

# Application of Random Search Methods in the Determination of Learning Rate for Training Container Dwell Time Data Using Artificial Neural Networks

Justice Awosonviri Akodia<sup>1</sup>, Clement K. Dzidonu<sup>2</sup>, David King Boison<sup>3</sup>, Philip Kisembe<sup>4</sup>

<sup>1</sup>Accra Institute of Technology, Advanced School of Systems and Data Studies (ASSDAS), Accra, Ghana

<sup>2</sup>Accra Institute of Technology, Accra, Ghana

<sup>3</sup>Knowledge Web Centre, Accra, Ghana

<sup>4</sup>Department of Computer Engineering, Ghana Communication Technology University, Accra, Ghana

Email: Justiceajal@gmail.com, dzidonu@ait.edu.gh, dvboison@gmail.com, pkisembe@gctu.edu.gh

**How to cite this paper:** Akodia, J.A., Dzidonu, C.K., Boison, D.K. and Kisembe, P. (2024) Application of Random Search Methods in the Determination of Learning Rate for Training Container Dwell Time Data Using Artificial Neural Networks. *Intelligent Control and Automation*, 15, 109-124.

<https://doi.org/10.4236/ica.2024.154007>

**Received:** July 29, 2024

**Accepted:** September 26, 2024

**Published:** September 29, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

**Purpose:** This study aimed to enhance the prediction of container dwell time, a crucial factor for optimizing port operations, resource allocation, and supply chain efficiency. Determining an optimal learning rate for training Artificial Neural Networks (ANNs) has remained a challenging task due to the diverse sizes, complexity, and types of data involved. **Design/Method/Approach:** This research used a RandomizedSearchCV algorithm, a random search approach, to bridge this knowledge gap. The algorithm was applied to container dwell time data from the TOS system of the Port of Tema, which included 307,594 container records from 2014 to 2022. **Findings:** The RandomizedSearchCV method outperformed standard training methods both in terms of reducing training time and improving prediction accuracy, highlighting the significant role of the constant learning rate as a hyperparameter. **Research Limitations and Implications:** Although the study provides promising outcomes, the results are limited to the data extracted from the Port of Tema and may differ in other contexts. Further research is needed to generalize these findings across various port systems. **Originality/Value:** This research underscores the potential of RandomizedSearchCV as a valuable tool for optimizing ANN training in container dwell time prediction. It also accentuates the significance of automated learning rate selection, offering novel insights into the optimization of container dwell time prediction, with implications for improving port efficiency and supply chain operations.

---

---

## Keywords

Container Dwell Time Prediction, Artificial Neural Networks (ANNs), Learning Rate Optimization, RandomizedSearchCV Algorithm and Port Operations Efficiency

---

## 1. Background

Artificial Neural Networks (ANN), a remarkable tool from the Machine Learning (ML) arsenal, have found extensive applications across various fields, including literature, health, agriculture, criminology, and statistics [1]-[3]. With the advent of Big Data, the optimization of ANN for effective outcomes has become a significant challenge, especially considering the size and complexity of modern datasets. The efficiency of ANN projects revolves around two central factors: speed, often referred to as the Learning Rate (LR), and prediction or classification accuracies. While the importance of accuracy is well-established, the role of the Learning Rate has emerged as a critical aspect of training neural networks [4]. Despite numerous advances, the arbitrariness in LR selection and its dynamic nature depending on various dataset characteristics have often led to subpar outcomes [5]. This challenge manifests significantly in modeling container dwell time, a key indicator within the maritime sector [6]. Previous efforts using ANN have not achieved the desired prediction accuracies and learning speed [6] [7]. While container dwell time may sound technical, it is simply the time a container spends before being transferred to its next destination. Although various tools have been applied to model this indicator, ANN's application has faced issues. The problem boils down to the selection of the learning rate, something that may seem trivial but has profound implications for the prediction outcomes. This study aims to innovate by applying random search methods to determine the Learning Rate. By eradicating the arbitrariness in its selection, this novel approach aspires to achieve improved speed and accuracy in training ANNs for predicting container dwell time. This will not only bridge the existing methodological gap but also enhance the practical applicability of ANN in the context of complex real-world data. The present study leverages the strengths of ANN, taking into account the recent literature [5] [8] and the challenges faced in modeling container dwell time [6]. As a result of the utilization of random search methods for LR determination, the study offers a fresh perspective on how to optimize prediction outcomes, thereby contributing a valuable addition to the ongoing discourse in the field.

