Least Squares Boosting Ensemble and Quantum-Behaved Particle Swarm Optimization for Predicting the Surface Roughness in Face Milling Process of Aluminum Material

Mahdi S. Alajmi 1,∗ and Abdullah M. Almeshal 2

Abstract: Surface roughness is a significant factor in determining the product quality and highly impacts the production price. The ability to predict the surface roughness before production would save the time and resources of the process. This research investigated the performance of state-of-the-art machine learning and quantum behaved evolutionary computation methods in predicting the surface roughness of aluminum material in a face-milling machine. Quantum-behaved particle swarm optimization (QPSO) and least squares gradient boosting ensemble (LSBoost) were utilized to simulate numerous face milling experiments and have predicted the surface roughness values with high extent of accuracy. The algorithms have shown a superior prediction performance over genetics optimization algorithm (GA) and the classical particle swarm optimization (PSO) in terms of statistical performance indicators. The QPSO outperformed all the simulated algorithms with a root mean square error of RMSE = 2.17% and a coefficient of determination $R^2 = 0.95$ that closely matches the actual surface roughness experimental values.

Keywords: machine learning; optimization; quantum computing; face milling; surface roughness; ensemble learning

1. Introduction

Machining is the most significant features of any production action. Involving several machining procedures, milling is broadly utilized procedure to create compound geometries in several applications of dies as well as molds, turbine rotors, etc. [1]. However, the milling procedure is the most commonly utilized machining procedure in manufacturing sector. The major aim of this process is to produce high-quality parts within reasonable period of time with high surface quality. Surface roughness of any machining procedure has become essential due to the increased quality demands, and still there are probabilities of refusing the element for the absence of essential surface finish, even if the component’s dimensions are good in the dimensional tolerance. It is a significant measure of the product’s quality along with also significantly effects the production price and mostly dependent on numerous parameters for example cutting speed, tool nomenclature, feed, cutting force, inflexibility and depth of cut of the machine [2]. The conventional method in choosing the machining parameters is based on trial and error and the expert knowledge using the machining handbooks. This method is time consuming and has an exhausting procedure. A human procedure organizer chooses appropriate machining procedure parameters by employing his/her own experience or machining tables. In many cases, the preferred parameters are conventional, as well as away from optimum.

Therefore, it becomes essential to introduce a vigorous method to forecast the machining parameters prior to machining to acquire the required product’s surface roughness.
There are numerous researches outlining intelligent methods that were considered for the optimization of the machining procedure: for example, applied quadratic programming [4], nonlinear programming [5], sequential programming [6], goal programming [7] and dynamic programming [8], to resolve the issues by developing it as a multi-objective function model. The optimization issue was additionally resolved through introducing several non-traditional optimization approaches, which include genetic algorithms (GAs) [9,10], particle swarm optimization (PSO) [11,12], scatter search (SS) [13,14], ant colony optimization (ACO) [15,16], fuzzy-logic-based expert systems [17] and differential evolution (DE) [18]. Hybrid optimization approach built on difference evolution algorithm, as well as Taguchi’s method, is established by Yildiz [19]. The use of hybrid strategy is utilized in the studies to optimize machining parameters in multipass turning activities. Rao et al. [20] proposed an advanced algorithm known as teaching-learning-based optimization (TLBO) algorithm, that was utilized for the machining parameters optimization of preferred modern machining procedures. Rao et al. [21] presented the MO-Jaya (Multi-objective Jaya) algorithm to optimize the abrasive waterjet machining process and compared the outcomes with well-known optimization algorithms. Alajmi et al. [22] introduced quantum based optimization method (QBOM) to resolve the surface-grinding-process optimization. Quantum-based optimization method’s performance is examined against two tests; the first is a rough grinding procedure and the second is a finished grinding procedure. Yildiz and Solanki [23] used the latest hybrid optimization technique (HPSI), according to the particle swarm algorithm, as well as receptor modifying the immune system’s property for multi-objective crashworthiness optimization of a full vehicle model along with milling optimization issues. Baraheni and Amini [24] utilized ANOVA (analysis of variance) to figure out the effect of drilling, as well as material parameters comprising cutting velocity, feed rate, ultrasonic vibration and plate thickness on thrust force, along with delamination. Bustillo et al. [25] proposed a strategy to avoid the limitations of high experimental costs in relative to dataset size and to accomplish the smart manufacturing patterns of the friction-drilling process. The extension of the introduced methodology to other datasets from the friction drilling process will help to detect the most accurate machine learning algorithm for this industrial task. The results on this dataset showed that the AdaBoost ensembles offered the highest accuracy and were more easily optimized than artificial neural networks. Sanchez et al. [26] used a hybrid computer-integrated system for the development of the accuracy of corner cutting that combines experimental knowledge of the process and numerical simulation. The system accepts the user to choose the optimum cutting strategy, either by wire path modification or by cutting regime modification (when high accuracy is required). The validity of the system has been verified through a series of case studies, which demonstrate the improvements in accuracy and productivity with respect to the normally used strategies.

