary Research*, Springer, 2022, pp. 261–279.

[7] G. Fargetta and L. Scrimali, "A two-stage variational inequality for medical supply in emergency management," in *Optimization and Decision Science: ODS, Virtual Conference, November 19, 2020*, Springer, 2021, pp. 91–102.

[8] G. Fargetta, A. Maugeri, and L. Scrimali, "A stochastic nash equilibrium problem for medical supply competition," *Journal of Optimization Theory and Applications*, vol. 193, no. 1, pp. 354–380, 2022.

[9] J. Chen, C. Meng, Y. Gao, and Y. Liu, "Multi-fidelity neural optimization machine for digital twins," *Structural and Multidisciplinary Optimization*, vol. 65, no. 12, p. 340, 2022.

[10] J. Chen and Y. Liu, "Neural optimization machine: A neural network approach for optimization and its application in additive manufacturing with physics-guided learning," *Philosophical Transactions of the Royal Society A*, vol. 381, no. 2260, p. 20 220 405, 2023.

[11] X. Zhang, B. Lu, L. Zhang, *et al.*, "An enhanced grey wolf optimizer boosted machine learning prediction model for patient-flow prediction," *Computers in Biology and Medicine*, vol. 163, p. 107 166, 2023.

[12] Y.-S. Jeong, Y.-J. Byon, M. M. Castro-Neto, and S. M. Easa, "Supervised weighting-online learning algorithm for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1700–1707, 2013.

[13] C. Franco, "A simulation model to evaluate pharmaceutical supply chain costs in hospitals: The case of a colombian hospital," *DARU Journal of Pharmaceutical Sciences*, vol. 28, pp. 1–12, 2020.

[14] D. Justus, J. Brennan, S. Bonner, and A. S. McGough, "Predicting the computational cost of deep learning models," in *2018 IEEE international conference on big data (Big Data)*, IEEE, 2018, pp. 3873–3882.

[15] Y. Zhang and X. Chen, "An optimization model for the vehicle routing problem in multi-product frozen food delivery," *Journal of applied research and technology*, vol. 12, no. 2, pp. 239–250, 2014.

[16] K. Tan, W. Liu, F. Xu, and C. Li, "Optimization model and algorithm of logistics vehicle routing problem under major emergency," *Mathematics*, vol. 11, no. 5, p. 1274, 2023.

[17] O. Candogan, K. Bimpikis, and A. Ozdaglar, "Optimal pricing in networks with externalities," *Operations Research*, vol. 60, no. 4, pp. 883–905, 2012.

[18] G. Colajanni, P. Daniele, and D. Sciacca, "Reagents and swab tests during the covid-19 pandemic: An optimized supply chain management with uavs," *Operations Research Perspectives*, vol. 9, p. 100 257, 2022.

[19] G. Fargetta and L. Scrimali, "Closed-loop supply chain network equilibrium with online second-hand trading," in *Optimization in Artificial Intelligence and Data Sciences: ODS, First Hybrid Conference, Rome, Italy, September 14-17, 2021*, Springer, 2022, pp. 117–127.

[20] G. Fargetta and L. R. Scrimali, "A sustainable dynamic closed-loop supply chain network equilibrium for collectibles markets," *Computational Management Science*, vol. 20, no. 1, p. 19, 2023.

[21] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.