## 2. Literature Review

Given the paramount significance of neural networks in various facets of life [6] [9] [10], an uptick in research has been observed aimed at optimizing their performance. A cornerstone of these studies was the attempt to mathematically represent the operations of human neurons. The pioneering work in this arena was

conducted by [11], who proposed a method of cyclically feeding data into a network to establish pattern recognition, enabling the network to replicate or produce similar outputs when provided with distinct inputs. Their model incorporated various network components, most notably hyperparameters, which are contingent upon the data's characteristics. An essential hyperparameter in this context is the learning rate, which necessitates precise control to yield optimal results. Despite its foundational importance, the model by McCulloch & Pitts faced critiques, primarily pertaining to gradient behavior. Nevertheless, their groundbreaking work set the stage for subsequent contributions such as Hebbian Learning [12], the Backpropagation Algorithm [13], Reinforcement Learning [14] [15], and Spike-Timing-Dependent Plasticity (STDP) [16]. Donald Hebb's 1949 concept illuminated the dynamics of information flow and elucidated the facilitation and inhibition processes of neurons. However, the Hebbian hypothesis wasn't without its shortcomings, particularly its comparative inefficiency vis-à-vis error-correcting methods that facilitate adjustments during both the training phase and at convergence [17]. To address these inadequacies, Paul Werbos introduced the concept of backpropagation in 1974, emphasizing gradient descent for neural network training. This methodology, while revolutionary, had its pitfalls, notably its propensity to get trapped in local minima, the complexities associated with gradient behaviors, and computational demands, among others. Concurrently, other learning algorithms were emerging, such as Reinforcement Learning, characterized by its agent-environment interactive paradigm, which, despite its merits, came with high costs, ethical concerns, and other challenges. STDP, another essential paradigm, is rooted in the biological principles of neural functioning. It theorizes the modulation of synaptic strength based on the exact timing of spikes in pre- and post-synaptic neurons. Despite its biologically inspired foundation, it faces challenges such as noise, limited applicability, and intricate network dynamics. In the pursuit of optimizing learning speed and controlling error oscillations, various methods have been put forth, including the introduction of momentum techniques such as Nesterov Momentum, Gradient Descent, and Stochastic Gradient Descent (SGD). Enhancements to SGD birthed techniques like AdaGrad, RMSprop, AdaDelta, and Adam. All of these developments aimed to refine the rate of learning. However, amidst this vast landscape of algorithms and learning rates, pinpointing the perfect amalgamation remains a daunting task. Further complicating matters, [5] introduced the Adaptable Learning Rate Tree Algorithm (ALR), which, when tested on the MINST dataset, outperformed various SGD methods. This underscores the complexity inherent in choosing the ideal hyperparameters a decision contingent on data type and intricacy. Such revelations have led to the rise of hyperparameter tuning techniques, exemplified by approaches like grid search and random search, as underscored by [18]. While the former involves exhaustive combinations, the latter offers potential efficiency gains. However, despite these advances, a noticeable arbitrariness persists in the choice of learning rates and other hyperparameters among many ML practitioners.

In one notable study by [6], an artificial neural network was applied to model container dwell time. Using a dataset collected over a year from a Middle Eastern container terminal, the analysis focused on 13,733 import containers. After 2000 training cycles with specific configurations, the model achieved a classification accuracy of 65.5%. A misclassification rate of 25.5% suggests that such a model might not be wholly reliable. This research postulates that the suboptimal modeling outcomes could stem from arbitrary choices in learning rate. Furthermore, current explorations like the optimized reinforcement learning method exhibit tangible benefits in practical applications like path planning [19]. Parallel advancements are evident in studies that deploy artificial neural networks and machine learning techniques for diverse applications, ranging from vessel activity labeling to housing price variation investigations [19] [20]. Another commendable stride in this domain is the ARIMAX-LSTM approach, which further underscores the superiority of deep learning models over their traditional counterparts, especially in the realm of container throughput forecasting [21].

The challenge over the years, however, is the ability to combine all of these hyperparameters and algorithms in a way that achieves optimum results. This led to the development of two main algorithms in the Scikit-Learn library of the Python programming language [22]. These two are Random Search and RandomizedSearchCV. According to [23], whilst Random Search considers the permutation of all parameters during training, RandomizedSearchCV considers a few samples of the hyperparameter space, making it suitable for larger datasets [23]. As such, the major difference between the two algorithms is that while Random Search is suitable for smaller datasets and may require a lot more computing resources to be able to train larger datasets, RandomizedSearchCV, which includes cross-validation, is able to handle larger datasets better with faster convergence [23].

This study, rather than devising a novel learning rate function, aspires to address the methodological gap highlighted in [6]. The approach involves the application of RandomizedSearchCV, a component of the Scikit-Learn library, due to its documented advantages over Random Search, to discern the ideal learning rate for classifying or predicting container dwell time. The adoption of this approach helps the model to learn and identify the patterns in the data irrespective of its original distribution (Time Series, Regression, etc) as documented in [6] [24]-[26] and others.