This research contributes in assessing the accuracy of state-of-the-art machine learning and quantum evolutionary optimization algorithm in predicting the surface roughness values in a face-milling process. Ensemble learning based on least squares gradient boosting (LSBoost) and quantum-behaved particle swarm optimization (QPSO) are investigated and assessed in terms of statistical performance indicators, to predict the surface roughness with a high extent of accuracy. Such prediction would allow the process operator to save time and labor and achieve the required production quality.

1.1. Face-Milling Mathematical Model

The mathematical model is expressed according to the machining parameter’s influence on the workpiece’s surface roughness. The machining parameters involved in the process are accomplished from the data of the machine limitations. The mathematical model of the milling procedure offered in the literature [2] is used in the present study.
1.2. Objective Function

The purpose of prediction the machining parameters in the face-milling procedure is to attain the preferred workpiece’s surface-roughness values from least machining time. According to the machining parameter’s influence on the reactions, the surface-roughness equations were expressed as demonstrated in Equations (1) and (2).

For \(1000 \ll v \ll 3000 \text{ rpm}, \quad R_a = 0.7675(v^{-0.562}f^{0.630}a^{0.530})\) \hspace{1cm} (1)

For \(3000 \ll v \ll 4000 \text{ rpm}, \quad R_a = 1.7918(v^{-0.562}f^{0.630}a^{0.530})\) \hspace{1cm} (2)

Subsequently the measured cutting speed’s range is quite broad, so a single equation might not suit the forecast surface roughness for the whole series. Therefore, Equations (1) and (2) were expressed to fit every situation by minimal errors. The experimental constants in Equations (1) and (2) can be discovered, depending on the machining parameter’s influence on machining time, as well as surface roughness. The ordinary equations for certain machining time are provided in Equations (3) and (4).

\[
T_m = \frac{y + L + \Delta}{f} \hspace{1cm} (3)
\]

\[
y = 0.5 \left( a^2 - \sqrt{a^2 - B^2} \right) \hspace{1cm} (4)
\]

where \(\Delta\) is over travel = 2.5 mm, \(L\) denotes the milled surface’s length in mm and \(Y\) denotes the cutter approach length in mm.

1.3. Constraints

There are numerous constraints which occur in the real machining status for the optimization of the objective purpose. A specified depth of cut and feed rate, along with cutting speed, are selected, and consideration is given to the struggle among the surface roughness and machining time. The boundaries of these parameters are presented in Table 1 and are examined for optimizing the machining parameters and defined as follows.

| Parameter | Minimum Value | Maximum Value |
|-----------|---------------|---------------|
| \(v\) | 1000 rpm | 3000 rpm |
| \(f\) | 180 mm/min | 300 mm/min |
| \(a\) | 0.2 mm | 0.6 mm |

2. Methodology

2.1. Particle Swarm Optimization Algorithm (PSO)

Particle swarm optimization (PSO) algorithm is a population-based swarm intelligence optimization algorithm that was introduced by Kennedy and Eberhart [27] and has been widely used by scientists in various applications. PSO algorithm was inspired by the flock of birds and fish schools. When compared to other evolutionary algorithms, such as bacterial foraging algorithm (BFA), it can be observed that PSO is simpler to implement and more computationally efficient in converging to optimal solutions [28]. Moreover, PSO is robust to control parameters and easy to implement in various software packages [29].

The implementation of PSO algorithm starts by distributing a population of particles within the defined search space. These particles move towards the optimal solution by computing the objective function at each iteration. Each particle then computed the personal best (pbest) solution and was then compared with all the particles. The global best (gbest) solution can then be determined at each iteration until the termination criterion is met and the optimal solution is defined.
The position and velocity of each particle is computed and updated at each iteration, according to specific equations that corresponds to the values of personal best and global best solutions. The following are the set of equations that correspond to each step in the pseudocode of Algorithm 1 of the classical PSO algorithm. In Step 1, let the position of the $i$-th particle in the $d$-dimensional search space be represented as follows:

$$X_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^T$$

Similarly, the velocity array can be represented as follows:

$$V_i = [v_{i1}, v_{i2}, \ldots, v_{id}]^T$$

During each iteration, the fitness of each particle is evaluated and compared with its personal best array of solutions; if the current fitness is better than the previous fitness values, the current personal best solution is updated with the new fitness value and added to the array of personal best solutions, which can be defined as follows:

$$P_i = [p_{i1}, p_{i2}, \ldots, p_{id}]^T$$

In Step 4 of the PSO pseudocode, the global best solution is determined at each iteration among all the fitness values of the whole population, and the array of global best solutions is defined as follows:

$$P_g = [p_{g1}, p_{g2}, \ldots, p_{gd}]^T$$

The particle’s velocity and position are updated in Equations (9) and (10), respectively, as follows:

$$V_i(t + 1) = V_i(t) + c_1 r_1 (P_i - X_i(t)) + c_2 r_2 (P_g - X_i(t))$$

$$X_i(t + 1) = X_i(t) + \Delta t V_i(t + 1)$$

Algorithm 1 Classical PSO algorithm

| Step 1: Setting population size and random initialisation of particle positions and velocities. |
| Step 2: Evaluation of particles fitness according to required objective function |
| Step 3: Evaluation of personal best solution of each particle |
| Step 4: Evaluation of global best solution |
| Step 5: Velocity update |
| Step 6: Position update |
| Step 7: Repeat Steps 2-6 until termination criteria met. |

The PSO algorithm will be terminated once the termination criterion is met; whether it is a maximization or minimization problem, the terminal criterion is usually the number of iterations that is defined by the user. The selection of PSO control parameters is a critical step that can negatively impact the optimization process and may result in premature convergence and narrow down the search diversity and local maxima traps [29]. Several studies, such as in References [29–31], have investigated the selection of PSO parameters in terms of the population size, social coefficient, cognitive coefficient, maximum particle velocity and the maximum iterations.

2.2. Quantum-Behaved Particle Swarm Optimization Algorithm (QPSO)

In the literature, there exist many variations of PSO, such as quantum-behaved PSO, deep learning driven-PSO, hybrid PSO-Bacterial foraging optimization (BFO) and PSO-Grey Wolf Optimizer (GWO) hybrid approach [32–35]. PSO variations aimed at providing faster convergence rates, avoiding local maxima traps and parameter selection rules for better performance of the PSO. In this paper, we focus on utilizing the quantum-behaved PSO
(QPSO) presented by Sun et al. [36] that has fewer parameters to control and outperforms the original PSO algorithm.

In classical PSO, the particles’ positions and velocities are calculated at each iteration step and updated to diversify the search space and converge towards the optimal solution. Thus, the trajectory of the movement of particles within the search space is deterministic. However, in quantum mechanics, and according to Heisenberg’s uncertainty principle, the velocity and the position of the particle cannot be determined simultaneously [37] and the state of the particle is described by Schrödinger’s wave function \( \psi(x,t) \). By solving Schrödinger’s equation to get the probability density function of the particles location in the space, and using Monte Carlo simulation, the position of movement of the particle can be presented as follows:

\[
\begin{align*}
    x_{i,j}(t+1) &= p_i(t) + \beta \left| M_{best,j}(t) - x_{i,j}(t) \right| \ln \left( \frac{1}{u} \right) \quad \text{if } k \geq 0.5 \\
    x_{i,j}(t+1) &= p_i(t) - \beta \left| M_{best,j}(t) - x_{i,j}(t) \right| \ln \left( \frac{1}{u} \right) \quad \text{if } k < 0.5
\end{align*}
\]  

(11)

where we have the following:

\( M_{best,j}(t) \) is the Mainstream thought or mean best value, \( x_{i,j}(t+1) \) the position of the \( i \)-th particle in the \( j \)-th dimension of the space, \( u \) and \( k \) are the uniform probability distribution parameters in the range \([0,1]\), \( \beta \) denotes the contraction-expansion coefficient and the \( p_i \) as the local attractor point.

The mean optimal value is described as the mean of each personal greatest of the population and can be evaluated as follows:

\[
M_{best,j}(t) = \frac{1}{N} \sum_{j=1}^{N} p_{g,j}(t)
\]  

(12)

In which \( g \) represents the best particle’s index in the population. In addition, the local attractor, \( p_i \), guarantees the convergence of the algorithm [37] and is defined as follows:

\[
p_i(t) = \frac{c_1 p_{k,i} + c_2 p_{g,i}}{c_1 + c_2}
\]  

(13)

where \( p_{k,i} \) and \( p_{g,i} \) represent the \( p_{best} \) and \( g_{best} \), respectively. The pseudocode of the QPSO algorithm is as presented in Algorithm 2.

### Algorithm 2 QPSO algorithm

1. Setting population size and random initialisation of particle positions and velocities.
2. Evaluation of particles fitness according to required objective function
3. Evaluation of personal best solution of each particle
4. Evaluation of global best solution
5. Calculating of Mean best (\( M_{best} \)) of all the \( p_{best} \) of the population
6. Position update
7. Repeat Steps 2–6 until termination criteria met.

