actuators

MDPI

Article

Machine Learning Models for Assistance from Soft Robotic Elbow Exoskeleton to Reduce Musculoskeletal Disorders

Sanjana Suresh, Inderjeet Singh, Muthu B. J. Wijesundara *

Biomedical Technologies Division, The University of Texas at Arlington Research Institute, 7300 Jck Newell Blvd S, Fort Worth, TX 76118, USA; sanjana.suresh2@uta.edu (S.S.); inderjeet.singh@uta.edu (I.S.)
* Correspondence: muthuw@uta.edu

Abstract: Musculoskeletal disorders are very common injuries among occupational and healthcare workers. These injuries are preventable in many scenarios using exoskeleton-based assistive technology. Soft robotics initiates an evolution in exoskeleton devices due to their safe human interactions, ergonomic design, and adaptive characteristics. Despite their enormous advantages, it is a challenging task to model and control soft robotic devices due to their inherent nonlinearity and hysteresis. Learning-based approaches are becoming more popular to overcome these limitations. This work proposes an approach to estimate the pressure input for a pneumatically actuated soft robotic elbow exoskeleton to assist occupational workers to avoid musculoskeletal disorders. An elbow exoskeleton design made up of modular pneumatic soft actuators is discussed, which helps to flex/extend an elbow joint. Machine learning (ML) approaches are used to develop a relationship between the air pressure, the bending angle of the elbow, and the percentage of the weight of the arm to be assisted by the exoskeleton. The most popular and widely used regression-based ML approaches are applied and compared in terms of accuracy and computation cost. Further, a modified KNN (K-Nearest Neighbor) approach is proposed, which outperforms the accuracy of other approaches.

Keywords: soft robotics; exoskeletons; machine learning

check for updates

Academic Editor: Steve Davis

Received: 11 December 2024
Revised: 7 January 2025
Accepted: 17 January 2025
Published: 22 January 2025

Citation: Suresh, S.; Singh, I.; Wijesundara, M.B.J. Machine Learning Models for Assistance from Soft Robotic Elbow Exoskeleton to Reduce Musculoskeletal Disorders. *Actuators* **2025**, *14*, 44. https://doi.org/10.3390/act14020044

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

People with physical disabilities, such as limb weakness, constantly face challenges that can negatively impact their daily activities of living; therefore, the introduction of exoskeleton devices has been vastly researched to be used as assistive technology [1,2]. Numerous assistive rigid exoskeleton devices are actively being explored and developed for different joints [3–5]. Many years ago, the field of soft robotics was proven to greatly improve the design of exoskeletons. Due to the nature of soft materials that are meant to produce dynamic characteristics while operating, the result is leading to an ongoing issue of nonlinearity that cannot be explained effectively by a simple mathematical model. The problem, however, can be easily solved by the application of Machine Learning (ML) algorithms [6,7]. ML is solves many complex problems due to its various techniques that are strictly learned from observational data. The completed model can finally deliver the learned ability by capturing the nonlinear representation of the given task. It has been successfully employed in many working fields from different industries, and the results are very well-acknowledged.

There are a few previous works where different ML models are used in the field of exoskeletons. A GMR ML model is used for generating the fingertip force trajectories for

the SAFER exoskeleton used for hand rehabilitation [8]. Various major ML algorithms like Decision Tree, KNN, and SVM (Support Vector Machine) were used for the purpose of classification of EMG (Electromyography) signals from the three distinct movements of the elbow and the shoulder. Over 40 models were thoroughly reviewed in order to conclude that the combined model of the TKEO (Teager–Kaiser energy operator) and ensemble (subspace KNN) offered the highest score for all the evaluation metrics to classify EMG signals [9]. Chen et al. [10] developed a system where an MYO (muscle) armband employs machine learning algorithms to recognize gestures from the non-affected hand of a stroke patient. These gestures are then replicated on the affected hand using an exoskeleton glove, facilitating synchronized motor recovery. The Neural Network ML algorithm is very much established in the field of robotics due to its flexibility, and it is becoming more relevant in many research areas. For instance, using Neural Work with NARX for the gait classifier was analyzed to be an adequate replacement for the foot sensors in the lower limb exoskeleton robot [11]. The popularity of the Neural Network certainly cannot be denied; however, it does not always carry out the absolute best prediction most of the time. One such example is the prediction for human activities when the Random Forest algorithm obtained the leading performance from all the conducted experiments [12]. Meanwhile, KNN with the AdaBoost method managed to surpass both the Neural Network and even Random Forest for the same goal of human activities prediction [13]. The main explanation is actually because the datasets employed in both of the learning models are not the same. In other words, ML is heavily based on a data interpretation of a specific application, and each source of data is preferred as a modality [14]. In the work of Zheng et al. [14], several sources like the IMU sensors (knee, ankle, and toe joint angles), the body measurements of the participants, etc., were given as the inputs for constructing the models of the hip exoskeleton. In contrast, the joint angle can also be served as the output if the task is now asking for the angle estimation [15]. Unlike the IMU sensors used to collect ML input data in the above-mentioned work, different types of sensors, e.g., EMG, the distance sensor, etc., can be used for different types of input data according to the application. The data from the hand exoskeleton was acquired by using the EMG sensor, and then the resulting data collection was applied to train the model of the exoskeleton glove for the purpose of recognizing the four different postures of a hand [16]. Like EMG signals for learning, the accuracy score for labeling the forearm positions was relatively high when feeding the FMG signals as the alternative inputs [17]. sEMG signals are also widely utilized to enhance grasp assistance by enabling the precise estimation of the user's intended grasping force, which is essential for rehabilitation and prosthetic control. This has been achieved through regression models that map filtered sEMG signals to force predictions, providing an effective framework for adaptive and personalized assistive technologies [18].

