


## Article

# Bayesian Optimized Deep Convolutional Network for Electrochemical Drilling Process

Yanfei Lu <sup>1,\*</sup> , Zengyan Wang <sup>2</sup>, Rui Xie <sup>3</sup> and Steven Liang <sup>1,4</sup>

<sup>1</sup> The George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

<sup>2</sup> Department of Computer Science, University of Georgia, Athens, GA 30605, USA

<sup>3</sup> Department of Statistics & Data Science, University of Central Florida, Orlando, FL 32816, USA

<sup>4</sup> College of Mechanical Engineering, Donghua University, Shanghai 201620, China

\* Correspondence: ylu318@gatech.edu; Tel.: +1-404-894-8164

Received: 11 June 2019; Accepted: 11 July 2019; Published: 14 July 2019



**Abstract:** Electrochemical machining is a promising non-traditional manufacturing process to make high-quality parts. The benefits of minimal thermally and mechanically induced stresses, free of burr, and a low surface roughness are appealing for industry and research institutes. However, the combined chemical reaction, electric field, fluid mechanics, and material properties involve a significant number of independent parameters which are difficult to analyze in order to draw comprehensive conclusions. To our current knowledge, process responses such as the material removal rate, optimal feed rate, and cutting profile cannot be represented accurately by analytical solutions. In recent years, deep learning has had tremendous success in analyzing sophisticated systems. The improved computation efficiency and reduced size of the training dataset required for deep learning have enabled various prediction models in the manufacturing industry. In this paper, a new approach is developed using the deep convolutional network with the Bayesian optimization algorithm to predict the diameters of the drilled hole from an electrochemical machining process. The Keras application programming interface (API) was used to build the deep convolutional network; the feed rate, pulse-on time, and voltage were used as input parameters to provide a fair comparison with a neural network from previous research. Random dropout layers were added to prevent overfitting of the network. Instead of tuning the network parameter by trial and error, the Bayesian parameter optimization algorithm was implemented to find the optimal set of parameters of the deep convolutional network that yields the minimum mean square error. The proposed algorithm was compared with a previously developed neural network with partially embedded physical knowledge. Improved training speed and accuracy were observed in comparison with the traditional neural network. The prediction model using the proposed deep learning algorithm demonstrated better prediction accuracy and provided a more systematic way to select the hyperparameter for the deep convolutional network.

**Keywords:** Bayesian optimization; convolutional neural network; deep learning; electrochemical micro-machining

## 1. Introduction

Starting from the early casting process to today's hybrid subtractive and additive processing, the increasing demand for product quality has driven the new development of manufacturing processes. Maximizing productivity while reducing cost and achieving a better surface finish that is free of induced residual stress has always been the goal of manufacturing industries. In recent years, the popularity of non-traditional manufacturing processes has increased significantly because

of the ability to process difficult-to-cut material while minimizing mechanically induced stresses. The electrical discharge machining (EDM) process is capable of machining materials with high hardness such as tungsten carbide and carbon fiber-reinforced polymers [1,2]. However, the recast layer by the electrode deposition and thermally influenced machining zone could affect the material properties [3,4]. The heat-affected zone (HAZ) will potentially induce residual stresses and phase transformation for heat-sensitive materials [5]. For materials used in critical applications such as nickel titanium alloys the thermally induced phase transformation could significantly influence the proper function of the device and reduce the fatigue life [6,7]. The laser beam machining (LBM) process is a non-contact process that melts and vaporizes material from the parent material [8]. This process is generally used for the high-precision manufacture of complex shapes. The high-energy beam provides a high material removal rate and is not sensitive to material hardness [9]. However, based on the nature of the material removal process of LBM, it poses the same or an even worse problem as the EDM process. The HAZ could be as large as 3 mm in depth during the LBM process of Ti6Al4V, as reported by Yang et al. [10]. In comparison with the previously mentioned method, the electrochemical machining (ECM) process is capable of processing hard-to-cut material while generating a residual stress-free surface with low roughness [11]. However, the complexity of the ECM process and limited physical understandings have prevented the processing from competing with other processes in mass production.

Lohrengel et al. investigated the anodic dissolution of the ECM process and reported that the oxide film and supersaturated viscous film consist of the dissolved material and depleted ions could affect the material removal [12]. However, these quantities are not easily measurable during the machining process, and hence are difficult to control. Bhattacharyya et al. examined the influence of current efficiency, power supply, tool design, electro-gap, and electrolyte of ECM [13]. Although it provides insights into the qualitative selection of process parameters, quantitative measures such as the applied voltage, current, and feed rate are important for manufacturers in the setup of the machining process. McGeough presented analytical solutions for several process parameters in [14]. The hydrogen bubble at the inter electro-gap is a key factor for the determination of the material removal rate. Thorpe et al. [15] developed an analytical representation of the void fraction in the electrolyte between the electrode. However, the equilibrium cannot be maintained during the process because of the small inter electro-gap size and insufficient electrolyte supplies. Because of the abovementioned difficulties, industry and research tend to seek help from intelligent techniques which are capable of analyzing highly nonlinear problems through learning from experience. The emerging deep learning algorithm has quickly gained favor across different industries for solving prediction problems.