### 3. Methodology

The study's focal point is an exhaustive dataset comprising 307,594 records of freight containers delivered to the Port of Tema, a significant hub in Ghana's maritime network, from 2014 to 2022. This dataset, extracted from the Terminal Operating System (TOS) of the Ghana Ports and Harbours Authority, offers an in-depth view into the operations and logistics within the port. The dependent variable of interest is the dwell days of containers, representing the time a container

spends within the port's premises from arrival to departure. This variable is crucial in assessing the efficiency and effectiveness of the port's operations. The main independent variables in this study are the various characteristics of the containers, which may include size, type, weight, contents, origin, destination, and other relevant attributes. These factors could hold significant insights into the underlying dynamics that affect the dwell time.

Below are the features in the data (**Table 1**).

**Table 1.** Attributes of variables.

No	Variable Name	Classification	Scale of Measurement	Variable Type
1	Container Number	Subject	Nominal	String
2	Trade	Shipment-Level	Categorical (nominal)	String
3	Commodity	Shipment-Level	Categorical (nominal)	String
4	Size	Shipment-Level	Categorical (nominal)	String
5	Weight	Shipment-Level	Ratio	Decimal
6	Shipment Type	Shipment-Level	Categorical (nominal)	String
7	Day of Delivery	Shipment-Level	Categorical (nominal)	String
8	Date of Discharge	Shipment-Level	Ordinal	Date
9	Date of Delivery	Shipment-Level	Ordinal	Date
10	Last Port of call	Shipment-Level	Categorical (nominal)	String
11	Region of Origin	Shipment-Level	Categorical (nominal)	String
12	Fiscal Regime	Shipment-Level	Categorical (nominal)	String
13	Density of Value	Shipment-Level	Numeric	Decimal
14	Shipping Agency	Non-Shipment-Level	Categorical (nominal)	String
15	Freight Forwarder	Non-Shipment-Level	Categorical (nominal)	String
16	Carrier	Non-Shipment-Level	Categorical (nominal)	String
17	Trucking Company	Non-Shipment-Level	Categorical (nominal)	String
18	Risk Level	Non-Shipment-Level	Categorical (nominal)	String
19	Post-Entry	Non-Shipment-Level	Categorical (nominal)	Boolean
20	Scan	Non-Shipment-Level	Categorical (nominal)	Boolean
21	Customs Inspection	Non-Shipment-Level	Categorical (nominal)	Boolean

The method used for this study is adopted from [27] who, after reviewing about 97 Machine Learning (ML) research articles in ten different application areas recommended for the adoption of an appropriate life cycle in conducting ML projects. They recommended the consideration of Data Collection, Data Pre-processing, Model Training, Model Testing and Model Evaluation as the main components of the life cycle. This involved the extraction of Container Dwell Time (CDT) data from the TOS system of Ghana Ports and Harbours Authority. This

extraction was done to include the all the required features of the container after which the dwell days were computed for each record. To be able to make the data machine learning-ready, the following pre-processing steps were taken:

- 1) Outliers were detected and treated.
- 2) The data was divided into categorical and numeric features.
- 3) One-hot encoding was applied to the categorical features so they could assume a standardized scale.
- 4) Standard Scaling was applied to the numeric features.
- 5) Principal Component Analysis was applied to drop non-informative numeric features.
- 6) The dataset was then reorganized and divided into input and output features.

### 3.1. Model Training and Testing

The pre-processed data was split into Training, Test and Validation sets.

To achieve the desired benefits outlined in [27] without having to go through a laborious, ineffective manual process of training, the study adopts a better python algorithm named RandomizedSearchCV, a scikit-learn library in python. The choice of RandomizedSearchCV over Random Search is influenced by the superior outcomes in network training speed and accuracy, as contained in [18].

Below is the set of default hyperparameter arguments required for the application of this algorithm:

RandomizedSearchCV takes the default hyperparameter arguments, (*estimator*, *param\_distributions*, \*, *n\_iter* = 10, *scoring* = None, *n\_jobs* = None, *refit* = True, *cv* = None, *verbose* = 0, *pre\_dispatch* = '2\*n\_jobs', *random\_state* = None, *error\_score* = nan, *return\_train\_score* = False) which are variable.

The data was then trained with sets of the set hyperparameters till convergence.