2.3. Least Squares Boosting Ensemble (LSBoost)

The gradient boosting ensemble method consists of a finite set of weak learners and a meta learner that assigns weights to each learner and combines the predictive results of each one, using voting methods to provide a better predictive performance for regression problems. Boosting is a supervised machine learning method where the data are split into training, validation and testing sets. The algorithm starts by training the individual weak learners, sequentially, that are in the form of decision trees, and fits the residual of errors to achieve a better performance. The LSBoost method utilizes the least squares as the loss criteria. The LSBoost pseudocode is presented in Algorithm 3, as presented by Friedman [38].
Algorithm 3 LSBoost Algorithm

Define \( x_i \) and \( y_i \) as explainable variables and \( M \) as the number of iterations

Define the training set \( \{(x_i,y_i)\}_{i=1}^n \), a loss function as \( L(y,F) = \frac{(y-F)^2}{2} \) and \( F_m(x) \) as the regression function.

Initialization: \( F_0(x) = \bar{y} \)

For \( m = 1 \) to \( M \) do:

\( \tilde{y}_i = y_i - F_{m-1}(x_i) \) for \( i = 1,2,\ldots,N \)

\( (\rho_m,\alpha_m) = \arg\min_{\rho,\alpha} \sum_{i=1}^{N} [\tilde{y}_i - \rho h(x_i;\alpha)]^2 \)

\[ F_m(x) = F_{m-1}(x) + \rho_m h(x;\alpha_m) \]

End

To assess the prediction accuracy, various performance indicators are utilized as the root mean square error (RMSE), mean absolute error (MAE), coefficient of variation of root mean square error (CVRMSE), mean absolute percentage error (MAPE) and the coefficient of determination, \( R^2 \), as defined in Equations (14)–(18) respectively.

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}}
\]  

(14)

\[
MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|
\]  

(15)

\[
CVRMSE = \sqrt{\frac{\sum_{i=1}^{n} (\tilde{y}_i - y_i)^2}{\bar{y}}}
\]  

(16)

\[
MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right| \times 100\%
\]  

(17)

\[
R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}
\]  

(18)

3. Results

The QPSO algorithm was executed in the MATLAB software package, to optimize the machining time for a desired surface roughness of aluminum material in the milling-machine system introduced previously. The QPSO parameters utilized in the simulation are presented in Table 2 and were selected based on heuristic tuning and setting the cognitive acceleration and social coefficients with equal values, as recommended by some studies in the literature [27,39,40]. Meanwhile, the parameters of the LSBoost algorithm are presented in Table 3. In addition, the simulation parameters used in Matlab for the PSO and GA algorithms are presented in Tables 4 and 5, respectively.

Table 2. QPSO parameters.

| Parameter                  | Value |
|----------------------------|-------|
| Iterations                 | 1000  |
| Particles population       | 100   |
| Cognitive acceleration: \( c_1 \) | 2     |
| Social coefficient: \( c_2 \) | 2     |
| Contraction expansion factor: \( \beta \) | 0.8   |
Table 3. LSBoost parameters.

| Parameter          | Value |
|--------------------|-------|
| Number of learners | 50    |
| Learning rate      | 0.01  |
| Minimum leaf size  | 1     |

Table 4. PSO parameters.

| Parameter           | Value |
|---------------------|-------|
| Iterations          | 1000  |
| Particles population| 100   |
| Cognitive acceleration: $c_1$ | 2     |
| Social coefficient: $c_2$    | 2     |

Table 5. GA parameters.

| Parameter                        | Value |
|----------------------------------|-------|
| Maximum number of generation     | 100   |
| Number of individuals per generation | 25   |
| Generation gap                   | 0.5   |
| Crossover                        | 0.7   |
| Mutation rate                    | 0.04  |

A total of 36 pilot experiments were executed, to confirm the computational results of the QPSO and LSBoost algorithms, and compared with the performance of PSO and GA algorithms, which have been extensively reported in the literature, for the milling machine to accomplish the required surface roughness with the minimum machining time, as presented in Table 6.

The results obtained by the QPSO for the machining parameters show the ability of the QPSO algorithm to achieve minimum machining time in comparison to the experimental results for the required surface roughness with a high extent of accuracy. Table 7 provides the statistical performance indicators of each algorithm in terms of the MAPE, RMSE, MAE and $R^2$ values. It can be noted that QPSO has achieved the best accuracy in predicting the surface roughness values with RMSE of 2.17% and a high coefficient of determination value of $R^2 = 0.95$. The high value of $R^2$ clearly indicates that the predicted values closely match the actual experimental values of the surface roughness and is evident of the superior performance of the QPSO, due to the algorithm’s superiority in exploring the search space and avoiding local optima trap. The LSBoost provides the second best prediction values with an RMSE of 3.74% and $R^2$ of 0.88. This can be addressed to the gradient ensemble in combining different weak learners into a meta learner that provide the best prediction at each step. The GA algorithm resulted in a relatively comparable performance with LSBoost algorithm in terms of the coefficient of determination with $R^2 = 0.871$. However, it has a slightly higher RMSE value of 4.86%, which is higher than the LSBoost algorithm by 1.12%. On the other hand, the PSO algorithm resulted in an RMSE of 4.99% and the lowest $R^2$ value of 0.84.
Table 6. Simulation results of QPSO, LSBoost, PSO and GA algorithms for the prediction of surface roughness values of a face milling machine.