Zain et al. implemented neural network to predict surface roughness in the machining process using cutting speed, feed rate, coating, and radial rake angle [16]. Various network structures are compared to determine the best selection of network structure using trial and error. Lu et al. [11] developed a neural network with partially imbedded physical understanding to predict drilling diameter for the ECM process. The structure of the network is pre-defined using known analytical solutions; however, the rest of the neural network structure is determined by trial and error. Fu et al. proposed a deep belief network (DBN) to classify the cutting state for machine monitoring in [17]. The network has shown significant improvement in error reduction in comparison to the traditional neural network and support vector machine. However, the details of selection of the network structure were not discussed. Li et al. implemented DBN to predict machine backlash error using inputs such as machining torque, ambient temperature, and measured position [18]. The trained network yields an accurate prediction for the backlash error. However, the selection of the network structure appears to be arbitrary. Other similar works using learning algorithms are presented in [19]. The lack of a systematic method to select the optimal network structural remains a challenge today. To shorten the time-to-market, a more systematic and scientific approach needs to be developed for neural network hyperparameter optimization.

In this paper, a Bayesian optimization algorithm is implemented to optimize structures of a deep convolutional neural network which uses voltage, feed rate, and pulse-on time to predict the

ECM drilled diameter at the entry and exit of the hole. The deep learning convolutional neural network consists of one convolution layer, one random drop out layer, and three fully connected layers to map out the relationship between inputs (feed rate, voltage, pulse-on time) and outputs (entry and exit hole diameters). Because ECM is a very complicated process, the implementation of the highly nonlinear deep convolutional network helps improve the prediction accuracy. The Bayesian optimization algorithm aims to find an optimal set of parameters that will minimize the mean square error (MSE) without performing an exhaustive search that demands significantly more computation power. The proposed network is compared with a previously developed physics embedded neural network and a traditional neural network to demonstrate its improved performance. The rest of the paper is organized as follows: Section 2 briefly describes the deep convolutional network and Bayesian optimization; Section 3 describes the experimental setup of the  $\mu$ -ECM; Section 4 presents the results based on the proposed approach and a comparison with previous work; and Section 5 presents conclusions from the presented work as well as possible future directions.

## 2. Deep Convolutional Network Prediction Model for ECM

### 2.1. Deep Convolutional Network

The deep convolutional network, as a novel and powerful tool to capture the complex dependency between input and output signals, to the best of our knowledge, has not been implemented in the ECM process. In addition, the networks are generally tuned by trial and error. The Bayesian optimization algorithm facilitates the automatic tuning of the network without human intervention, which significantly reduces the effort or skill level required for industrial applications. In previous documented research, the convolutional neural network (CNN) was demonstrated to work well in identifying patterns in data. Moreover, it has been adopted in various applications in scenarios with the absence of human experts or for the adaption of solutions to specific cases [20–22] because of its capability of solving highly nonlinear problems and its generalization to different data types. It was initially proposed for pattern recognition with back propagation in 2D or 3D applications such as images and video frames [23]. Alternatively, 1D CNN was developed for applications over 1D signals such as electrocardiogram (ECG) and mechanical data. Our 1D CNN architecture has a combination of two types of layers: convolution and fully connected. In the convolution layers, the data are convolved using local learnable kernels to form the output feature maps. For the neural nets with a specified nonlinear function and choice of activation function, the following equation is given:

$$x_{out}^{(l)} = g(h^{(l-1)}) = \sigma(\sum x_{in} \times W_j + b_j) \quad (1)$$

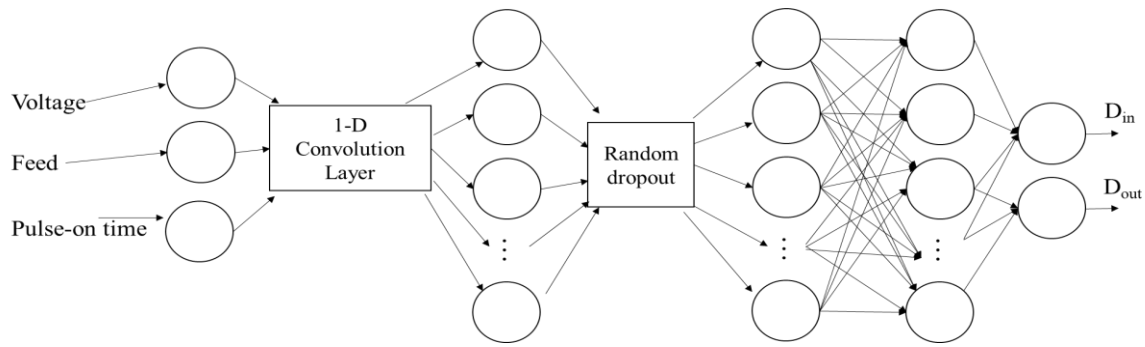
where  $x_{out}^{(l)}$  is the output feature map of the current layer  $l$ ,  $x_{in}$  is the input data, and  $W_j$  and  $b_j$  are the kernel and bias for the current layer.