### 3.2. Model Evaluation

The Mean Squared Error (MSE) is the loss function used to evaluate the performance of the model during training and testing. It measures the average of the squared deviations between the predicted and the expected. Represented by:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y - \hat{y})^2$$

$N$  = Number of observations

$Y$  = Actual observation

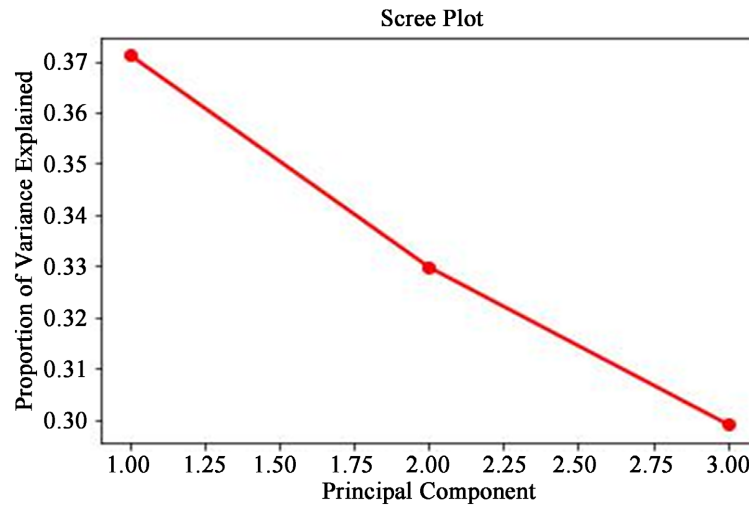
$\hat{y}$  = Predicted value

## 4. Findings

### 4.1. Principal Component Analysis of Numeric Features

The findings from the Principal Component Analysis (PCA) conducted on the three numeric features (BL Type, BL Version, and BOE Version) revealed substantial variations of 30%, 33%, and 37%, respectively. These variations are a

strong indicator that these features capture significant information, warranting their inclusion in the training data. **Figure 1**'s scree plot would provide visual insights into how these components were derived, further validating the need for retention.



**Figure 1.** Scree plot of principal components.

The choice to retain these features adds complexity to the model, but it also potentially increases the model's accuracy. Including features that explain significant variance within the data is a foundational concept in machine learning [28].

## 4.2. Network Training

The process of training a neural network involves selecting the architecture and tuning hyperparameters that influence how the network learns from the data. In the case presented, significant attention was paid to the learning rate, a crucial hyperparameter. A constant learning rate maintains the same learning rate throughout training. In this study, it took 25 hours to converge with an accuracy of 79%. This can offer stable convergence, as the learning rate doesn't change, leading to consistent updates to the model's weights. The adaptive learning rate alters the learning rate based on the training process. Here, it converged in 24 hours with an accuracy of 59%. Although it converged faster, the lower accuracy indicates that it might not have found the most optimal solution. This might be due to the adjustments of the learning rate not being perfectly attuned to the problem at hand. Invsaling (inverse scaling) learning rate gradually reduces the learning rate. It converged in 26 hours with an accuracy of 71%. This method typically slows down the updates to the weights as training progresses, possibly providing a more refined convergence to the minimum of the loss function. The differences in convergence time and accuracy between these learning rate strategies highlight the delicate balance required in selecting a learning rate: Too large a learning rate can cause the training to oscillate or even diverge and Too small a learning rate might lead to slow convergence or getting stuck in a local minimum. Thus, the

learning rate selection is both an art and a science, and it is a pivotal aspect of training neural networks effectively [29]. (Table 2)

**Table 2.** Hyperparameter space for individual learning rates.

Hyperparameter	Choices
Algorithm	RandomizedSearchCV
Training Cross Validations	10
Estimator	MLPRegressor
Number of Iterations	300
Activation Functions	Identity
Number of Hidden Layers	30
Learning Rate	[Constant, Adaptive, Invscaling]
Momentum	SGD
Scoring	Mean Squared Error

A manual comparison between these learning rates would have cumulatively taken 75 hours cumulatively, highlighting the often labor- and time-intensive nature of manual hyperparameter tuning. This extended time might be considered prohibitive in many practical scenarios, and it underscores the need for more automated and efficient methods of hyperparameter optimization [18]. RandomizedSearchCV represents a step towards addressing the time-intensiveness of hyperparameter tuning. By randomly sampling from a distribution of hyperparameters and algorithms, this study reduced training time to 22 hours and improved accuracy to 82%. This method often leads to better results in a shorter time compared to an exhaustive grid search. The success emphasizes the growing importance of automated hyperparameter tuning methods in contemporary machine learning practice, where both time and computational resources are valuable [30]. (Table 3)

**Table 3.** Outcome of training based on standard hyperparameters.

Learning Rate	Convergence Time (hrs)	Accuracy (%)
Constant	25	79
Adaptive	24	59
Invscaling	26	71

The hyperbolic tangent (tanh) function was identified as the optimal activation function in the given study. It’s a non-linear activation function that can capture complex relationships in the data. Unlike the sigmoid function, tanh scales its output to range between -1 and 1, which can provide certain advantages during training, such as mitigating the vanishing gradient problem. The selection of 3 hidden layers indicates the balance the model has achieved between complexity and

performance. Too many layers could lead to overfitting, where the network performs well on the training data but poorly on unseen data. Too few layers might not capture the complexity of the relationships within the data. This optimal number of layers shows the model's ability to generalize well from the training data. (Table 4)