In most of the literature, ML algorithms used for exoskeletons are used to estimate gait patterns, postures, or joint movements. Currently, no research specifically addresses the use of machine learning approaches to predict the required pressure for a soft robotic elbow exoskeleton, which is a critical aspect of soft robotics applications. This study aims to fill that gap by using ML models to predict pressure requirements. The objective of this work is to predict pressure for an elbow exoskeleton to help reduce musculoskeletal disorders in occupational workers by offloading a specified amount of weight from their elbow joints. The following are the main contributions of this paper:

- A relationship between pressure, weight, and bending angle is established using experimental data from the soft pneumatic elbow exoskeleton.
- Various machine learning models are applied and evaluated based on their accuracy and computational efficiency.

- A novel hybrid machine learning model (KNN-Linear) is proposed, demonstrating superior performance compared to other evaluated models.

2. Soft Robotic Elbow Exoskeleton

The soft robotic elbow exoskeleton, shown in Figure 1A, is fabricated using T4 silicone material. Detailed manufacturing procedures and the characterization of the exoskeleton are described in our previous work [19]. For collecting experimental data, a forearm and upper arm assembly, connected via an elbow joint, was designed and 3D printed to replicate human arm motion. The elbow exoskeleton is mounted on the test setup, as shown in Figure 1B. An IMU sensor is integrated to monitor the joint's bending angle. The hardware configuration includes two pneumatic pumps, two solenoid valves, and a pressure sensor. One pump pressurizes the exoskeleton to increase the bending angle, while the other creates a vacuum to decrease it. The solenoid valves toggle between pressure and vacuum modes. Additionally, the pressure sensor continuously monitors the internal pressure of the actuator during bending. The architecture of the control box for this exoskeleton is shown in our previous work [19]. A hollow section is incorporated into the forearm design to accommodate weights of known values. This allows for the collection of data correlating pressure (measured by the pressure sensor) with the bending angle (measured by the IMU sensor) under varying load conditions.

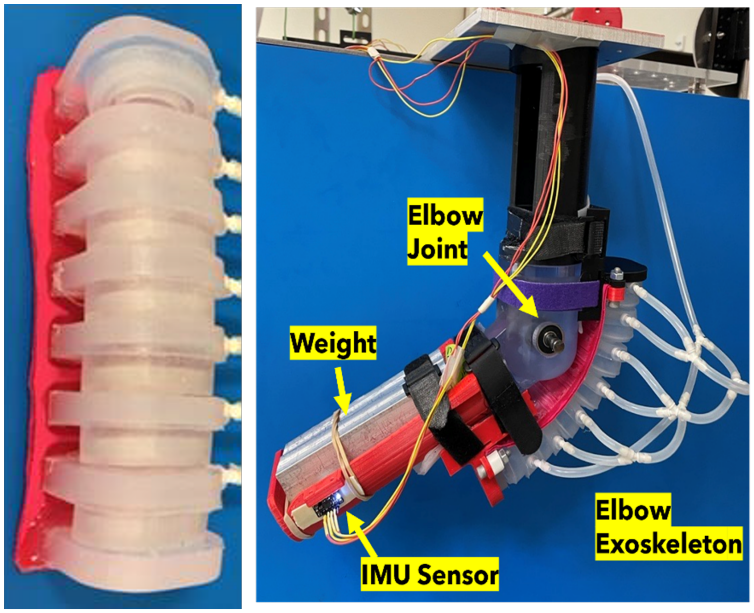


Figure 1. Elbow Exoskeleton: (**A**) soft pneumatic elbow exoskeleton; (**B**) experimental setup for data collection

3. Problem Statement

To prevent musculoskeletal disorders, it is essential to offload a portion of the load borne by occupational workers. This study focuses on reducing the load on the elbow joint using a pneumatically operated soft elbow exoskeleton. The objective is to develop a model capable of estimating the pressure required to offload a specific percentage of the load and achieve a target bending angle during occupational tasks.

Given the inherent nonlinearities of soft materials in robotic systems, a machine learning approach is employed to enhance model accuracy. Various machine learning models are applied and compared to identify the most suitable model for predicting the

required pressure under these conditions. This exploration aims to provide an efficient and reliable framework for integrating soft exoskeletons into occupational settings, minimizing strain on workers and improving ergonomic safety. The data used for machine learning models are collected using the experimental setup shown in Figure 1.

4. Machine Learning Models

This study outlines the various stages and ML techniques employed to train and build an effective pressure prediction model for the elbow exoskeleton, utilizing a data-driven approach to address complex nonlinear problems.

4.1. Data Collection and Pre-Processing

Raw data are collected by varying the pressure continuously from 0 kPa to 150 kPa, with weights ranging from 0 to 1500 g in 50-g increments. The IMU records the bending angle, while the pressure sensor captures the corresponding pressure for each test run. Our dataset comprises 80,430 data points, with weight and angle as feature variables and pressure as the the target variable, which is to be predicted.

To ensure the integrity and accuracy of our machine learning model, it is imperative to thoroughly analyze and address any potential data issues. Given the data-driven nature of machine learning, this step is crucial in mitigating bias in pressure prediction and managing complexity between features and the target variable by establishing clear relationships among them.

The initial phase involves identifying and eliminating any null or duplicate values in the dataset. Approximately 9500 duplicate entries are detected and subsequently removed so that the model is able to generalize to unseen data. Following this, a detailed analysis of the statistical mean and standard deviation revealed significant variations in data points from the mean, which could potentially lead to less accurate predictions. To address this issue, we applied feature scaling using the MinMaxScaler function [20], ensuring that all features were normalized within a specified range.

During the learning process, the elbow dataset is randomly divided into two subsets: the training data, comprising 75% of the dataset, and the test (or unseen) data, comprising the rest 25%. The training data are used for model development and learning, while the test data are reserved for assessing the model's performance and determining error metrics.