The output feature maps are then flattened and fed into the next layers for processing. The added convolution is sensitive to nonlinear functions such as sinusoidal and periodic functions—those used in Fourier transform. For the ECM application, because of the unknown intermediate parameters such as the gas bubble fraction, current density, and efficiency, adding the convolution layer could be beneficial to capture their interaction. The softmax function is discarded because the CNN is used for prediction instead of classification. The activation function  $\sigma(\cdot)$  between each layer is selected to be a Rectified Linear Unit (ReLU):

$$[\sigma(z)]_j = \max\{z_j, 0\} \quad (2)$$

for computation efficiency [22]. The voltage, feed, and pulse-on time are selected as the input of the network and the drilled hole diameter at the top and bottom are selected to be the output.

The network structure used in the ECM application is shown in Figure 1.



**Figure 1.** Deep convolutional network to predict electrochemical machining (ECM) drilling.

A random dropout layer with an optimized percentage of dropout ratio is added between the first and second hidden layer to prevent overfitting [24]. By dropping out subsets of features in the training process, dropout can effectively prevent overfitting. The random dropout for the  $l$ -th layer can be described as:

$$h_{drop}^{(l)} = h^{(l)} \odot mask^l \quad (3)$$

where  $\odot$  is the element-wise multiplication and  $mask^l$  is a vector of the independent and identically distributed (i.i.d.) Bernoulli dropout variable with success probability  $p$ . The data are split into training and validation sets. The validation split percentage, dropout ratio, and number of neurons in the first, second, and third hidden layers are optimized using the Bayesian optimization algorithm described in the following section.

## 2.2. Bayesian Optimization of Hidden Layers with Gaussian Process Priors

The Bayesian optimization aims to utilize Bayes' rule to find the minimum or maximum of an objective function  $f(x)$  within a bounded set  $\chi$ . In comparison with the traditional grid search method, which requires a significant amount of computational power, the Bayesian algorithm obtains an optimal set of solutions with less iterations, which is critical for online diagnostic applications. The Bayesian optimization is able to automatically quantify the uncertainty of the minimizer or maximizer. For a generic family of models with data observation  $D$  and parameter  $x \in \chi$ , we assume a prior distribution  $p(x)$  and a likelihood  $p(D|x)$  with given data  $D$ . We can infer the posterior distribution using Bayes' rule  $p(x|D) \propto p(D|x)p(x)$ . The maximizer (or minimizer) of the objective function then follows the maximum a posteriori (MAP) probability:

$$x^* = \underset{x \in \chi}{\operatorname{argmin}} f(x) = \underset{x \in \chi}{\operatorname{argmin}} (x|D). \quad (4)$$

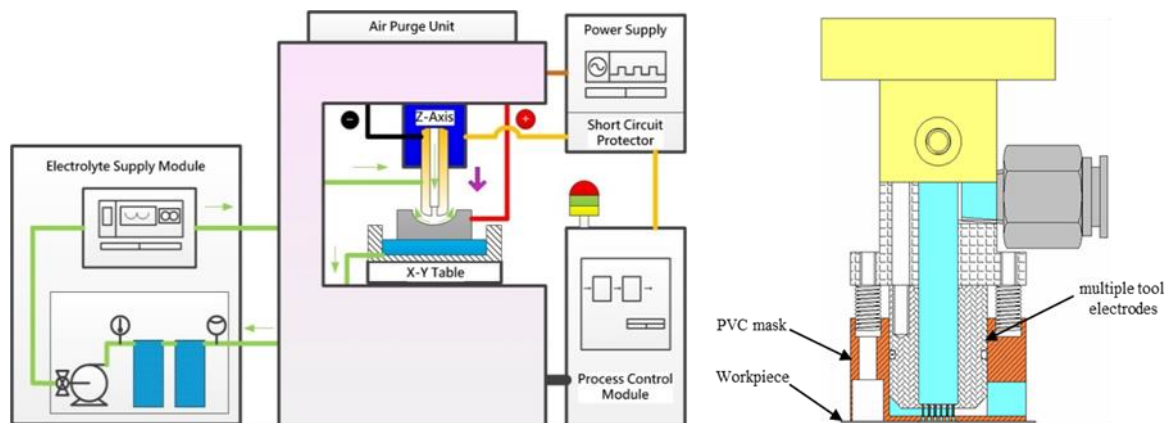
In this case, domain  $\chi$  will be the range of the parameter within the network. We assume the Gaussian process prior such that the observation  $D_{1:t} = \{x_{1:t}, y_{1:t}\}$  follows  $y_{1:t} \sim \text{Normal}(f(x_{1:t}), \Sigma(x_{1:t}, x_{1:t}))$ . Then the posterior probability distribution has the form  $f(x)|f(x_{1:t}) \sim \text{Normal}(\mu_t(x|D), \sigma_t^2(x|D))$ , where  $\mu_t(x|D)$  and  $\sigma_t^2(x|D)$  have complicated forms that are usually intractable. The Bayesian optimization establishes a probability model for  $f(x)$  by selecting various parameter values within the set  $\chi$ . The model stores the previous calculated  $f(x)$  value and evaluation the area with higher probability to generate the minimum or maximum of  $f(x)$  without relying on the local gradient [25]. Rather than using an exhaustive search algorithm, the Bayesian optimization targets the area with a higher density of lower values of the cost function  $f(x)$ , which reduces the computation effort significantly. The basic algorithm [26] is described as follows:

We assumed an optimized solution  $\hat{x}$  which solves the relationship  $\hat{x} = \underset{x \in \chi}{\operatorname{argmin}}_x f(x) = \underset{x \in \chi}{\operatorname{argmin}}_p(f(x)|f(x_{1:t}))$  or  $\hat{x} = \underset{x \in \chi}{\operatorname{argmax}}_x f(x) = \underset{x \in \chi}{\operatorname{argmax}}_p(f(x)|f(x_{1:t}))$ , the objective is assumed to be a continuous function that is differentiable. For  $t = 1, 2, \dots$ , select various  $x_t$  by optimizing the

acquisition function over the Gaussian process:  $x_t = \operatorname{argmax}_x u(x|D_{1:t-1})$ , where  $D_{1:t}$  stores the previous observations  $\{x_{1:t}, y_{1:t}\}$ . In this case, a prior distribution  $p(f(x))$  can be assumed as normal distribution. The dataset  $D_{1:t}$  will be augmented with the newly acquired  $\{x_t, y_t\}$ . Then, the whole process will repeat until certain criteria are met. The Bayesian optimization algorithm is combined with the CNN deep learning network to optimize the parameters within the network while reducing the training MSE. The Bayesian optimization is implemented to automatically tune the parameters in the CNN, playing the similar role as the widely used cross-validation. The next section explains the case study with the ECM drilling process.

### 3. Experimental Study

The experimental data collected in [11,27,28] were adopted to validate the proposed deep convolutional network. The experimental setup of the ECM process is shown in Figure 2. The system is composed of a three-axis computer numerical controlled table, a small-scale power supply of 100 A, and an electrolyte-delivering pump with a built-in filtration system for slag removal and electrolyte refreshing. The operating system is controlled by an RTX real-time Windows kernel program which shows the machining parameter and machine positions. A pulse generator supplies a periodical voltage to the ECM machine to ensure a sufficient replenishing rate of active ions in the electro-gap. A digital oscilloscope ensures that the pulse generator produces a rectangular waveform with the desired amplitude. A short-circuit protection/detection mechanism is implemented if the electrode feed rates are excessive, to prevent the tool from contacting the workpiece. Whenever the oscilloscope detects a short circuit, a signal is sent rapidly to the controller and the electrode is retracted gradually until the measured voltage returns to normal. The electrode consists of arrays of a cylindrical copper tool, a polyvinyl chloride (PVC) mask, and a tool fixture. The electrolyte is pumped to a multiple-electrode cell and exits through the small nozzles directed towards the anode workpiece.

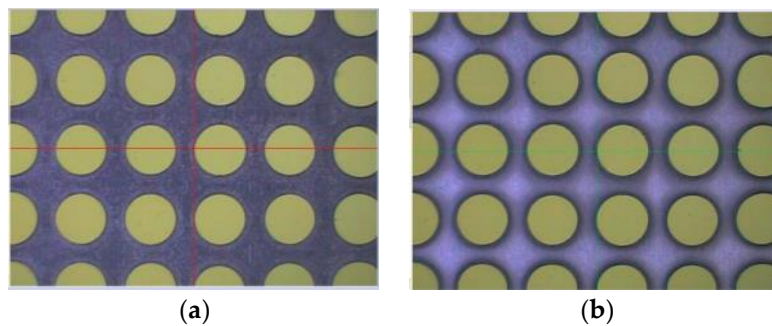


**Figure 2.** Schematic diagram of electrochemical micromachining system (left) and micro-array hole electrode module (right) [11,27,28].

The electrolyte velocity at the outlet of the pump was set at 10 m/s; the average electrolyte temperature was measured to be 27 °C; the initial gap between the tool and the workpiece was set at 100 µm; the total tool travel was 800 µm; the workpiece material was 304 stainless steel; the electrolyte used was 10% wt. NaNO<sub>3</sub>; the nominal diameter of the hole to be drilled was 900 µm; and the depth of the hole was 500 µm. The voltage, pulse-on-time, and feed rate were used as the controllable process parameters because of the relatively low difficulty in measuring these factors, while the entry diameter of the micro-hole  $D_{in}$  and the exit diameter  $D_{out}$  were the response variables. Figure 3 shows the charge coupled device (CCD) camera image of the array of holes drilled during the µ-ECM experiment. It was observed that even the cutting parameters such as feed rate, pulse-on time, and voltage were the same. The patterns of the drilled hole were different. The possible causes were determined from the



difference in electrolyte velocity at different parts of the workpiece, the interaction of the electrical fields between tools, and the existing active ions at different areas of the workpiece.