| Exp No. | \( f \) (mm/min) | \( a \) (mm) | \( t \) (s) | \( v \) (rpm) | \( R_a \) (µm) | QPSO | LSBOOST | PSO | GA |
|---------|----------------|-------------|-------------|-------------|--------------|-------|---------|-----|-----|
| 1       | 196.37         | 0.22        | 37          | 1672.72     | 0.18         | 0.19  | 0.23    | 0.19 |
| 2       | 238.15         | 0.22        | 31          | 1861.34     | 0.19         | 0.22  | 0.24    | 0.19 |
| 3       | 233.31         | 0.23        | 31          | 1307.27     | 0.23         | 0.22  | 0.27    | 0.25 |
| 4       | 236.51         | 0.43        | 31          | 1832.82     | 0.26         | 0.24  | 0.28    | 0.27 |
| 5       | 272.41         | 0.41        | 27          | 1982.63     | 0.28         | 0.30  | 0.29    | 0.29 |
| 6       | 260.54         | 0.34        | 28          | 1120.73     | 0.32         | 0.38  | 0.34    | 0.35 |
| 7       | 285.91         | 0.42        | 25          | 1405.22     | 0.35         | 0.34  | 0.36    | 0.39 |
| 8       | 275.87         | 0.28        | 26          | 3136.03     | 0.39         | 0.41  | 0.40    | 0.42 |
| 9       | 271.22         | 0.29        | 27          | 2922.42     | 0.41         | 0.41  | 0.40    | 0.43 |
| 10      | 238.58         | 0.48        | 30          | 1937.49     | 0.28         | 0.29  | 0.29    | 0.30 |
| 11      | 265.79         | 0.35        | 27          | 1701.76     | 0.27         | 0.32  | 0.30    | 0.29 |
| 12      | 262.26         | 0.47        | 28          | 1431.97     | 0.35         | 0.34  | 0.29    | 0.36 |
| 13      | 272.17         | 0.42        | 27          | 1769.65     | 0.29         | 0.30  | 0.29    | 0.32 |
| 14      | 293.61         | 0.26        | 25          | 2918.53     | 0.4          | 0.44  | 0.47    | 0.42 |
| 15      | 271.16         | 0.34        | 27          | 2437.62     | 0.49         | 0.49  | 0.47    | 0.44 |
| 16      | 242.95         | 0.51        | 30          | 1155.68     | 0.38         | 0.43  | 0.34    | 0.44 |
| 17      | 276.5          | 0.57        | 26          | 3767.02     | 0.52         | 0.48  | 0.57    | 0.46 |
| 18      | 284.65         | 0.53        | 26          | 2498.64     | 0.62         | 0.63  | 0.57    | 0.48 |
| 19      | 276.15         | 0.25        | 26          | 1874.84     | 0.23         | 0.24  | 0.23    | 0.26 |
| 20      | 289.18         | 0.5         | 25          | 1963.66     | 0.3          | 0.31  | 0.39    | 0.19 |
| 21      | 269.94         | 0.31        | 27          | 1156        | 0.32         | 0.32  | 0.29    | 0.19 |
| 22      | 245.47         | 0.3         | 30          | 1793.12     | 0.24         | 0.24  | 0.28    | 0.24 |
| 23      | 250.2          | 0.25        | 30          | 3321.12     | 0.36         | 0.35  | 0.36    | 0.29 |
| 24      | 244.37         | 0.39        | 31          | 3622.61     | 0.4          | 0.39  | 0.45    | 0.30 |
| 25      | 283.94         | 0.33        | 26          | 3716.77     | 0.41         | 0.40  | 0.42    | 0.43 |
| 26      | 271.17         | 0.42        | 27          | 3791.69     | 0.46         | 0.45  | 0.44    | 0.45 |
| 27      | 292.13         | 0.5         | 25          | 2913.97     | 0.58         | 0.57  | 0.49    | 0.63 |
| 28      | 263.87         | 0.2         | 28          | 1920.2      | 0.21         | 0.22  | 0.23    | 0.25 |
| 29      | 257.63         | 0.57        | 29          | 1824.02     | 0.27         | 0.26  | 0.29    | 0.32 |
| 30      | 256.73         | 0.24        | 29          | 3542.53     | 0.33         | 0.35  | 0.38    | 0.38 |
| 31      | 236.84         | 0.21        | 31          | 3622.95     | 0.3          | 0.29  | 0.31    | 0.38 |
| 32      | 256.7          | 0.31        | 29          | 3189.63     | 0.39         | 0.38  | 0.37    | 0.45 |
| 33      | 238.91         | 0.33        | 30          | 3936.37     | 0.36         | 0.35  | 0.37    | 0.39 |
| 34      | 241.7          | 0.25        | 31          | 2869.59     | 0.37         | 0.36  | 0.35    | 0.38 |
| 35      | 267.45         | 0.33        | 27          | 3263.39     | 0.42         | 0.41  | 0.39    | 0.41 |
| 36*     | 266.6          | 0.35        | 28          | 2559.31     | 0.48         | 0.54  | 0.43    | 0.49 |