To prepare the dataset for modeling, we conducted a comprehensive analysis of the relationships among the features and the target variable. It is observed that angle vs. pressure has a linear relation, as illustrated in Figure 2. However, since the objective is to accurately predict pressure based on a given weight and bending angle, establishing a relationship among all these variables is essential to better capture their variations. To achieve this relationship, we implemented polynomial feature transformation, which converts the input features into higher-degree polynomial features. The optimal polynomial degree was determined using grid search with cross-validation (GridSearchCV) combined with regression modeling to avoid overfitting concerns [21].

The optimal polynomial degree for feature transformation is determined by using the Mean Squared Error (MSE) on different subsets of the test data. GridSearchCV is performed with five folds where, in each iteration, four of the folds are used to train the model, while the remaining fold is used as the test set. Each polynomial degree, ranging from 1 to 10, is assessed by generating polynomial features, training a linear regression model on the transformed data, and computing the test set MSE for each degree. The degree that results in the lowest MSE across all five iterations is chosen as the optimal degree. This approach helps ensure that the model generalizes well, as it tests the model's performance on different subsets of the data.

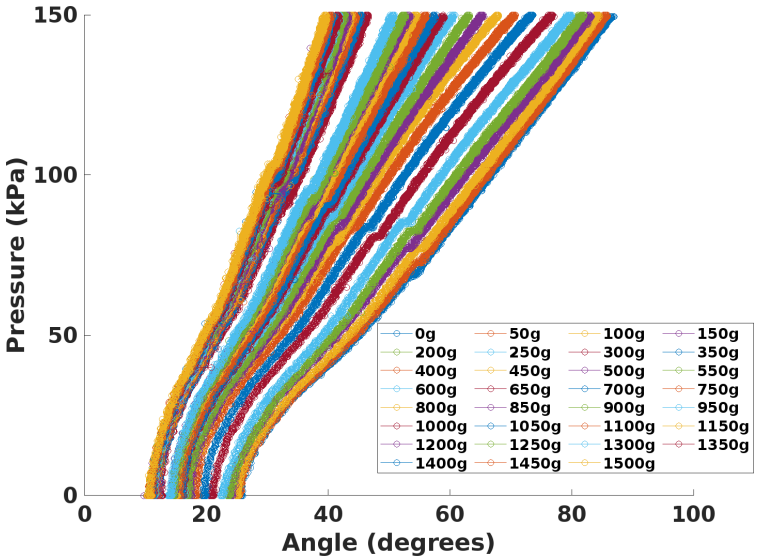


Figure 2. Pressure vs. angle plot with different weights.

As shown in Figure 3, it is evident that degree 6 minimized the MSE. As a result of this, our final dataset has 27 features capturing different combination of weight and angle to produce robust pressure predictions.

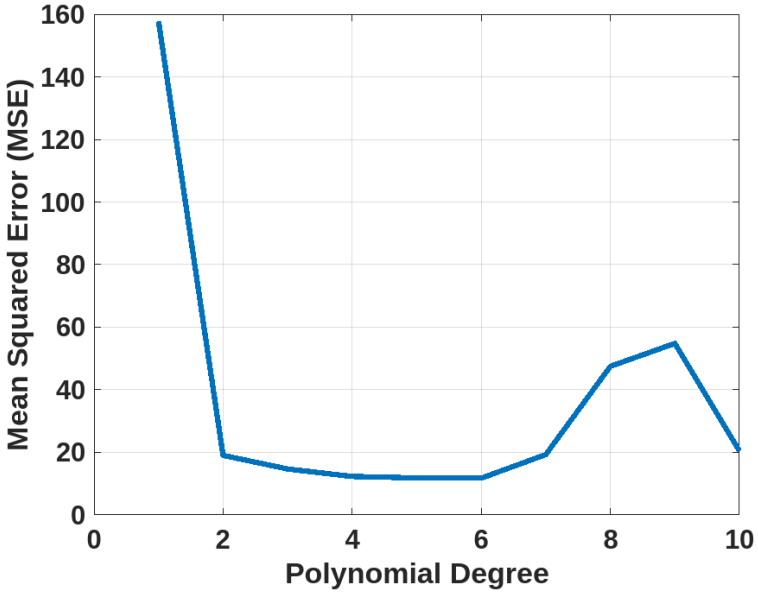


Figure 3. Plot between mean squared error (MSE) and polynomial degree.

Throughout the pre-processing stage, we handled significant challenges due to the complexity of our dataset using several key strategies. The challenge of a complex relationship between features such as weight with target variable pressure is handled through polynomial feature transformation. To determine the optimal degree, GridSearchCV and MSE are used, and degree 6 is found to be the optimal value that helps to setup the relationship between feature and target. However, since this generates 27 features, there was a risk of overfitting. Lasso Regularization is performed to select only the most significant features, which are discussed further in Section 4.2.1

4.2. Different Regression ML Models

Our objective is to predict pressure, a continuous variable, indicating the usage of regression modeling. In this section, we explore eight different regression models applied

to our dataset. Each model is chosen to address the specific aspects and requirements of our use case, incorporating various optimization techniques.

4.2.1. Linear Regression with Regularization

The initial model employed was linear regression. The objective of this algorithm is to identify the optimal line, plane, or hyperplane that minimizes the cost function, thereby reducing the discrepancy between the actual and predicted pressure values.

$$\hat{P} = \sum_{i=0}^{6} \sum_{j=0}^{6-i} \beta_{ij} A^i W^j + \epsilon \tag{1}$$

where $\hat{P}$ denotes predicted pressure; A and W represent angle and weight features, respectively; β denotes the coefficients of the polynomial terms with i and j to be the exponents of A and W, respectively, subject to the constraint $i + j \leq 6$; and ϵ represents the error term.