**Figure 3.** Picture taken by charge coupled device (CCD) camera. (a) The entry side of the hole (Din); (b) the exit side of the hole (Dout) [28].

In order to create a forward prediction model for ECM drilling, the design of the experiment was implemented with three different sets of experiments generated. The cutting input parameter and output parameter were recorded. The experimental data results are shown in Table A1 in Appendix A.

#### 4. Results

The proposed deep convolutional network was trained using the data shown in Table A1. The parameters to be optimized were the training and validation split, the drop-out ratio of the layer, the neurons in each layer, and the training iteration. The input and output were normalized from 0 to 1. Rather than using the time-consuming grid and random search methodology, the Bayesian-based optimization was implemented to optimize the model parameter. To initialize the tuning parameter boundary, a trial-and-error based approach was implemented to determine the basic structure of the deep convolution network. The structure of the network refers to the previous work in [11,27,28] with an added convolutional layer and dropout layer to add nonlinearity and prevent overfitting. The network was initialized with a convolutional layer and one dense layer with 20 neurons. The entry and exit drilled diameters were predicted by the network. After the MSEs between the actual and predicted values were obtained, one more layer was added while reducing the neurons in the previous dense layer. The procedure was repeated until no significant improvement (less than 10%) in reducing the MSE was observed. The network structure was determined to be one convolutional layer for the first layer with four subsequent dense layers. The dropout layer was added in between the first dense layer and second dense layer.

The loss function was selected to be the mean square error to provide a fair comparison with the previously developed work. The optimizer was selected to be the Nesterov adaptive moment estimation [29,30], which has been proven to improve the rate of convergence and reduce the loss function. The range of the parameter was selected as shown in Table 1.

**Table 1.** Deep convolutional network parameter range.

	Validation Split	Drop-Out Layer Ratio	Dense Layer 1 Neurons	Dense Layer 2 Neurons	Dense Layer 3 Neurons	Training Iteration
Range	0–0.2	0–0.3	3–8	3–8	3–8	50–400 in increments of 50
Data Type	Continuous	Continuous	Discrete	Discrete	Discrete	Discrete

The optimized network parameter by the Bayesian algorithm is shown in Table 2, and the result of the mean square error (MSE) and mean average error (MAE) at the end of the training is shown in Table 3. The average value was calculated based on five different simulations.

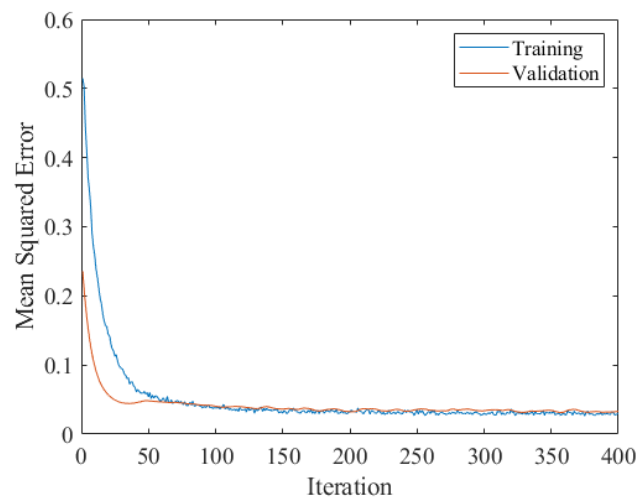
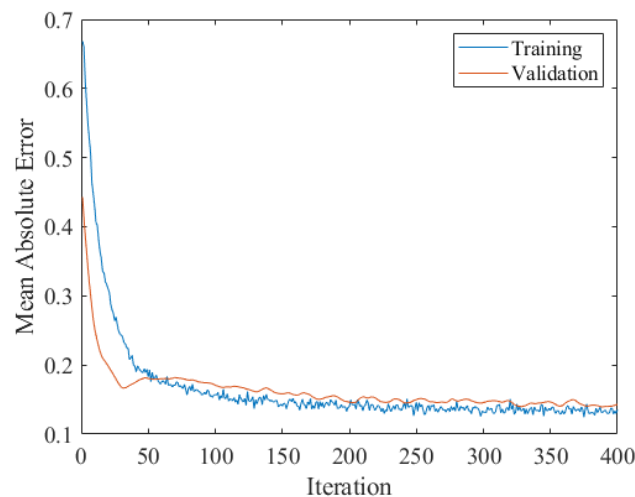
**Table 2.** Bayesian optimized deep convolutional network parameter for ECM.

	Validation Split	Drop-Out Layer Ratio	Dense Layer 1 Neurons	Dense Layer 2 Neurons	Dense Layer 3 Neurons	Training Iteration
Value	89.5% training 10.5% validation	5.23%	8	7	6	400

**Table 3.** Bayesian optimized deep convolutional network performance in terms of the prediction of entry and exit diameters of the ECM drilled holes.

Simulation #	Training MSE	Validation MSE	Training MAE	Validation MAE
1	0.0297	0.0329	0.135	0.143
2	0.0257	0.0219	0.125	0.119
3	0.0246	0.0235	0.121	0.121
4	0.0276	0.0261	0.131	0.129
5	0.0248	0.0207	0.122	0.115
Average	0.0265	0.0250	0.127	0.125

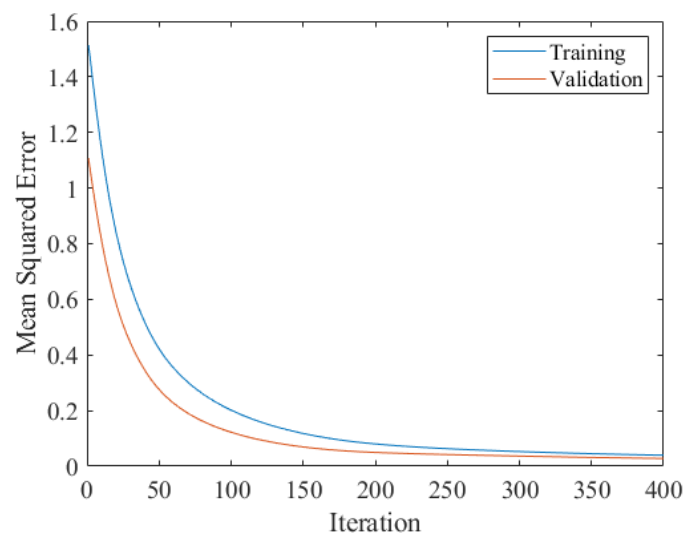
Figures 4 and 5 show the change of MSE and MAE during the training and validation processes. It can be observed that the training converges within 50 iterations, which shows the high training efficiency of the deep CNN.

**Figure 4.** Deep convolutional neural network (CNN) mean squared error during training and validation.**Figure 5.** Deep CNN mean average error during training and validation.

To validate our proposed model, the networks developed in [11,27,28] were compared with the proposed CNN. The mean least square support vector regression machines in [31] were also added for comparison. The result is shown in Table 4. It can be observed that the deep CNN yields the minimum error among all four models. In addition, by comparing Figures 4 and 6, it can be observed that the CNN has a faster convergence speed than the traditional neural network. A statistical test with a null hypothesis that the compared method is equivalent was conducted between the proposed method and the multioutput least square support vector regression (MLS-SVR) method in [31]. The Statistical Tests for Algorithms Comparison (STAC) platform from [32] was implemented. The  $p$ -value was 0.019. Therefore, the null hypothesis was rejected.

**Table 4.** Comparison with different models.

	Deep CNN with Bayesian	Physics Embedded NN [4]	NN from [5]	MLS-SVR [31]
Average Validation MSE	0.0250	0.090	0.114	0.0617
Average Validation MAE	0.125	0.198	0.222	0.234



**Figure 6.** NN mean squared error during training and validation.

## 5. Conclusions

In this paper, a new approach using the deep CNN with Bayesian optimization was introduced to increase the prediction accuracy and rate of convergence for the ECM drilling process. Rather than tuning the parameter by hand or using the grid and random search method, the Bayesian-based optimization algorithm helps navigate to the optimal set of parameters of the CNN. An initial guess of the network structure is made by trial and error to search for the network structure. Then the boundary of the parameter to be optimized is determined based on knowledge of the process and previous documented research. The Bayesian algorithm searches within the prescribed boundary and finds areas that have a higher probability of obtaining the optimal set of parameters. With sufficient sampling, an optimal set of parameters of the deep convolutional network is obtained. The automatic parameter tuning does not require knowledge of the ECM process and the network structure, which is ideal for industrial applications. The deep convolutional network prediction model is compared with a traditional neural network and physics-based NN. The proposed model has the advantages of requiring fewer training iterations to converge and fewer prediction errors in comparison with the previously proposed prediction models. In addition, with the added random dropout layer, the possibility of overfitting the network is decreased. With enough experimental data and more inputs, the proposed model offers a viable route to predict the relationship among the input and output parameters of an



ECM process. The algorithm can be built into the computer system of an ECM drilling machine to facilitate the accurate process control and improve the throughput of the ECM drilling process.

**Author Contributions:** Y.L., Z.W., and R.X. created the model and analyzed the data; S.L. provided feedback on the concept; Y.L., Z.W., and R.X. wrote the paper.