As discussed earlier, the constant learning rate emerged as optimal, reinforcing its robustness in this particular scenario. It offers stable learning and avoids some pitfalls of adaptive methods that might oscillate or overshoot the optimal solution. Stochastic Gradient Descent (SGD) with momentum accelerates the convergence by adding a proportion of the previous weight update to the current update. It helps overcome local minima or saddle points and stabilize the optimization process. (Table 5 & Table 6)

**Table 4.** Hyperparameter Space using RandomizedSearchCV.

Hyperparameter	Choices
Algorithm	RandomizedSearchCV
Training Cross Validations	10
Estimator	MLPRegressor
Number of Iterations	300
Activation Functions	[Identity, logistic, tanh, relu]
Number of Hidden Layers	[3, 5, 10, 15, 20, 25, 30]
Learning Rate	[Constant, Adaptive, Invscaling]
Momentum	[Adam, SGD, ibfgs]
Scoring	Mean Squared Error

**Table 5.** Optimum hyperparameters.

Hyperparameter	Best Value
Activation Functions	tanh
Number of Hidden Layers	3
Learning Rate	Constant
Momentum	SGD

**Table 6.** Comparison of actual and predicted CDT.

SN	CDT (Actual)	CDT (Predicted)
0	13.97	14.43
1	12.53	13.39
2	12.00	11.85
3	12.53	10.81
4	12.53	11.99

**Continued**

5	10.53	11.05
6	12.53	11.2
7	12.53	11.15
8	12.53	12.51
9	12.53	12.2
10	16.94	17.33
11	12.53	14
..	..	..
..	..	..
307,594	16.22	16.66

The selected hyperparameters' synergy emphasizes that neural network performance doesn't depend on individual components but on their interactions. This finding may guide similar studies, stressing the importance of considering hyperparameters in combination rather than isolation. (**Table 7**)

**Table 7.** Comparison of accuracy between RandomizedSearchCV and Nugroho *et al.* (2020) for different Datasets.

SN	Name of Dataset	Accuracy (%)	
		Nugroho <i>et al.</i> (2020)	RandomizedSearchCV
1	Iris	96.7	97.4
2	Ecoli	84.0	90.3
3	Wine	95.2	96.6

In **Table 7**, the effectiveness of RandomizedSearchCV comes to play; the prediction and classification accuracy of the three standard datasets (Iris, Ecoli and Wine) downloaded from Kaggle.com shows improvements after performing neural network training using the methodology presented above.

## 5. Discussions

The findings highlight significant variations of 30%, 33%, and 37% (**Figure 1**) across three numeric features, suggesting that these features capture essential information. This validates their inclusion in the training data, consistent with the concept of retaining features that explain significant variance within data, a fundamental idea in machine learning [28]. The use of PCA can be seen as an effective technique in feature selection, and it has a broad history of applications in various domains. The selection and behavior of the learning rate in training a neural network appear central to this study. Three types of learning rates were explored: constant, adaptive, and invscaling. The study found that the constant learning rate provided better stability and accuracy, despite taking a similar time to converge

as other learning rates. This observation is consistent with the understanding that learning rate selection requires a delicate balance, as it can greatly impact convergence and the quality of the solution [29]. Manual tuning of learning rates was found to be labor-intensive, taking 75 hours cumulatively in this study (Table 2). However, by employing RandomizedSearchCV, the study managed to reduce training time to 22 hours and improve accuracy to 82% (Table 6). This aligns with [30], who emphasized the efficiency of random search over exhaustive grid search methods. Such techniques are becoming increasingly vital in contemporary machine learning, given the premium on time and computational resources [18]. The study identified the hyperbolic tangent (tanh) function as an optimal activation function and chose 3 hidden layers as the ideal number to balance complexity and performance (Table 4). This reflects an understanding of the importance of non-linear activation functions like tanh in capturing complex relationships in data, as well as the careful consideration required in selecting the number of hidden layers to prevent overfitting or underfitting. The methodology and insights from this study are rooted in a rich history of neural network research and optimization, dating back to the work of [11]. While the study did not propose a new learning rate function, it addressed gaps identified in previous research, such as that by [6], who pointed to potential inadequacies stemming from arbitrary learning rate choices. The approach of employing RandomizedSearchCV is in line with current trends towards automated hyperparameter tuning, reflecting the ongoing evolution of techniques for optimizing neural network training. Furthermore, the study's findings resonate with the broader landscape of neural network methodologies and learning rate adjustments, including innovations like Adaptable Learning Rate Tree Algorithm (ALR) (Takase *et al.*, 2018), various enhancements to SGD (Nesterov Momentum, AdaGrad, RMSprop, etc.), and optimization methods in practical applications like path planning and housing price variations [19] [20].