Given that the dataset has a complex relationship among its variables, as detailed in Section 4.1, a polynomial transformation of degree 6 is applied as shown in Equation (1). This transformation enabled the Linear Regression model to capture more complex patterns and achieve a better fit.

$$J(\beta) = \frac{1}{2n} \sum_{i=1}^{n} (P_i - \hat{P}_i)^2 + \lambda \sum_{j=1}^{27} |\beta_j| \tag{2}$$

where n represents the number of data points, P_i the actual pressure, and $\hat{P}_i$ the predicted pressure from Equation (1); λ denotes the Lasso regularization parameter that controls the strength of the penalty term; and $\sum_{j=1}^{27} |\beta_j|$ is the regularization term.

Here, Lasso regularization [22] is implemented to reduce the dimensionality of the feature space. This technique enabled the selection of significant features—A, W, AW, and W^6—by adding a penalty term proportional to the absolute value of coefficients. This approach drives coefficients to zero, reducing the cost function $J(\beta)$ (see Equation (2)) and mitigating multicollinearity and overfitting concerns.

4.2.2. K-Nearest Neighbor

From Section 4.2.1, the Linear Regression model requires an underlying relationship among features to achieve an optimal fit. In contrast, the K-Nearest Neighbors (KNN) algorithm [23], which does not assume a specific relationship between features and target variables, is ideal for capturing complex relationships, such as those between weight and pressure. KNN achieves this by identifying the nearest neighbors of a test data point within a training dataset and taking average of the result to predict pressure.

To optimize KNN performance, we determined the optimal value for k (the number of neighbors) to be five with the least Mean Absolute Error (MAE), as shown in Figure 4. This procedure is applied to cross-validated subsets of the dataset to mitigate the risk of overfitting during the selection of the optimal k value.

$$d((W_1, A_1), (W_2, A_2)) = \sqrt{(W_1 - W_2)^2 + (A_1 - A_2)^2} \tag{3}$$

where W_1 and A_1 represent the weight and angle of the specific training data point, respectively, and W_2 and A_2 represent the weight and angle of new test data point, respectively.

We computed the Euclidean distance in Equation (3) for each of the k neighbors and averaged these distances to obtain accurate pressure predictions for unseen data.

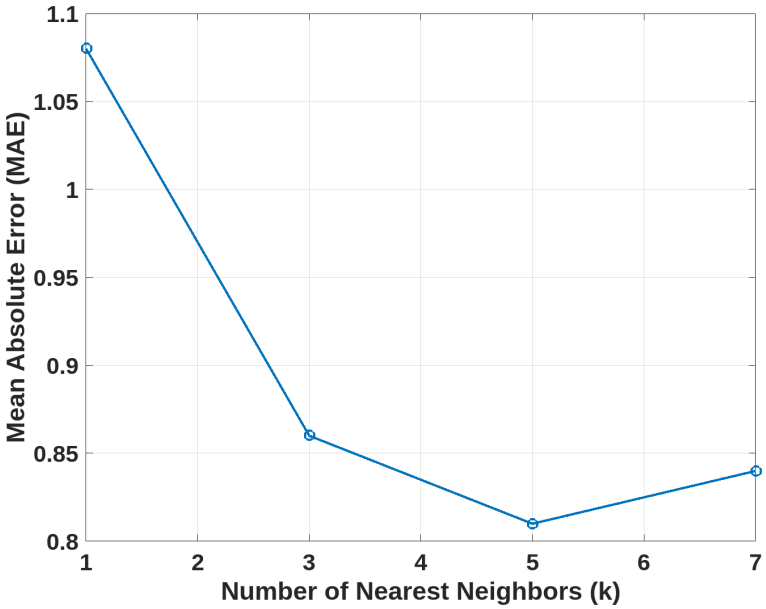


Figure 4. Plot between mean absolute error (MAE) and number of nearest neighbors (k).

4.2.3. Decision Tree

Another robust algorithm employed is the Decision Tree [24]. The decision rule for splitting the dataset is determined through a brute-force approach by systematically evaluating all possible threshold values to find the optimal split closest to the actual pressure values.

The optimal maximum tree depth is found to be 15 with the least MAE, as shown in Figure 5.

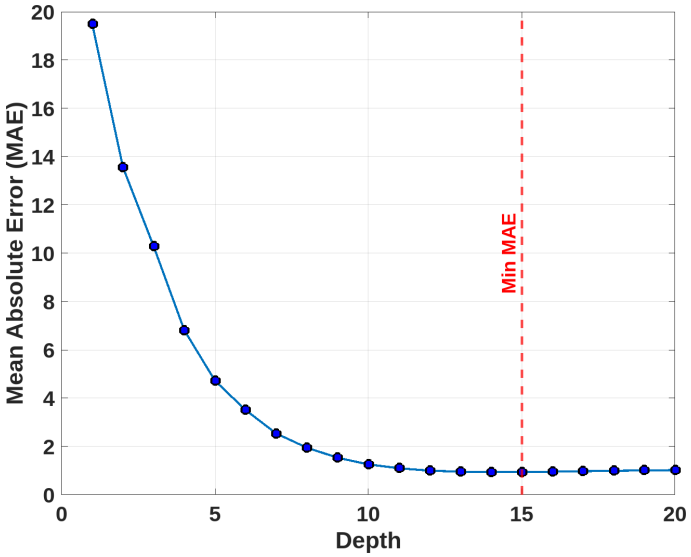


Figure 5. Plot between mean absolute error (MAE) and tree depth.

4.2.4. Random Forest

Due to the high dimensionality and complexity of our dataset, employing a simple decision tree approach carries a substantial risk of overfitting. To handle such scenarios, we employed an ensemble approach, namely, Random Forest [25]. This technique builds multiple smaller trees using random subsets of features, thereby reducing complexity and improving the model's exposure to a broader range of data points.

Similar to our approach with KNN, where cross-validation was used to improve model generalization, we utilized Out-Of-Bag (OOB) sampling [26] in Random Forests.

OOB sampling involves using the data points not included in each bootstrap sample to evaluate the model, offering a computationally efficient alternative to cross-validation or separate test datasets. We utilized the OOB score to optimize the hyperparameter $n_{\text{estimators}}$ (number of trees).

The optimal $n_{\text{estimators}}$ was identified as 300 that minimized the OOB score, as illustrated in Figure 6.

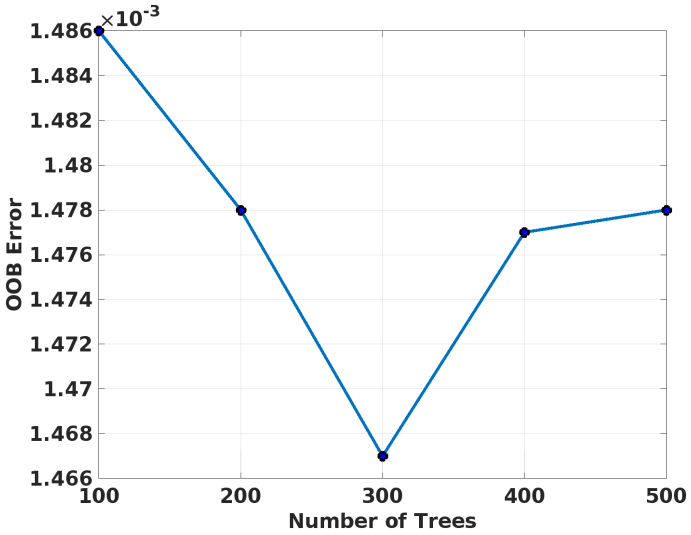


Figure 6. Plot between OOB error and number of trees ($n_{\text{estimators}}$).

4.2.5. Extra Trees

We implemented the Extra Trees or the Extremely Randomized Trees regressor [27], an advanced variant of the Random Forest algorithm, to explore its efficacy in machine learning tasks. Unlike the Random Forest, which requires significant computational time to determine optimal splits, Extra Trees improves the efficiency by randomizing the splits at each node and performing sampling without replacement. This randomness generates a diverse set of trees, which collectively enhances model performance and reduces computation time.

As illustrated in the Figure 7, the Out-Of-Bag (OOB) score reaches its minimum at $n_{\text{estimators}} = 400$.

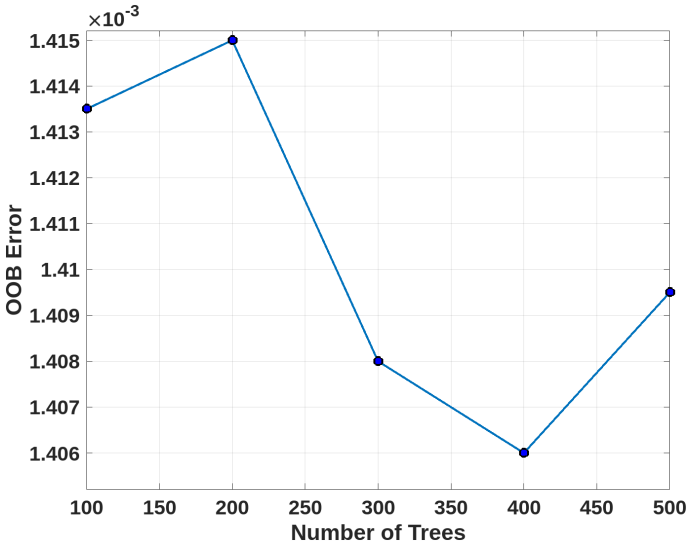


Figure 7. Plot between OOB error and number of trees ($n_{\text{estimators}}$).

4.2.6. Multi-Layer Perceptron

As observed from the ensembling techniques, we deduce that a large number of trees are needed to capture the internal complexities of the dataset. Artificial Neural Networks (ANNs) or Neural Networks are effective at addressing this issue [28]. Among ANNs, Multi-Layer Perceptrons (MLPs) are particularly suited for structured datasets with nonlinear dynamics [29], providing robust performance for such tasks.

In our application of pressure prediction, the MLP architecture includes:

The Input Layer: Receives data with two input neurons representing weight and angle.

Hidden Layers: Utilize the Rectified Linear Unit (ReLU) activation function, which handles nonlinearity and accommodates the broad range of pressure values (0 to 150 kPa). Each neuron performs a weighted sum of inputs followed by ReLU, enabling the network to capture complex patterns.

The Output Layer: Consists of a single neuron that generates the predicted pressure values based on the representations learned in the hidden layers.

Optimization is achieved through Adaptive Moment Estimation(Adam) [30], as it outperforms other methods by iteratively updating the network weights by leveraging adaptive learning rates and moment estimates of the training data.

Empirical testing determined that three hidden layers, each having seven hidden neurons, yielded the lowest error, as shown in Figure 8.

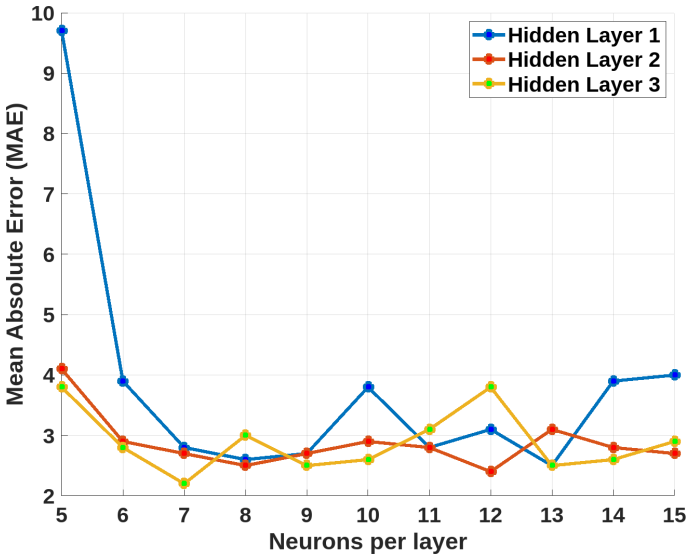


Figure 8. Plot between mean absolute error (MAE) and number of neurons for different hidden layers.