Table 7. Performance measures of the prediction algorithms.

|          | QPSO     | LSBOOST  | PSO      | GA       |
|----------|----------|----------|----------|----------|
| MAPE     | 5.03120722 | 9.11473319 | 12.1995498 | 10.7931852 |
| MAE      | 1.59%    | 2.92%    | 3.94%    | 3.83%    |
| RMSE     | 2.17%    | 3.74%    | 4.99%    | 4.86%    |
| CVRMSE   | 20.8152239 | 35.8338906 | 46.5072357 | 47.7823492 |
| \( R^2 \) | 0.95     | 0.88     | 0.84     | 0.871    |

MAPE, mean absolute percentage error; MAE, mean absolute error; CVRMSE, coefficient of variation of root mean square error; \( R^2 \), coefficient of determination.

Figure 1 illustrates a graphical comparison between the actual surface roughness values and the predicted values by QPSO, LSBoost, PSO and GA algorithms. The superior prediction performance of the QPSO can be clearly noted by the close match of each data point on the graph with the actual value of the surface roughness. Figure 2 provides the trendline between actual versus predicted values of each prediction algorithm, to provide the coefficient of determination values \( R^2 \).
Figure 1. Actual surface roughness values with the predicted values by QPSO, LSBoost, PSO and GA algorithms.

Figure 2. Actual values of the surface roughness vs. predicted values by QPSO, LSBoost, PSO and GA with $R^2$ trend lines.

In addition, Figure 3 illustrates the absolute error percentage of each data point predicted by the algorithms, to assess the performance of each algorithm at each experiment. The highest error percentage can be observed at the PSO predicted values, with maximum error values of 28%. The LSBoost resulted in an overshoot at some prediction data points, where the maximum error values reached 45% at Experiment 27. However, the overall error values of the LSBoost can be observed within the range of 25%. In addition, the GA algorithm resulted in a comparable performance with the LSBoost algorithm. The QPSO algorithm resulted in a promising performance with maximum error values of approximately 20%.
4. Conclusions

This paper presented an investigation of the accuracy of state-of-the-art machine-learning and quantum-behaved evolutionary algorithms in predicting the surface roughness of a face-milling machine. Ensemble learning based on least-square-gradient boosting and quantum-behaved particle-swarm optimization algorithm were utilized to predict the surface-roughness values, given the machining parameters of a face milling machine. The performance of these algorithms was compared with other existing methods in the literature, specifically the genetics algorithm and the particle swarm optimization, and have resulted in a better prediction performance. The accuracy of the algorithms was assessed based on the RMSE, MAPE, MAE, CVRMSE and $R^2$ values of each algorithm. QPSO achieved the best prediction accuracy in terms of its performance indicators with an RMSE of 2.17%, MAE of 1.59% and $R^2$ of 0.95. LSBoost achieved the second-best prediction performance with RMSE of 3.74%, MAE of 2.92% and $R^2$ of 0.88. It is evident that QPSO and LSBoost have resulted in a high extent of accuracy in predicting the surface roughness in the face-milling machine. Moreover, utilizing such methods in predicting the surface roughness before the production would save a lot of resources in terms of expenses, labor and time.

Author Contributions: Conceptualization, M.S.A. and A.M.A.; methodology, M.S.A. and A.M.A.; software, M.S.A. and A.M.A.; validation, M.S.A. and A.M.A.; formal analysis, M.S.A. and A.M.A.; investigation, M.S.A. and A.M.A.; resources, M.S.A. and A.M.A.; data curation, M.S.A. and A.M.A.; writing—original draft preparation, M.S.A. and A.M.A.; writing—review and editing, M.S.A. and A.M.A.; visualization, M.S.A. and A.M.A. Both authors have read and agreed to the published version of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.
Abbreviations

\( v_{\text{min}} \)  minimum spindle speed (rpm)

\( v_{\text{max}} \)  maximum spindle speed (rpm)

\( f_{\text{min}} \)  minimum feed rate (mm/rev)

\( f_{\text{max}} \)  maximum feed rate (mm/rev)

\( a_{\text{min}} \)  minimum depth of cut (mm)

\( T_m \)  machining time (s)

\( R_a \)  surface roughness (\( \mu \)m)

\( L \)  length of the work piece (mm)

\( B \)  width of the work piece (mm)

\( a_{\text{max}} \)  maximum depth of cut (mm)

References

1. Mundada, V.; Narala, S.K.R. Optimization of Milling Operations Using Artificial Neural Networks (ANN) and Simulated Annealing Algorithm (SA). *Mater. Today Proc.* 2018, 5, 4971–4985. [CrossRef]

2. Raja, S.B.; Baskar, N. Application of Particle Swarm Optimization technique for achieving desired milled surface roughness in minimum machining time. *Expert Syst. Appl.* 2012, 39, 5982–5989. [CrossRef]

3. Rao, R.V.; Rai, D.P.; Balic, J. A multi-objective algorithm for optimization of modern machining processes. *Eng. Appl. Artif. Intell.* 2017, 61, 103–125. [CrossRef]

4. Chua, M.S.; Loh, H.T.; Wong, Y.S.; Rahman, M. Optimization of cutting conditions for multi-pass turning operations using sequential quadratic programming. *J. Mater. Process. Technol.* 1991, 28, 253–262. [CrossRef]

5. Al-Ahmari, A.M.A. Mathematical model for determining machining parameters in multipass turning operations with con-straints. *Int. J. Prod. Res.* 2001, 39, 3367–3376. [CrossRef]

6. Lee, B.; Tarng, Y. Cutting-parameter selection for maximizing production rate or minimizing production cost in multistage turning operations. *J. Mater. Process. Technol.* 2000, 105, 61–66. [CrossRef]

7. Sundaram, R.M. An application of goal programming technique in metal cutting. *Int. J. Prod. Res.* 1978, 16, 375–382. [CrossRef]

8. Manna, A.; Salodkar, S. Optimization of machining conditions for effective turning of E0300 alloy steel. *J. Mater. Process. Technol.* 2008, 203, 147–153. [CrossRef]

9. Senthilkumar, J.S.; Selvarani, P.; Arunachalam, R.M. Intelligent optimization and selection of machining parameters in finish turning and facing of Inconel 718. *Int. J. Adv. Manuf. Technol.* 2011, 58, 885–894. [CrossRef]

10. Saravanan, R.; Asokan, P.; Sachidanandam, M. A multi-objective genetic algorithm (GA) approach for optimization of surface grinding operations. *Int. J. Mach. Tools Manuf.* 2002, 42, 1327–1334. [CrossRef]

11. Li, C.; Chen, X.; Tang, Y.; Li, L. Selection of optimum parameters in multi-pass face milling for maximum energy efficiency and minimum production cost. *J. Clean. Prod.* 2017, 140, 1805–1818. [CrossRef]

12. Yildiz, A.R. A novel particle swarm optimization approach for product design and manufacturing. *Int. J. Adv. Manuf. Technol.* 2008, 40, 617–628. [CrossRef]

13. Krishna, A.G.; Rao, K.M. Optimisation of machining parameters for milling operations using a scatter search approach. *Int. J. Adv. Manuf. Technol.* 2006, 31, 219–224. [CrossRef]

14. Nasiri, M.M.; Kianfar, F. A hybrid scatter search for the partial job shop scheduling problem. *Int. J. Adv. Manuf. Technol.* 2010, 52, 1031–1038. [CrossRef]

15. Liu, X.-J.; Yi, H.; Ni, Z.-H. Application of ant colony optimization algorithm in process planning optimization. *J. Intell. Manuf.* 2010, 24, 1–13. [CrossRef]

16. Li, P.; Zhu, H. Parameter Selection for Ant Colony Algorithm Based on Bacterial Foraging Algorithm. *Math. Probl. Eng.* 2016, 1–12. [CrossRef]

17. Vundavilli, P.R.; Parappagoudar, M.; Kodali, S.; Benguluri, S. Fuzzy logic-based expert system for prediction of depth of cut in abrasive water jet machining process. *Knowl. Based Syst.* 2012, 27, 456–464. [CrossRef]

18. Rana, P.; Lalwani, D. Parameters optimization of surface grinding process using Modified \( \varepsilon \) constrained Differential Evolution. *Mater. Today Proc.* 2017, 4, 10104–10108. [CrossRef]

19. Yildiz, A.R. Hybrid Taguchi-differential evolution algorithm for optimization of multi-pass turning operations. *Appl. Soft Comput.* 2013, 13, 1433–1439. [CrossRef]