## 6. Conclusions

The study conducted PCA on three numeric features (BL Type, BL Version, and BOE Version), finding significant variations. The substantial variations indicate the importance of these features in the model, leading to the conclusion that they should be retained in the training data. This conclusion aligns with fundamental principles in machine learning where features explaining significant variance are crucial for model complexity and accuracy [28]. An in-depth analysis of different learning rate strategies was conducted, revealing a nuanced balance between convergence time and accuracy. The results indicate that the constant learning rate emerged as optimal, offering stable learning and avoiding pitfalls. This echoes the sentiment in the literature that learning rate selection is a delicate art and science in training neural networks (Smith, 2018). The study highlighted the complexity of manual hyperparameter tuning and introduced automated methods such as RandomizedSearchCV to improve efficiency. This not only reduced training time

but also increased accuracy, reaffirming the importance of automated hyperparameter tuning in contemporary machine learning [18] [30]. The chosen activation function (tanh) and optimal number of hidden layers (3) demonstrate a careful balance between complexity and performance. These choices reflect the ability to capture complex relationships without overfitting or underfitting, highlighting the importance of synergistic interactions between hyperparameters. The study builds on foundational work by pioneers like [11] and various advancements in learning rates and algorithms. By focusing on the application of Randomized SearchCV for container dwell time, it addresses a methodological gap and situates itself in an ongoing conversation about optimizing neural.

## 7. Recommendations

Further research may be needed to explore the complexities identified in the study, possibly through different methodologies or additional datasets to validate the findings. Integration with existing theories might also be recommended to create more comprehensive models that better explain phenomena. These actions could enhance the theoretical understanding of the subject and pave the way for more informed scientific explorations. The study's findings might lead to a recommendation for the standardization or widespread adoption of certain empirical methods in other fields where they may be applicable. Additionally, there may be encouragement for similar empirical studies in other industries to gauge if the findings can be generalized more broadly. Such recommendations could strengthen the empirical literature by providing common ground and wider applicability. The study might recommend wider adoption of specific methodologies and technologies proven effective across industries facing similar challenges. This could lead to a more effective and efficient way of handling particular tasks or problems. Additionally, the study may suggest the need for specialized training or professional development in these areas to ensure that practitioners are equipped to leverage these tools effectively.

## 8. Implications for Theory, Policy and Practice

### 8.1. Implications for Theory

The study's findings offer significant contributions to the theoretical framework within the field. By focusing on the complex interactions between different hyperparameters and employing methods such as PCA and automated hyperparameter tuning, a deeper understanding of neural network behaviors is achieved. This not only enhances the current theoretical models but also paves the way for future research to explore these complexities further, encouraging a more nuanced approach to AI and machine learning.

### 8.2. Implications for Empirical Literature

The research bridges gap in empirical literature, specifically concentrating on container dwell time and the methodologies associated with it. This adds novelty to the body of knowledge, making the study's insights potentially generalizable to

other domains. Furthermore, by employing advanced techniques like RandomizedSearchCV, the research supports the growing empirical evidence related to automated hyperparameter tuning. This methodological advancement could serve as a precedent for further experimentation in the area, enriching the empirical landscape.

### 8.3. Policy Relevance

*The study has clear implications for policy, particularly in the shipping and logistics industry.* Accurate predictions of container dwell time can lead to more precise resource allocation, influencing industry-wide policies. Moreover, the insights into AI methodologies may inform broader policies related to the adoption of machine learning in various sectors. This relevance emphasizes the bridge between research and real-world application, encouraging more data driven and evidence based policymaking.

### 8.4. Implications for Practice

From a practical standpoint, the study's findings could dramatically affect the shipping and logistics industry by enhancing operational efficiency in container management. The reduction in container dwell time could lead to substantial cost savings and waste reduction. Furthermore, the techniques and methodologies used could be applied across different industries facing similar challenges, highlighting the study's broad relevance in various practical domains.

### 8.5. Social Impact

*The social implications of the study are multifaceted.* Improved efficiency in container management could have substantial environmental benefits by reducing fuel consumption and emissions, aligning with societal sustainability goals. The ripple effect on economic growth through enhanced productivity within the shipping and logistics industry adds to the broader societal relevance of the research. Moreover, the educational value of advancing understanding in neural network behaviors supports academic and industry learning, underscoring the broader societal benefits that transcend the specific focus of the study.