4.2.7. Kolmogorov–Arnold Network

A recent advancement in Neural Networks is the Kolmogorov–Arnold Network (KAN) [31], which extends the traditional MLP by incorporating learnable B-spline activation functions at each edge. Unlike MLPs, which utilize fixed activation functions, KAN's learnable activation functions allow for the capture of most of the complex interactions between variables, a feature commonly found in soft robotic applications [32]. KAN is also more parameter-efficient than MLP, as the use of B-splines allows for smoother function approximation with fewer parameters. The KAN's grid system adapts dynamically to the data distribution, further minimizing the need for additional parameters. Our experiments revealed that a pressure prediction model with five hidden layers and tree cubic splines yielded the lowest error. Training was conducted using the Adam optimizer, chosen for its effectiveness on large datasets.

$$\hat{P} = 0.2874A - 0.1541W + 95.9605 \quad (4)$$

where $\hat{P}$ is predicted pressure, A is the input angle, and W is the input weight.

Additionally, KAN's auto_symbolic feature was employed to convert the trained KAN into a symbolic formula (see Equation (4)). This conversion approximates the learned spline functions by suggesting symbolic candidates. As a result, it enhances model interpretability, a feature often lacking in traditional MLPs.

4.2.8. KNN-Linear Regression Hybrid

To enhance the performance and optimize the results, we implemented a novel hybrid model that combines the strengths of KNN and Linear Regression. This approach integrates the benefits of both algorithms to improve predictive accuracy.

In our hybrid model, the learning phase resembles the traditional KNN method. Initially, the model retrieves the relevant input features, such as weight and angle, and identifies the closest neighbors by calculating distances. While the standard KNN approach typically predicts by averaging the outcomes of these neighbors, our method diverges by using the neighbors' data as inputs for a Linear Regression model, which will be trained to yield the slope for the plane.

As shown in Figure 9, this hybrid approach allows the Linear Regression model to generate a more accurate fit based on the nearest neighbors.

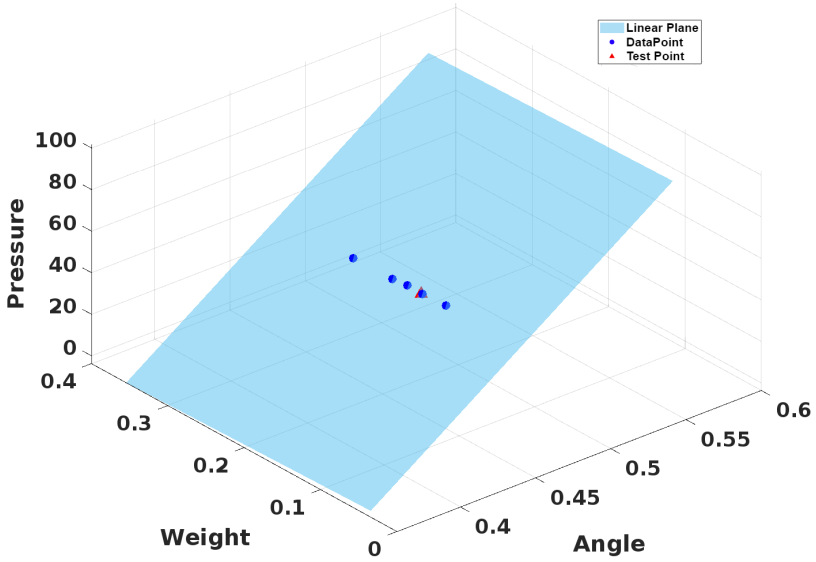


Figure 9. Linear Regression fit with K Nearest Neighbors.

5. Results and Discussion

In this section, we present a comparative analysis of the performance of the various ML models developed using the training dataset collected from the soft robotic elbow exoskeleton set-up. This comparison will aid in selecting the most appropriate model for actuator pressure prediction. Table 1 summarizes the evaluation metrics used to assess the efficacy of each model. The metrics include the MAE, MSE, and percentage accuracy, all of which were calculated based on the performance of the models on the test dataset.

Additionally, we evaluated the training time for each model, as it is crucial for managing memory usage and computational costs.

To ensure the robustness and reliability of our results, we further validated each machine learning model by evaluating their performance on unseen data. This step helps assess how well the models generalize to new, unobserved samples and how closely their predictions align with actual outcomes. Table 2 presents the results of these predictions on the experimental dataset.

Table 1. Various ML model comparisons based on various evaluation metrics.

Model	MAE	MSE	Accuracy (%)	Training Computation Time (s)
Linear	2.2177	8.7314	99.44	14
KNN	0.8493	2.0976	99.92	11
Decision Tree	0.9496	2.3065	99.85	11
Random Forest	0.9121	2.9392	99.82	28
Extra Trees	0.8842	2.8250	99.81	16
MLP	2.0971	5.3270	99.35	60
KAN	32.1900	38.9000	88.50	300
KNN-Linear	0.8434	1.6950	99.89	21

Table 2. Various ML model comparison based on actual vs. predicted pressure using experimental data.

Actual Pressure (kPa)	Linear Regression Prediction (kPa)	KNN Prediction (kPa)	Decision Tree Prediction (kPa)	Random Forest Prediction (kPa)	Extra Trees Prediction (kPa)	MLP Prediction (kPa)	KAN Prediction (kPa)	KNN-Linear Prediction (kPa)
77.08	74.51	77.28	77.41	77.50	77.27	79.80	81.36	77.18
113.25	111.21	112.85	112.69	112.80	112.86	113.99	96.22	112.84
122.04	123.59	121.69	122.04	121.81	121.79	122.94	95.09	121.73
135.24	137.10	135.73	135.73	135.81	135.69	133.99	144.95	135.79
136.71	136.17	136.64	136.08	136.82	136.79	137.26	131.89	136.53
141.59	142.74	141.94	142.57	142.41	142.28	141.70	169.33	142.35
143.55	142.92	142.86	142.82	142.76	142.57	142.46	174.10	142.95
147.95	146.75	147.54	148.44	148.28	148.25	143.90	151.29	147.78
148.92	147.74	148.93	148.92	148.76	148.75	149.42	142.47	148.85
149.41	148.22	149.06	148.92	148.92	148.96	151.53	153.9	149.26

Ten distinct points are selected from the experimental data, each corresponding to different combinations of weights and angles. This was fed as input to the eight machine learning models developed for this study. The predicted pressure values from each model are recorded and compared to the actual pressure measurements, as shown in Table 2.

Table 3 presents a summary of the results, where the average absolute error for the predictions made by each machine learning model is computed. As shown, the KNN-Linear hybrid model demonstrated the best performance, achieving the lowest error of 0.330, which facilitated an additional level of comprehensive evaluation of the model.

The results compare various machine learning models based on multiple evaluation parameters, allowing users to select the most suitable model for the soft robotic application based on their specific requirements. For instance, the model with the lowest mean squared error (MSE) may be chosen for tasks where accuracy is critical. However, if the application requires real-time implementation, computational efficiency must also be considered. Some models perform poorly in both accuracy and computational efficiency. These models can be directly excluded from consideration as they are not practical or worthwhile for the soft robotic application.

Table 3. Various ML model comparisons based on average absolute error using experimental data.

Machine Learning Model	Average Absolute Error
Linear Regression	1.391
KNN	0.332
Decision Tree	0.470
Random Forest	0.437
Extra Trees	0.395
MLP	1.403
KAN	13.536
KNN-Linear	0.330

6. Conclusions

This study presents a comprehensive analysis of machine learning models for predicting the pressure required to operate a soft robotic elbow exoskeleton. The primary objective is to develop a reliable and efficient model to assist occupational workers in reducing musculoskeletal disorders by offloading a portion of the load on the elbow joint. Various machine learning models are applied and evaluated based on their accuracy and computational time. Additionally, a novel hybrid machine learning approach is also proposed by combining KNN and a linear regression model. The results indicate that the KNN-Linear Regression hybrid model outperformed other models, achieving the lowest MAE and MSE and demonstrating superior predictive accuracy and computational efficiency. This hybrid approach effectively combines the strengths of KNN's local prediction capabilities with the linear regression model's ability to fit complex relationships, making it particularly suitable for the nonlinear dynamics of soft robotic systems. Future work will focus on further refining these models, exploring additional machine learning techniques, and conducting real-world trials to validate the practical applicability of the proposed solutions.

Author Contributions: Conceptualization, I.S. and S.S.; methodology, S.S. and I.S.; software, S.S.; validation, S.S. and I.S.; formal analysis, S.S. and I.S.; investigation, S.S. and I.S.; writing—original draft preparation, S.S. and I.S.; writing—review and editing, S.S. and I.S.; and supervision, M.B.J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: We want to thank Khoa Pham for his support and contribution during data collection.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Triwiyanto, T.; Yulianto, E.; Mak'ruf, M.R.; Titisari, D.; Rahmawati, T.; Luthfiyah, S.; Hamzah, T.; Syaifudin, S.; Wisana, I.D.G.H. A Review on Robotic Hand Exoskeleton Devices: State-of-the-Art Method. In Proceedings of the 1st International Conference on Electronics, Biomedical Engineering, and Health Informatics: ICEBEHI , Surabaya, Indonesia, 8–9 October 2020; Springer: Singapore, 2021; pp. 331–341.
2. Chen, J.; Yu, Z.; Zuo, Z.; Yan, G.; Cheng, Y. Human Intention Recognition for Lower Limb Exoskeleton Robot. In *Chinese Intelligent Systems Conference*; Springer Nature: Singapore, 2023; pp. 353–368.
3. Inoue, Y.; Kuroda, Y.; Yamanoi, Y.; Yabuki, Y.; Yokoi, H. Development of Wrist Separated Exoskeleton Socket of Myoelectric Prosthesis Hand for Symbrachydactyly. *Cyborg Bionic Syst.* **2024**, *5*, 0141. [CrossRef] [PubMed]
4. Liang, F.; Mo, L.; Sun, Y.; Guo, C.; Gao, F.; Liao, W.H.; Cao, J.; Li, B.; Song, Z.; Wang, D.; et al. Interlimb and Intralimb Synergy Modeling for Lower Limb Assistive Devices: Modeling Methods and Feature Selection. Cyborg and Bionic *Systems* **2024**, *5*, 0122. [CrossRef]
5. Yu, F.; Liu, Y.; Wu, Z.; Tan, M.; Yu, J. Adaptive gait training of a lower limb rehabilitation robot based on human–robot interaction force measurement. *Cyborg Bionic Syst.* **2024**, *5*, 0115. [CrossRef]
6. Kim, D.; Kim, S.-H.; Kim, T.; Kang, B.B.; Lee, M.; Park, W.; Ku, S.; Kim, D.; Kwon, J.; Lee, H.; et al. Review of machine learning methods in soft robotics. *PLoS ONE* **2021**, *16*, e0246102. [CrossRef] [PubMed]
7. Chin, K.; Hellebrekers, T.; Majidi, C. Machine learning for soft robotic sensing and control. *Adv. Intell. Syst.* **2020**, *2*, 1900171. [CrossRef]
8. Ben-Tzvi, P.; Danoff, J.; Ma, Z. The design evolution of a sensing and force-feedback exoskeleton robotic glove for hand rehabilitation application. *J. Mech. Robot.* **2016**, *8*, 051019. [CrossRef]

9. Laksono, P.W.; Kitamura, T.; Muguro, J.; Matsushita, K.; Sasaki, M.; Amri bin Suhaimi, M.S. Minimum mapping from EMG signals at human elbow and shoulder movements into two DoF upper-limb robot with machine learning. *Machines* **2021**, *9*, 56. [CrossRef]
10. Chen, X.; Gong, L.; Zheng, L.; Zou, Z. Soft exoskeleton glove for hand assistance based on human-machine interaction and machine learning. In Proceedings of the 2020 IEEE International Conference on Human-Machine Systems (ICHMS), Rome, Italy, 7–9 September 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
11. Jung, J.Y.; Heo, W.; Yang, H.; Park, H. A neural network-based gait phase classification method using sensors equipped on lower limb exoskeleton robots. *Sensors* **2015**, *15*, 27738–27759. [CrossRef] [PubMed]
12. Ng, Y.L.; Jiang, X.; Zhang, Y.; Shin, S.B.; Ning, R. Automated Activity Recognition with Gait Positions Using Machine Learning Algorithms. *Eng. Technol. Appl. Sci. Res.* **2019**, *9*, 4. [CrossRef]
13. Thakur, N.; Han, C.Y. Exoskeleton-based multimodal action and movement recognition: Identifying and developing the optimal boosted learning approach. *arXiv* **2021**, arXiv:2106.10331. [CrossRef]
14. Zheng, Y.; Song, Q.; Liu, J.; Song, Q.; Yue, Q. Research on motion pattern recognition of exoskeleton robot based on multimodal machine learning model. *Neural Comput. Appl.* **2020**, *32*, 1869–1877. [CrossRef]
15. Lee, T.; Kim, I.; Lee, S.H. Estimation of the continuous walking angle of knee and ankle (talocrural joint, subtalar joint) of a lower-limb exoskeleton robot using a neural network. *Sensors* **2021**, *21*, 2807. [CrossRef] [PubMed]
16. Chen, Y.; Yang, Z.; Wen, Y. A soft exoskeleton glove for hand bilateral training via surface emg. *Sensors* **2021**, *21*, 578. [CrossRef]
17. Xiao, Z.G.; Elnady, A.M.; Menon, C. Control an exoskeleton for forearm rotation using FMG. In Proceedings of the 5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics, Sao Paulo, Brazil, 12–15 August 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 591–596.
18. Sierotowicz, M.; Lotti, N.; Nell, L.; Missiroli, F.; Alicea, R.; Zhang, X.; Xiloyannis, M.; Rupp, R.; Papp, E.; Krzywinski, J.; et al. EMG-driven machine learning control of a soft glove for grasping assistance and rehabilitation. *IEEE Robot. Autom. Lett.* **2022**, *7*, 1566–1573. [CrossRef]
19. Erel, V.; Singh, I.; Lindsay, A.R.; Shi, W.Y.; Wijesundara, M.B. Development and Characterization of a Modular Soft Actuator Enabled Elbow Exoskeleton for Assistive Movements. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Virtual, 17–20 August 2021; American Society of Mechanical Engineers: New York, NY, USA, 2021; Volume 85444, p. V08AT08A046.
20. Subasi, A. *Practical Machine Learning for Data Analysis Using Python*; Academic Press: Cambridge, MA, USA, 2020.
21. Owen, L. *Hyperparameter Tuning with Python: Boost your Machine Learning Model's Performance via Hyperparameter Tuning*; Packt Publishing Ltd.: Birmingham, UK, 2022.
22. Muthukrishnan, R.; Rohini, R. LASSO: A feature selection technique in predictive modeling for machine learning. In Proceedings of the 2016 IEEE International Conference on Advances in Computer Applications (ICACA), Coimbatore, India, 24 October 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 18–20.
23. Hackeling, G. *Mastering Machine Learning with Scikit-Learn*; Packt Publishing Ltd.: Birmingham, UK, 2017.
24. James, G.; Witten, D.; Hastie, T.; Tibshirani, R.; Taylor, J. Tree-based methods. In *An Introduction to Statistical Learning: With Applications in Python*; Springer International Publishing: Cham, Switzerland, 2023; pp. 331–366.
25. Salam, M.A.; Azar, A.T.; Elgendy, M.S.; Fouad, K.M. The effect of different dimensionality reduction techniques on machine learning overfitting problem. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 641–655. [CrossRef]
26. Altman, N.; Krzywinski, M. Ensemble methods: Bagging and random forests. *Nat. Methods* **2017**, *14*, 933–935. [CrossRef]
27. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [CrossRef]
28. Dongare, A.D.; Kharde, R.R.; Kachare, A.D. Introduction to artificial neural network. *Int. J. Eng. Innov. Technol. (IJEIT)* **2012**, *2*, 189–194.
29. Chen, S.A.B.S.; Billings, S.A. Neural networks for nonlinear dynamic system modelling and identification. *Int. J. Control.* **1992**, *56*, 319–346. [CrossRef]
30. Kingma, D.P. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
31. Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T.Y.; Tegmark, M. Kan: Kolmogorov–Arnold networks. *arXiv* **2024**, arXiv:2404.19756.
32. Peng, Y.; He, M.; Hu, F.; Mao, Z.; Huang, X.; Ding, J. Predictive Modeling of Flexible EHD Pumps using Kolmogorov–Arnold Networks. *arXiv* **2024**, arXiv:2405.07488. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.