20. Rao, R.V.; Rai, D.P.; Balic, J. Multi-objective optimization of machining and micro-machining processes using non-dominated sorting teaching–learning-based optimization algorithm. *J. Intell. Manuf.* 2016, 29, 1715–1737. [CrossRef]

21. Rao, R.V.; Rai, D.P.; Balic, J.; Balic, J. Optimization of Abrasive Waterjet Machining Process using Multi-objective Jaya Algorithm. *Mater. Today Proc.* 2018, 5, 4930–4938. [CrossRef]

22. Alajmi, M.S.; Alfares, F.S.; Alfares, M.S. Selection of optimal conditions in the surface grinding process using the quantum based optimisation method. *J. Intell. Manuf.* 2017, 30, 1469–1481. [CrossRef]

23. Yildiz, A.R.; Solanki, K.N. Multi-objective optimization of vehicle crashworthiness using a new particle swarm based ap-proach. *Int. J. Adv. Manuf. Technol.* 2012, 59, 367–376. [CrossRef]
24. Baraheni, M.; Amini, S. Comprehensive optimization of process parameters in rotary ultrasonic drilling of CFRP aimed at minimizing delamination. *Int. J. Light. Mater. Manuf.* 2019, 2, 379–387. [CrossRef]

25. Bustillo, A.; Urbikain, G.; Perez, J.M.; Pereira, O.M.; Lopez de La Calle, L.N. Smart optimization of a friction-drilling process based on boosting ensembles. *J. Manuf. Syst.* 2018, 48, 108–121. [CrossRef]

26. Sanchez, J.A.; López de La Calle, L.N.; Lamikiz, A. A computer-aided system for the optimization of the accuracy of the wire electro-discharge machining process. *Int. J. Comput. Integr. Manuf.* 2004, 17, 413–420. [CrossRef]

27. Eberhart, R.C.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; IEEE: New York, NY, USA, 1995; pp. 39–43.

28. Rashmi, L.M.; Karthik, R.M.C.; Arunkumar, S.; Shrikantha, S.R.; Herbert, M.A. Machining parameters optimization of AA6061 using response surface methodology and particle swarm optimization. *Int. J. Precision Eng. Manuf.* 2018, 19, 670–695.

29. Zhang, Y.; Wang, S.; Ji, G. A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. *Math. Prob. Eng.* 2015, 2015, 1–38. [CrossRef]

30. Mirjalili, S.; Dong, J.S.; Lewis, A.; Sadiq, A.S. Particle Swarm Optimization: Theory, Literature Review, and Application in Airfoil Design. *Stud. Comput. Intell.* 2020, 35, 1359–1373. [CrossRef]

31. Chen, F.A.; Gökçe, F.; Yüksel, A.S.; Yiğit, T. A novel hybrid PSO–GWO algorithm for optimization problems. *Eng. Comput.* 2019, 35, 1359–1373. [CrossRef]

32. Imran, M.; Hashim, R.; Khalid, N.E.A. An Overview of Particle Swarm Optimization Variants. *Procedia Eng.* 2013, 53, 491–496. [CrossRef]

33. Qin, J.; Liu, Y.; Grosvenor, R.; Lacan, F.; Jiang, Z. Deep learning-driven particle swarm optimisation for additive manufacturing energy optimisation. *J. Clean. Prod.* 2020, 245, 118702. [CrossRef]

34. Raju, M.; Gupta, M.K.; Bhanot, N.; Sharma, V.S. A hybrid PSO-BFO evolutionary algorithm for optimization of fused depositions process parameters. *J. Intell. Manuf.* 2019, 30, 2743–2758. [CrossRef]

35. Sun, J.; Feng, B.; Xu, W. Particle swarm optimization with particles having quantum behavior. In Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753), Portland, OR, USA, 19–23 June 2004; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA, 2004; pp. 325–331.

36. Sun, J.; Fang, W.; Wu, X.; Palade, V.; Xu, W. Quantum-Beheaved Particle Swarm Optimization: Analysis of Individual Particle Behavior and Parameter Selection. *Evol. Comput.* 2012, 20, 349–393. [CrossRef]

37. Xi, M.; Sun, J.; Xu, W. An improved quantum-behaved particle swarm optimization algorithm with weighted mean best po-sition. *Appl. Math. Comput.* 2008, 205, 751–759.

38. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* 2001, 1189–1232. [CrossRef]

39. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, Anchorage, AK, USA, 4–9 May 1998; IEEE: New York, NY, USA, 2002; pp. 69–73.

40. Boeringer, D.W.; Werner, D.H. Particle Swarm Optimization Versus Genetic Algorithms for Phased Array Synthesis. *IEEE Trans. Antennas Propag.* 2004, 52, 771–779. [CrossRef]