## 9. Limitations and Future Research Direction

### 9.1. Limitations

The study's limitations begin with the variation analysis, highlighting variations of 30%, 33%, and 37% across three numeric features, validating their inclusion. However, the study might be limited in whether other feature selections were thoroughly explored and the effect of these variations on different types of models. This may affect the generalizability of the findings to other contexts or datasets. Additionally, while the study found that a constant learning rate provided stability and accuracy, this exploration might be limited to specific types of learning rates, possibly overlooking other state-of-the-art learning rate strategies. The specific

focus on certain learning rates could have prevented a more comprehensive understanding of how various learning rate strategies interact with different neural network configurations. In terms of manual tuning, the 75 hours spent is a significant time investment. Though the randomized search CV reduced the training time, a limitation might exist regarding whether other optimization techniques could have been more efficient. This raises questions about potential trade offs in model robustness or interpretability that were not addressed in the study. Furthermore, the study identified the tanh function and three hidden layers as optimal but might be limited in the exploration of other activation functions or hidden layer configurations. This choice could limit understanding of how different functions and configurations interact with various types of data or problems. Additionally, the context specific nature of the findings might limit the broader applicability of the methodologies employed, restricting their transferability to different domains or problems.

## 9.2. Future Research Directions

Future research could broaden the scope by exploring different variations or combinations of features, thus deepening the understanding of feature importance in neural network models. Investigating how these features affect performance across diverse applications could unlock new avenues of model optimization. The finding regarding constant learning rate efficiency opens the door to investigating the application of other innovative learning rate strategies across various models. This direction could lead to new insights into how different learning rates influence the convergence and quality of different neural network architectures. Efficiency in hyperparameter tuning, as shown by RandomizedSearchCV, points to future exploration of different optimization techniques. Research could delve into methods such as Bayesian optimization or Genetic Algorithms to further improve efficiency, balancing accuracy and computational resource demands. A study focusing on different activation functions and hidden layer configurations could be a promising future direction. Understanding how they interact with different types of data and problems can lead to more nuanced and effective model designs. Lastly, transferring the methodologies and findings to different domains or problems would broaden the study's impact. The specificity of the findings presents an opportunity for future research to adapt and apply these insights to various fields, potentially transforming practices across diverse areas of machine learning.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Obermeyer, Z. and Emanuel, E.J. (2016) Predicting the Future—Big Data, Machine Learning, and Clinical Medicine. *New England Journal of Medicine*, **375**, 1216-1219. <https://doi.org/10.1056/nejmp1606181>

- [2] Rajkomar, A., Dean, J. and Kohane, I. (2019) Machine Learning in Medicine. *New England Journal of Medicine*, **380**, 1347-1358.  
<https://doi.org/10.1056/nejmra1814259>
- [3] Mazzei, D. and Ramjattan, R. (2022) Machine Learning for Industry 4.0: A Systematic Review Using Deep Learning-Based Topic Modelling. *Sensors*, **22**, Article 8641.  
<https://doi.org/10.3390/s22228641>
- [4] Liang, J., Xu, Y., Bao, C., Quan, Y. and Ji, H. (2019) Barzilai-Borwein-Based Adaptive Learning Rate for Deep Learning. *Pattern Recognition Letters*, **128**, 197-203.  
<https://doi.org/10.1016/j.patrec.2019.08.029>
- [5] Takase, T., Oyama, S. and Kurihara, M. (2018) Effective Neural Network Training with Adaptive Learning Rate Based on Training Loss. *Neural Networks*, **101**, 68-78.  
<https://doi.org/10.1016/j.neunet.2018.01.016>
- [6] Kourouniotti, I., Polydoropoulou, A. and Tsiklidis, C. (2016) Development of Models Predicting Dwell Time of Import Containers in Port Container Terminals—An Artificial Neural Networks Application. *Transportation Research Procedia*, **14**, 243-252. <https://doi.org/10.1016/j.trpro.2016.05.061>
- [7] Kourouniotti, I. and Polydoropoulou, A. (2017) Identification of Container Dwell Time Determinants Using Aggregate Data. *International Journal of Transport Economics*, **44**, 567-588.
- [8] Zhao, H., Liu, F., Zhang, H. and Liang, Z. (2019) Research on a Learning Rate with Energy Index in Deep Learning. *Neural Networks*, **110**, 225-231.  
<https://doi.org/10.1016/j.neunet.2018.12.009>
- [9] López, O.A.M., López, A.M. and Crossa, J. (2022) Multivariate Statistical Machine Learning Methods for Genomic Prediction. Springer.  
<https://doi.org/10.1007/978-3-030-89010-0>
- [10] Montesinos López, O.A., Montesinos López, A. and Crossa, J. (2022) Fundamentals of Artificial Neural Networks and Deep Learning. In: Montesinos López, O.A., Montesinos López, A. and Crossa, J., Eds., *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, Springer International Publishing, 379-425.  
[https://doi.org/10.1007/978-3-030-89010-0\\_10](https://doi.org/10.1007/978-3-030-89010-0_10)
- [11] McCulloch, W.S. and Pitts, W. (1943) A Logical Calculus of the Ideas Immanent in Nervous Activity. *The Bulletin of Mathematical Biophysics*, **5**, 115-133.  
<https://doi.org/10.1007/bf02478259>
- [12] Choe, Y. (2014) Hebbian Learning. In: Jaeger, D. and Jung, R., Eds., *Encyclopedia of Computational Neuroscience*, Springer, 1-5.  
[https://doi.org/10.1007/978-1-4614-7320-6\\_672-1](https://doi.org/10.1007/978-1-4614-7320-6_672-1)
- [13] Werbos, P. (1974) Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Harvard University Press.  
<https://www.researchgate.net/publication/35657389>
- [14] Samuel, A.L. (1959) Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, **3**, 210-229.  
<https://doi.org/10.1147/rd.33.0210>
- [15] Sutton, R.S. and Barto, A.G. (2018) Reinforcement Learning: An Introduction. 2nd Edition, MIT Press.  
<https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf>
- [16] Gerstner, W. and Kistler, W.M. (2002) Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press.  
<https://doi.org/10.1017/cbo9780511815706>