**Funding:** This research received no external funding.

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

## Appendix A

**Table A1.** Data used for training and testing the neural network [28].

No.	Voltage (V)	Pulse-On Time ( $\mu$ s)	Feed Rate ( $\mu$ m/s)	Din ( $\mu$ m)	Dout ( $\mu$ m)	Taper	Overcut ( $\mu$ m)
1	16	25	8	893	860	0.066	3.5
2	18	25	8	929	913	0.032	14.5
3	20	25	8	923	910	0.026	11.5
4	16	25	6	904	892	0.024	2
5	18	25	6	934	931	0.006	17
6	20	25	6	999	977	0.044	49.5
7	16	25	4	983	979	0.008	41.5
8	18	25	4	1050	1045	0.01	75
9	20	25	4	1125	1123	0.004	112.5
10	8	50	8	657.5	627.5	0.06	121.25
11	10	50	8	809.5	807.25	0.0045	45.25
12	12	50	8	866.25	858	0.0165	16.875
13	8	50	6	760	741	0.038	70
14	10	50	6	828.5	829.5	0.002	35.75
15	12	50	6	908.75	905.5	0.0065	4.375
16	8	50	4	781.75	780.25	0.003	59.125
17	10	50	4	887.25	881.75	0.011	6.375
18	12	50	4	957.75	970	0.0245	28.875
19	8	60	8	771.33	759.33	0.024	64.335
20	10	60	8	806.75	799.5	0.0145	46.625
21	12	60	8	862.75	847	0.0315	18.625
22	8	60	6	756.5	739.75	0.0335	71.75
23	10	60	6	776.75	777.5	0.0015	61.625
24	12	60	6	840.25	841.25	0.002	29.875
25	8	60	4	769	771.5	0.005	65.5
26	10	60	4	854.75	865.25	0.021	22.625
27	12	60	4	928.25	945.5	0.0345	14.125
28	8	70	8	718	721.5	0.007	91
29	10	70	8	779	796.75	0.0355	60.5
30	12	70	8	841.5	849.75	0.0165	29.25
31	8	70	6	736.5	744.5	0.016	81.75
32	10	70	6	802	829.75	0.0555	49
33	12	70	6	858.75	865	0.0125	20.625
34	8	70	4	783.25	783.25	0	58.375
35	10	70	4	878.75	872	0.0135	10.625
36	12	70	4	946.25	955.25	0.018	23.125
37	8	50	8	874	704	0.34	13
38	9	50	8	914	789	0.25	7
39	10	50	8	999	827	0.344	49.5
40	8	50	6	922	765	0.314	11
41	9	50	6	955	807	0.296	27.5
42	10	50	6	1039	837	0.404	69.5
43	8	50	4	932	797	0.27	16
44	9	50	4	1044	790	0.508	72
45	10	50	4	1130	858	0.544	115

Table A1. Cont.

No.	Voltage (V)	Pulse-On Time ( $\mu$ s)	Feed Rate ( $\mu$ m/s)	Din ( $\mu$ m)	Dout ( $\mu$ m)	Taper	Overcut ( $\mu$ m)
46	8	60	8	903	708	0.39	1.5
47	9	60	8	967	766	0.402	33.5
48	10	60	8	1084	817	0.534	92
49	8	60	6	917	760	0.314	8.5
50	9	60	6	1043	856	0.374	71.5
51	10	60	6	1115	871	0.488	107.5
52	8	60	4	1071	754	0.634	85.5
53	9	60	4	1087	972	0.23	93.5
54	10	60	4	1263	1044	0.438	181.5
55	8	70	8	875	789	0.172	12.5
56	9	70	8	1071	842	0.458	85.5
57	10	70	8	1158	862	0.592	129
58	8	70	6	987	846	0.282	43.5
59	9	70	6	1212	886	0.652	156
60	10	70	6	1243	1056	0.374	171.5
61	8	70	4	1134	877	0.514	117
62	9	70	4	1260	935	0.65	180
63	10	70	4	1348	1016	0.664	224

## References

- Lee, S.; Li, X. Study of the effect of machining parameters on the machining characteristics in electrical discharge machining of tungsten carbide. *J. Mater. Process. Technol.* **2001**, *115*, 344–358. [\[CrossRef\]](#)
- Yue, X.; Yang, X.; Tian, J.; He, Z.; Fan, Y. Thermal, mechanical and chemical material removal mechanism of carbon fiber reinforced polymers in electrical discharge machining. *Int. J. Mach. Tools Manuf.* **2018**, *133*, 4–17. [\[CrossRef\]](#)
- Klocke, F.; Lung, D.; Antonoglou, G.; Thomaidis, D. The effects of powder suspended dielectrics on the thermal influenced zone by electrodischarge machining with small discharge energies. *J. Mater. Process. Technol.* **2004**, *149*, 191–197. [\[CrossRef\]](#)
- Yadav, V.; Jain, V.K.; Dixit, P.M. Thermal stresses due to electrical discharge machining. *Int. J. Mach. Tools Manuf.* **2002**, *42*, 877–888. [\[CrossRef\]](#)
- Guo, Y.; Klink, A.; Fu, C.; Snyder, J. Machinability and surface integrity of Nitinol shape memory alloy. *CIRP Ann.* **2013**, *62*, 83–86. [\[CrossRef\]](#)
- Bose, A.; Hartmann, M.; Henkes, H.; Liu, H.-M.; Teng, M.M.H.; Szikora, I.; Berlis, A.; Reul, J.; Yu, S.C.; Forsting, M.; et al. A novel, self-expanding, nitinol stent in medically refractory intracranial atherosclerotic stenoses: The Wingspan study. *Stroke* **2007**, *38*, 1531–1537. [\[CrossRef\]](#) [\[PubMed\]](#)
- Pelton, A.; Huang, G.; Moine, P.; Sinclair, R. Effects of thermal cycling on microstructure and properties in Nitinol. *Mater. Sci. Eng. A* **2012**, *532*, 130–138. [\[CrossRef\]](#)
- Dubey, A.K.; Yadava, V. Laser beam machining—A review. *Int. J. Mach. Tools Manuf.* **2008**, *48*, 609–628. [\[CrossRef\]](#)
- Parandoush, P.; Hossain, A. A review of modeling and simulation of laser beam machining. *Int. J. Mach. Tools Manuf.* **2014**, *85*, 135–145. [\[CrossRef\]](#)
- Yang, J.; Sun, S.; Brandt, M.; Yan, W. Experimental investigation and 3D finite element prediction of the heat affected zone during laser assisted machining of Ti6Al4V alloy. *J. Mater. Process. Technol.* **2010**, *210*, 2215–2222. [\[CrossRef\]](#)
- Lu, Y.; Rajora, M.; Zou, P.; Liang, S.Y. Physics-embedded machine learning: Case study with electrochemical micro-machining. *Machines* **2017**, *5*, 4. [\[CrossRef\]](#)
- Lohrengel, M.; Rataj, K.; Munninghoff, T. Electrochemical Machining—Mechanisms of anodic dissolution. *Electrochim. Acta* **2016**, *201*, 348–353. [\[CrossRef\]](#)
- Bhattacharyya, B.; Munda, J.; Malapati, M. Advancement in electrochemical micro-machining. *Int. J. Mach. Tools Manuf.* **2004**, *44*, 1577–1589. [\[CrossRef\]](#)

14. McGeough, J.A. *Principles of Electrochemical Machining*; Chapman & Hall: London, UK, 1974.
15. Thorpe, J.; Zerkle, R. Analytic determination of the equilibrium electrode gap in electrochemical machining. *Int. J. Mach. Tool Des. Res.* **1969**, *9*, 131–144. [CrossRef]
16. Zain, A.M.; Haron, H.; Sharif, S. Prediction of surface roughness in the end milling machining using Artificial Neural Network. *Expert Syst. Appl.* **2010**, *37*, 1755–1768. [CrossRef]
17. Fu, Y.; Zhang, Y.; Qiao, H.; Li, D.; Zhou, H.; Leopold, J. Analysis of feature extracting ability for cutting state monitoring using deep belief networks. *Procedia CIRP* **2015**, *31*, 29–34. [CrossRef]
18. Li, Z.; Wang, Y.; Wang, K. A data-driven method based on deep belief networks for backlash error prediction in machining centers. *J. Intell. Manuf.* **2017**, 1–13. [CrossRef]
19. Kim, D.-H.; Kim, T.J.Y.; Wang, X.; Kim, M.; Quan, Y.-J.; Oh, J.W.; Min, S.-H.; Kim, H.; Bhandari, B.; Yang, I.; et al. Smart machining process using machine learning: A review and perspective on machining industry. *Int. J. Precis. Eng. Manuf. Technol.* **2018**, *5*, 555–568. [CrossRef]
20. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
21. Shin, H.-C.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M.; Hoo-Chang, S. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* **2016**, *35*, 1285–1298. [CrossRef]
22. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems (NIPS): San Diego, CA, USA, 2012.
23. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
24. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
25. Snoek, J.; LaRochelle, H.; Adams, R.P. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems (NIPS): San Diego, CA, USA, 2012.
26. Brochu, E.; Cora, V.M.; De Freitas, N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv* **2010**, arXiv:1012.2599.
27. Zou, P.; Rajora, M.; Ma, M.; Chen, H.; Wu, W.; Liang, S.Y. Electrochemical Micro-Machining Process Parameter Optimization Using a Neural Network-Genetic Algorithm Based Approach. In Proceedings of the International Conference on Manufacturing Technologies, San Diego, CA, USA, 6–9 January 2017.
28. Rajora, M.; Zou, P.; Yang, Y.G.; Fan, Z.W.; Chen, H.Y.; Wu, W.C.; Li, B.; Liang, S.Y. A split-optimization approach for obtaining multiple solutions in single-objective process parameter optimization. *SpringerPlus* **2016**, *5*, 1424. [CrossRef] [PubMed]
29. Dozat, T. Incorporating Nesterov Momentum into Adam. 2016. Available online: [http://cs229.stanford.edu/proj2015/054\\_report.pdf](http://cs229.stanford.edu/proj2015/054_report.pdf) (accessed on 12 July 2019).
30. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
31. Xu, S.; An, X.; Qiao, X.; Zhu, L.; Li, L. Multi-output least-squares support vector regression machines. *Pattern Recognit. Lett.* **2013**, *34*, 1078–1084. [CrossRef]
32. Rodríguez-Fdez, I.; Canosa, A.; Mucientes, M.; Bugarín, A. STAC: A web platform for the comparison of algorithms using statistical tests. In Proceedings of the 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Istanbul, Turkey, 2–5 August 2015; pp. 1–8.