- [17] McClelland, J.L. (2005) Hebbian Learning 1 Running Head: HEBBIAN LEARNING How Far Can You Go with Hebbian Learning, and When Does It Lead You Astray? <https://www.semanticscholar.org/paper/Running-head-%3A-HEBBIAN-LEARNING-How-Far-Can-You-Go-McClelland/881ec0684974e5ee986d56bfd770f0ff7e638c50#extracted>
- [18] Bergstra, J. and Bengio, Y. (2012) Random Search for Hyper-Parameter Optimization. <http://scikit-learn.sourceforge.net>
- [19] Liu, H., Fu, Q., Du, L., Zhang, T., Yu, G., Han, S., *et al.* (2022) Learning Rate Perturbation. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, Atlanta, 17-21 October 2022, 4234-4238. <https://doi.org/10.1145/3511808.3557626>
- [20] Cankaya, B., Eren Tokgoz, B., Dag, A. and Santosh, K.C. (2021) Development of a Machine-Learning-Based Decision Support Mechanism for Predicting Chemical Tanker Cleaning Activity. *Journal of Modelling in Management*, **16**, 1138-1165. <https://doi.org/10.1108/jm2-12-2019-0284>
- [21] Shankar, S., Punia, S. and Ilavarasan, P.V. (2021) Deep Learning-Based Container Throughput Forecasting: A Triple Bottom Line Approach. *Industrial Management & Data Systems*, **121**, 2100-2117. <https://doi.org/10.1108/imds-12-2020-0704>
- [22] Fabianpedregosa, F.P., *et al.* (2011) Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, **12**, 2825-2830. <http://scikit-learn.sourceforge.net>
- [23] Fabianpedregosa, F.P., *et al.* (n.d.) Comparing Randomized Search and Grid Search for Hyperparameter Estimation. [https://scikit-learn.org/stable/auto\\_examples/model\\_selection/plot\\_randomized\\_search.html](https://scikit-learn.org/stable/auto_examples/model_selection/plot_randomized_search.html)
- [24] Yoo, S. and Oh, D. (2018) An Artificial Neural Network-Based Fall Detection. *International Journal of Engineering Business Management*, **10**, 1-7. <https://doi.org/10.1177/1847979018787905>
- [25] Olawoyin, A. and Chen, Y. (2018) Predicting the Future with Artificial Neural Network. *Procedia Computer Science*, **140**, 383-392. <https://doi.org/10.1016/j.procs.2018.10.300>
- [26] Basheer, I.A. and Hajmeer, M. (2000) Artificial Neural Networks: Fundamentals, Computing, Design, and Application. *Journal of Microbiological Methods*, **43**, 3-31. <https://www.sciencedirect.com/journal/journal-of-microbiological-methods> [https://doi.org/10.1016/s0167-7012\(00\)00201-3](https://doi.org/10.1016/s0167-7012(00)00201-3)
- [27] Kamiri, J. and Mariga, G. (2021) Research Methods in Machine Learning: A Content Analysis. *International Journal of Computer and Information Technology* (2279-0764), **10**, 78-91. <https://doi.org/10.24203/ijcit.v10i2.79>
- [28] Jolliffe, I.T. (2002) *Principal Component Analysis*. 2nd Edition, Springer-Verlag. <https://doi.org/10.1007/b98835>
- [29] Smith, L.E. (2018) Convergence and Divergence in Statistical and Programmatic Approaches to Address Child Stunting and Wasting. *The Journal of Nutrition*, **148**, 823-824. <https://doi.org/10.1093/jn/nxy098>
- [30] Bergstra, J., Bardenet, R., Bengio, Y. and Kégl, B. (2011) Algorithms for Hyper-Parameter Optimization. *Proceedings of the 24th International Conference on Neural Information Processing Systems*, Granada, 12-15 December 2011, 2546-2554. [https://proceedings.neurips.cc/paper\\_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf)