entropy

MDPI

Article

ML-Based Analysis of Particle Distributions in High-Intensity Laser Experiments: Role of Binning Strategy

Yury Rodimkov [1], Evgeny Efimenko [1,2], Valentin Volokitin [1,3], Elena Panova [1], Alexey Polovinkin [4], Iosif Meyerov [1,3,*] and Arkady Gonoskov [1,2,5,*]

1 Department of Mathematical Software and Supercomputing Technologies, Lobachevsky University, 603950 Nizhni Novgorod, Russia; rodimkov@bk.ru (Y.R.); evgeny.efimenko@gmail.com (E.E.); valyav95@mail.ru (V.V.); alyona-gra98@yandex.ru (E.P.)
2 Institute of Applied Physics of the Russian Academy of Sciences, 603950 Nizhni Novgorod, Russia
3 Mathematical Center, Lobachevsky University, 603950 Nizhni Novgorod, Russia
4 Adv Learning Systems, TDAA, Intel, Chandler, AZ 85226, USA; alexey.polovinkin@gmail.com
5 Department of Physics, University of Gothenburg, SE-41296 Gothenburg, Sweden
* Correspondence: meerov@vmk.unn.ru (I.M.); arkady.gonoskov@physics.gu.se (A.G.)

Abstract: When entering the phase of big data processing and statistical inferences in experimental physics, the efficient use of machine learning methods may require optimal data preprocessing methods and, in particular, optimal balance between details and noise. In experimental studies of strong-field quantum electrodynamics with intense lasers, this balance concerns data binning for the observed distributions of particles and photons. Here we analyze the aspect of binning with respect to different machine learning methods (Support Vector Machine (SVM), Gradient Boosting Trees (GBT), Fully-Connected Neural Network (FCNN), Convolutional Neural Network (CNN)) using numerical simulations that mimic expected properties of upcoming experiments. We see that binning can crucially affect the performance of SVM and GBT, and, to a less extent, FCNN and CNN. This can be interpreted as the latter methods being able to effectively learn the optimal binning, discarding unnecessary information. Nevertheless, given limited training sets, the results indicate that the efficiency can be increased by optimizing the binning scale along with other hyperparameters. We present specific measurements of accuracy that can be useful for planning of experiments in the specified research area.

Keywords: laser physics; artificial neural networks; fully-connected neural networks

Citation: Rodimkov, Y.; Efimenko, E.; Volokitin, V.; Panova, E.; Polovinkin, A.; Meyerov, I.; Gonoskov, A. ML-Based Analysis of Particle Distributions in High-Intensity Laser Experiments: Role of Binning Strategy. *Entropy* **2021**, *23*, 21. https://dx.doi.org/10.3390/e23010021

Received: 11 December 2020
Accepted: 22 December 2020
Published: 25 December 2020

1. Introduction

In many experimental studies, the absence of clearly interpretable features in the measured data leads to the necessity of solving inverse problems for revealing the underlying properties of explored physical systems. Nevertheless, the problem may be intractable due to probabilistic or stochastic nature of the studied process or due to the presence of latent parameters that are not known within a necessary accuracy. These difficulties can be circumvented by the use of big data acquisition followed by statistical analysis carried out with the help of machine learning (ML) [1,2]. One way of arranging this is to develop a computational model that can reproduce the experimental data with sufficient accuracy for any given values of the unknown latent variables and the parameters that quantify the properties to be explored in an experiment. Using this computational model, we can generate possible outcomes of many experiments for various values of input parameters and train a machine learning algorithm to guess the values of these input parameters based on the simulated outcome. Even in the case of the probabilistic nature of the simulated process, the outcome may contain patterns that are sufficiently prominent to be learned and used by the ML model to unambiguously determine some of the latent parameters from the data measured in the experiment. Already this can be a crucial simplification for

interpreting experimental results and for obtaining heuristic conclusions (see, for example, [3]). In less certain conditions and/or for more rigorous assessments, one can use the trained ML model as a generator that can dramatically increase the convergence rate of the approximate Bayesian computation (ABC) [4–9]. The application of the described routine can be useful in the experimental studies of strong-field quantum electrodynamics with the help of high-intensity lasers [10]. In many such experiments, beams of accelerated electrons collide with tightly focused laser pulses and the energy-angular distribution functions of the outgoing electrons and/or photons are measured [11,12]. Although some basic properties of certain processes can be studied via prominent features [13], the probabilistic nature of strong field quantum electrodynamics (SFQED) processes and uncontrollable (and unknown) variation of the interaction parameters (such as the impact parameter that quantifies the misalignment between the laser focus and the electron beam center) lead to the necessity of drawing statistical inferences from the data collected in a large series of experiments. ML methods can play an important role in the automatization of data processing for reinforcing not only experimental, but also theoretical studies [14].

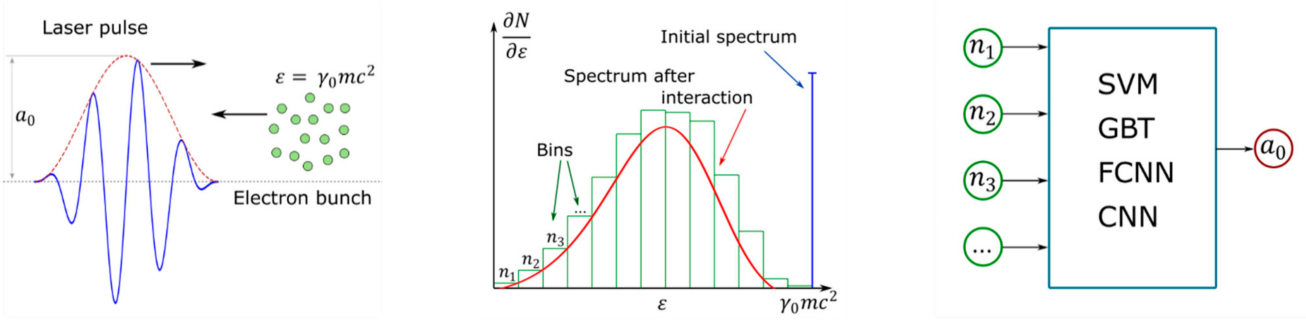
In this paper, we assess the factor of binning, which is applied as a preprocessing of the measured distribution of particles. The choice of small bins leads to an increased level of noise, whereas the use of large bins reduces the noise at the cost of losing information due to reduced resolution. Although one can apply more advanced strategies, such as principle component analysis (PCA) and spectral filtering, the choice of optimal bin size can be sufficient in some cases, whereas various ML methods can differ in terms of their tolerance to this aspect. The consideration of a simple uniform binning strategy can be advantageous in sophisticated conditions, whereas the use of an optimal ML model can mitigate the effect of non-optimized binning. We analyze this aspect using a simplified computational model, which is designed to mimic the properties of particle distribution in the upcoming experiments with high-intensity lasers.

2. Problem Statement

The problem considered in the present paper is a simplified yet descriptive model of a numerical experiment that is closely related to novel experiments on radiation reaction [11,12]. In these experiments, head-on collisions of a high-intensity laser pulse with a high energy electron beam was used to find the experimental evidence on how the radiation reaction affects the electron dynamics. Here we analyze the employment of machine learning techniques to the problem of identification of latent parameters in such experiments. One known example is the impact parameter, which can vary uncontrollably from shot to shot if the alignment is not controlled sufficiently well [11]. In case of such a misalignment, the particles of the beam propagate aside of the pulse peak and effectively experience weaker electromagnetic fields. If we could identify the misalignment from the measured spectrum of electrons (or photons), we would be able to exclude unsuccessful shots and account for the misalignment in the remaining cases, thus making possible the further statistical analysis. To examine such a possibility, we model the effect of misalignment by the variation of the laser pulse amplitude in one-dimensional interaction process. Specifically, we aim to determine the laser pulse amplitude based on the spectra of an initially monoenergetic electron beam after interaction with this pulse in the presence of a quantum radiation reaction.

The schematic description of the numerical experiment is as follows. An ultra-intense laser pulse propagates through a counter-propagating monoenergetic electron bunch, see Figure 1a. In the strong-field region, the effects of SFQED lead to a notable probability for an electron to emit one or several photons, and these events cause the corresponding loss of its kinetic energy. The process of photon emission is probabilistic, and in a single act of emission the electron may emit a photon, carrying away an arbitrary part of its kinetic energy. After the interaction, the energy distribution (spectrum) of electrons in the bunch has a finite width with a shift to lower energies with respect to the initial energy, see Figure 1b. To quantify electron spectra in a form suitable for a machine learning

task, the full energy range from zero up to the initial energy is split into a number of bins, and the number of electrons in each bin is calculated. The resulting histogram representing the energetic spectrum of electrons is used as an input vector for the machine learning regression task, see Figure 1c. The problem of obtaining the pulse amplitude is solved by means of different machine learning techniques including fully connected and convolutional neural networks.



(**a**) (**b**) (**c**)

Figure 1. Schematic of numerical experiment. (**a**) Head-on collision of an ultra-intense laser pulse with an electron bunch. (**b**) Electron spectrum modification and binning to produce a resulting spectrum histogram. (**c**) A histogram serves as an input for different ML methods used to determine dimensionless amplitude of the laser pulse a_0.

The interaction of the laser pulse with the electron bunch can be modeled by the following system of equations. The dynamics of electromagnetic field is governed by the Maxwell equations:

$$\begin{array}{c} \frac{\partial \vec{E}}{\partial t} = -4\pi \vec{j} + c\nabla \times \vec{B} \\ \frac{\partial \vec{B}}{\partial t} = -c\nabla \times \vec{E} \end{array}, \tag{1}$$

where $\vec{E}$, $\vec{B}$ are the electric and magnetic fields, respectively, $\vec{j}$ is the current density, and c is the speed of light.

The motion of articles is described by Newton's law of motion:

$$\begin{array}{c} \frac{\partial \vec{p}}{\partial t} = \vec{F}_L + \vec{F}_{RR};\ \vec{F}_L = e\cdot\left(\vec{E} + \frac{1}{c}\vec{v} \times \vec{B}\right) \\ \frac{\partial \vec{r}}{\partial t} = \vec{v} = \frac{\vec{p}}{m}\left(1 + \frac{p^2}{m^2c^2}\right)^{-1/2}, \end{array} \tag{2}$$

where $\vec{r}$, $\vec{p}$, $\vec{v}$ are the position, momentum, and velocity of the particle, m, e are its mass and charge, respectively, $\vec{F}_L$ is the Lorentz force due to electromagnetic field acting on the particle. The term $\vec{F}_{RR}$ provides the semiclassical description of the radiation reaction by instantaneous changes of momentum (the recoil due to photon emission) that occur probabilistically with the rate computed within SFQED (see, for example, [15]).

The scheme of the numerical experiment is close to the one used in Ref. [16]. Initially, the electrons in the bunch have the energy $\varepsilon_0 = mc^2\gamma_0$, where $\gamma_0 = 1000$ is the relativistic Lorentz-factor. The number of electrons in the bunch is varied in the experiments from 25 to 100,000. As the number of electrons in the bunch is sufficiently small, we neglect all types of their influence on the laser beam, such as the refraction and the depletion of the electromagnetic fields of the pulse. We also neglect the direct electron–electron interactions inside the bunch. With these simplifications we can consider the interaction between each electron and the laser pulse independently.

For simplicity, the one-dimensional problem is studied and the laser field is set as a short laser pulse propagating in the x direction:

$$\mathrm{E_y} = -\mathrm{B_z} = a_0 E_{\mathrm{rel}} \sin^2\left(\pi \frac{x + \mathrm{ct}}{\lambda}\right) \sin\left(2\pi \frac{x + \mathrm{ct}}{\lambda}\right), \tag{3}$$

where $\lambda = 1\ \mu m$ is the laser wavelength, and a_0 is the dimensionless amplitude in relativistic units $E_{\mathrm{rel}} = \frac{2\pi \mathrm{mc}^2}{\lambda e}$. The pulse is evolved according to Equation (3) over the total simulation time $T = L/c$ with the number of time steps equal to 100. The dimensionless amplitude a_0 is varied from 10 to 1000. This covers a wide range of intensities from 10^{20} W/cm^2, where radiation losses are weak and radiative friction can be treated classically, up to 10^{24} W/cm^2, where radiative friction becomes essentially probabilistic. In the latter case, the electrons can lose a major part of their energy, and a significant spectrum broadening is observed.

The described problem is modeled using the Hi-Chi open-source framework [17]. The photon emission and electron recoil are accounted for in the following way. On each time step for each electron, we generate a uniformly distributed value $\delta = \frac{\hbar\omega}{mc^2\gamma}$, which is the ratio of photon energy to the full energy of the original particle $\varepsilon = mc^2\gamma$, and then we sample the new photon with probability density $P(\delta)$:

$$P(\delta) = \left[\Delta t \frac{e^2 mc}{\hbar^2}\right] \cdot \frac{\sqrt{3}}{2\pi} \cdot \frac{\chi}{\gamma} \cdot \frac{1-\delta}{\delta} \cdot \left(F(z_q) + \frac{3}{2}\delta\chi z_q G(z_q)\right), \tag{4}$$

where $\hbar$ is the reduced Planck constant, Δt is the time step, $F(x)$ and $G(x)$ are the first and second synchrotron functions, $z_q = \frac{2}{3}\chi^{-1}\frac{\delta}{1-\delta}$, and $\chi \equiv \frac{e\hbar}{m^3c^4}\sqrt{\left(\frac{\vec{\varepsilon E}}{c} + \vec{p} \times \vec{B}\right)^2 - \left(\vec{p}\cdot\vec{E}\right)^2}$ is a dimensionless parameter characterizing the transverse acceleration of the particle in the field. For electrons, this parameter can be calculated as $\chi = \gamma\frac{H_{eff}}{E_s}$, where $E_s = \frac{m^2c^3}{e\hbar}$ is the Schwinger field and H_{eff} is the effective field that acts on the particle. The generated photon is assumed to have the same direction of propagation as the parent particle. An electron's momentum and energy are updated accordingly. We consider laser field intensities achievable on existing laser facilities. We neglect the effect of Breit–Wheeler pair production. We also assume that the electron bunch duration is sufficiently short so that we can neglect the interactions of emitted photons with the electron bunch after they have been emitted. After all electrons have interacted with the laser pulse, we retrieve the electron energy distribution for the given amplitude a_0. Since the process of photon emission is probabilistic and for a small number of electrons the energy distribution can be noisy, we collected several realizations for each a_0.

In the next stage, we divided the whole energy range from 0 to $mc^2\gamma_0$ into a number of bins. For each realization, we counted the number of electrons in each bin, denoted as n_i for the i-th bin in Figure 1b,c. We generated a dataset consisting of a vector of n_i as a feature vector and a_0 as a label, and trained ML models using Support Vector Machine (SVM), Gradient Boosting Trees (GBT), Fully-Connected Neural Network (FCNN), and Convolutional Neural Network (CNN) on generated data to solve the regression problem of estimating a_0 based on the histogram of electrons' energy spectra. The aim of this paper is to examine the role of the binning strategy, so the accuracy of numerical methods was investigated with respect to the combination of the number of bins and the number of electrons per bin. After dimensionality reduction by means of the principal component analysis method and fine-tuning, we found the most relevant model and analyzed its results.

3. Methods

3.1. Hi-Chi Project Overview

The project High-Intensity Collisions and Interactions (Hi-Chi) is an open-source collection of Python-controlled tools for performing simulations and data analysis in the research area of strong-field particle and plasma physics. The project is intended to offer an environment for testing, benchmarking, and aggregative use of individual components, ranging from basic routines to supercomputer codes. The components are being developed in C++ and optimized for state-of-the-art high-performance CPUs. In this way, the project combines the flexibility of Python and the efficiency of resource-intensive computations at the C++ level, achieving high performance using either desktops or supercomputers.

A high-level architecture of the project is depicted in Figure 2. The project's architecture is designed as an independent set of primitives and modules that can interact with each other. Currently, there are two types of modules: (I) Working with an electromagnetic field and (II) interacting with ensembles of particles. Modules of the first type include finite-difference time-domain (FDTD) [18] and spectral (PSTD, PSATD) field solvers [19–22], several implementations of boundary conditions (periodic, PML, field generator), transformations of electromagnetic field (rotation, shift, scaling, etc.). Modules of the second type include several particle motion equations solvers (e.g., the Boris method), a number of particle resampling methods (various particles thinning and merging techniques [23]), and a module taking into account quantum electrodynamic effects (the QED module) [15]. Each module interacts with relevant primitives. Thus, the field solvers are associated with collocated and staggered grids capable of performing field interpolation at any point of a computational domain. For this purpose, the CIC and TSC form factors are currently supported. The particle pushers work with ensembles of particles which are stored employing the Structure of Arrays (SoA) or Array of Structures (AoS) patterns. All C++ classes and objects are exported from C++ to Python by means of the pybind11 software [17].

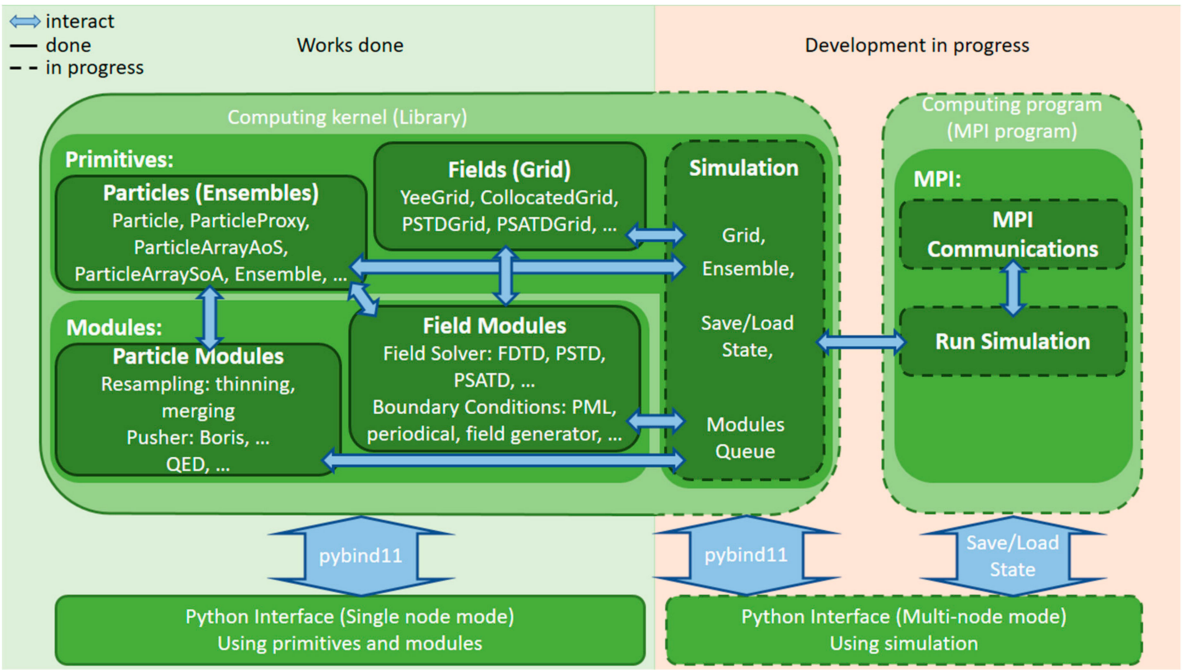


Figure 2. The interaction scheme of the High-Intensity Collisions and Interactions (Hi-Chi) modules.

The Hi-Chi implementation is based on the experience of the development of the high-performance plasma simulation PICADOR code [24,25] and currently employs shared-memory parallelism using the OpenMP technology. Main computational kernels are optimized for state-of-the-art CPUs including vectorization and parallelization of performance-critical computational loops, cache optimizations, and NUMA (Non-Uniform Memory

Access) [26] optimizations. The code is under ongoing modifications and improvement. One of the main directions of further development is the creation of a distributed version of the code that allows you to utilize a supercomputer through the use of MPI technology. Note that the interaction of Python and C++ in a distributed mode is not straightforward. However, Python and C++ modules can interact, saving and loading states in the file system. For this purpose, the user-defined configuration can be saved in the file system and a chosen number of MPI processes will be launched. Then, each process can download the configuration from the file system, perform calculations, and save the final results. The results can be further read and processed by a Python-based control program. The distributed version of the code is under development. The code is publicly available (see Supplementary Materials section for the details).

3.2. Data Generation

We collected data as follows. Firstly, we performed numerical simulations with the peak amplitude a_0 in the range $[10;\ 1000]$ and with $N = 100,000$ electrons up to time $T = L/c$, integrating the electron motion equations and taking into account the QED effects. The resulting data array, hereinafter referred to as DATA, contained N energies for each a_0. It was used to randomly sample the resulting values with their subsequent aggregation into N_b bins. After sampling, all values were normalized to the range $[0; 1]$ to improve the performance of training machine learning models.

Secondly, we used the DATA array to train several machine learning models and test how their accuracies depend on the number of electrons involved in the numerical simulations. In this regard, we fixed different values of the number of bins N_b and the average number of electrons per bin N_e and randomly selected $N_b \times N_e$ electrons from the DATA array. The values N_b and N_e varied in the range $[5;\ 2000]$, while the total number of electrons varied in the range $[25;\ 20,000]$. All samples were taken without repetitions. When forming the training dataset, the specified procedure was performed three times, while at the stage of creating the validation and test samples it was done only once.

3.3. Machine Learning Techniques

We evaluated and compared several state-of-the-art supervised machine learning algorithms to solve the regression problem for the estimation of a_0 based on the histogram of electron spectra.

Support vector regression machine [27] (evolution of support vector machine (SVM) [28] for classification problems) is a powerful algorithm that can balance tolerance to the errors, both through setting an acceptable error margin and through tuning the cost of falling outside this acceptable error margin. One of the main SVM advantages is the use of kernels for learning linear predictors in high dimensional feature spaces that allows us to handle high-dimensional problems effectively.

Gradient boosting trees (GBT) [29] is an ensemble of decision trees [30] where every new tree is built using the data from previously learnt trees. At each iteration of GBT, a new tree is fitted to the generalized residuals with respect to a loss function. The GBT algorithm can deal with both classification and a regression problem, works with mixed type data, effectively processes missing data, and is invariant to monotonic transformations of the input variables. All these factors make GBT one of the most accurate and universal supervised machine learning algorithms.

Neural networks and their applications have been widely developed recently due to explosive growth of computational capabilities and accumulation of a large amount of data necessary for effective training of these models. According to Cybenko theorem [31], a feed-forward neural network with one hidden layer can approximate any continuous function of many variables with any given precision. In recent studies, in particular [32], it has been proven that any Lebesgue integrable function of many variables can be approximated by a fully connected neural network with ReLU activations. In this work, we also consider convolutional neural networks [33] that consider local special data dependencies.

4. Experimental Results

4.1. Methodology

The experimental part of the paper is as follows. Firstly, we run some preliminary experiments to determine the appropriate hyperparameter values for each of the machine learning methods used (SVM, GBT, CNN, FCNN). Having temporarily fixed these parameters, we empirically investigate how the accuracy of solving the problem depends on the number of bins and electrons involved in the numerical simulation. We consider from 5 to 2000 bins and from 5 to 2000 electrons per bin. For each point in the $\{N_b; N_e\}$ parameter space, we train ML models. Stopping the training of neural networks is carried out based on the error in the validation dataset, and the accuracy is estimated using the test dataset. Realizing that the chosen "generic" hyperparameter values may not be optimal, we selectively examine some configurations $\{N_b;\ N_e\}$ by manually adjusting the hyperparameters. Indeed, experiments show that accuracy can be improved by fine-tuning, but we did not find any dramatic changes.

The main idea behind the series of experiments described above is to gain an intuition as to how accurately specific machine learning methods can solve a given problem, to understand which of them are most promising for further tuning, and also to establish how stable the results are when the number of electrons and bins decreases. Based on these experiments, we choose the most promising configurations $\{N_b;\ N_e\}$ and investigate them in more detail, adjusting the parameters to improve the results.

Finally, we examine the feasibility of feature selection and dimensionality reduction techniques. The feature selection does not lead to an improvement in the results, while the dimensionality reduction employing the principal component method makes it possible to reduce the number of features and simplify the architecture of the artificial neural networks, with relevant accuracy.

4.2. Results and Discussion

4.2.1. How Accuracy of ML Models Depends on the Number of Bins and the Number of Electrons per Bin?

Firstly, we performed massive experiments to establish how the accuracy of reconstruction of the peak amplitude of a laser pulse depends on the parameters $\{N_b;\ N_e\}$. Given that a full consideration of all relevant combinations of hyperparameters for four machine learning methods for each pair $\{N_b;\ N_e\}$ would require huge computational resources, we performed preliminary experiments for some pairs, and then fixed the parameters as follows. We employed the XGBRegressor method from the XGBoost library [34] and the SVR method from the scikit-learn library [35] as the implementation of the GBT and SVM methods, respectively. In the GBT method, we used 110 trees of depth 5, the learning rate was set to 6×10^{-2} [36]. In the SVM method, we used the radial basis function (RBF) kernel, the epsilon was equal to 1×10^{-3} [37]. The default values were used for the rest of the parameters.

The parameters of neural networks in the CNN and FCNN methods were selected by optimizing the error on the validation set taking into account the dimension of the input vector. We employed the following architectures and considered them in the specified ranges of hyperparameters (the selected optimal parameters are detailed in Section 4.2.2): FCNN with 3–5 hidden layers, CNN with 1–6 convolution layers with a kernel of size 3 at the beginning, and 2–4 fully connected hidden layers at the end. The numbers of neurons in the fully-connected layers were taken from the range 4–200. We used the Adam optimizer from the Keras framework [38] with default parameters and the ReLU activation function. The numbers of neurons in each layer were selected based on the dimension of the input data. For different pairs $\{N_b;\ N_e\}$ the architectures and parameters of the neural networks could be slightly different in order to improve the accuracy. Further, for the most promising combinations $\{N_b;\ N_e\}$ we fine-tuned the hyperparameters for all the methods used. The best found configurations are given in Section 4.2.2.

Figure 3 shows how four ML methods reconstruct the peak amplitude of a laser pulse depending on the number of bins N_b and the number of electrons N_e, while $N_b \in [5;\ 2000]$, $N_b \in [5;\ 2000]$, and the total number of electrons $N_b \times N_e$ varies in the range [25; 20,000]. The results show, as expected, that an increase in the number of electrons usually leads to a decrease in the error. We can also compare the methods and conclude how the parameters $\{N_b;\ N_e\}$ should be chosen.

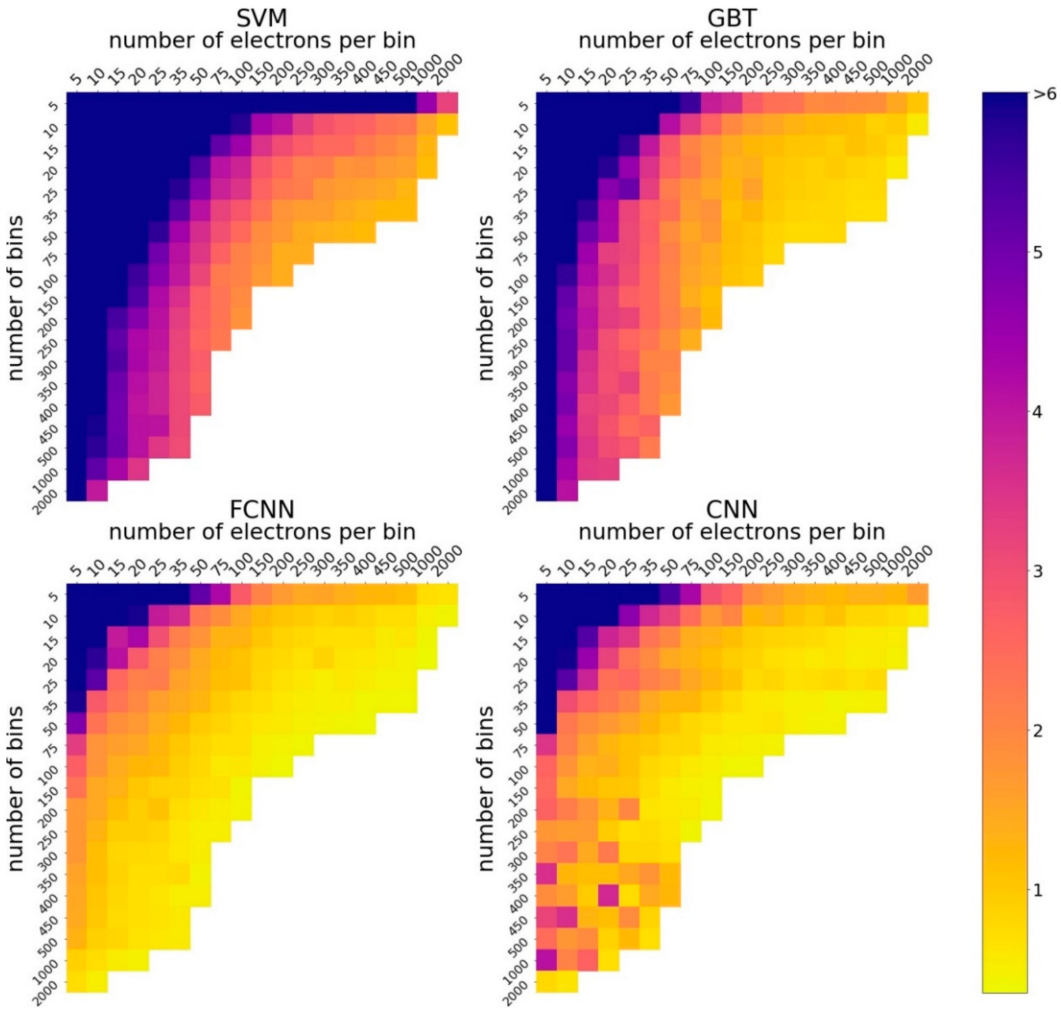


Figure 3. Heat maps demonstrate how the accuracy of the Support Vector Machine (SVM), Gradient Boosting Trees (GBT), Fully-Connected Neural Network (FCNN), Convolutional Neural Network (CNN) methods in reconstructing the peak amplitude of a laser pulse depends on the number of bins and the number of electrons per bin. Accuracy is given as a percentage of the mean relative error. Blue squares correspond to a large error, yellow squares to a small error.

The FCNN demonstrates perfect stability in terms of accuracy when a reasonable configuration is chosen, even with the fixed network architecture and parameters. The CNN shows good accuracy, but the results seem less stable. We observe that for a small number of electrons and a large number of bins, the accuracy varies over a wide range, even with a small change in the parameters. The SVM and GBT methods are inferior in accuracy to neural networks, but still show reasonable results.

Next, we fix the relevant number of electrons in a numerical experiment, equal to 10,000, and analyze how the error changes when the number of bins increases (Figure 4). It turned out that for the GBT method and FCNN, the optimal number of bins is equal to 20. For the SVM method, it is equal to 10, but the accuracy for 10 bins only slightly exceeds the accuracy for 20 bins. Thus, the considered methods work best with approximately the

same small number of bins. For the CNN, 100 bins are optimal. Based on these results, we fine-tuned the models. The results obtained in this case, as well as the optimal parameter configurations, are described below.

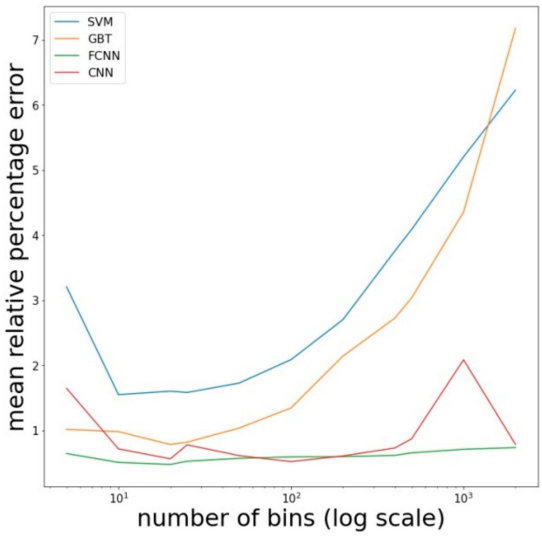


Figure 4. The dependence of the mean relative percentage error for the four considered machine learning methods (SVM, GBT, FCNN, CNN) on the number of bins used to build the histogram. The number of electrons in the experiment is equal to 10,000.

4.2.2. Optimal Configuration of the Parameters

This section describes the hyperparameters of the models in their best configurations. Firstly, empirically optimal parameters for GBT and SVM methods are considered. We tune the parameters of cross-validated XGBRegressor method from the XGBoost library [34] and the SVR method from the scikit-learn library [35] as the implementation of the GBT and SVM methods, respectively. We found that the GBT method performed best when using 110 trees with maximum tree depth equal to 6 and a learning rate of 0.1, without regularization [36]. The SVM method showed the best results when using the radial basis function (RBF) kernel, the L2 regularization with parameter 30, and the epsilon equal to 3×10^{-4} [37]. The default values were used for other parameters.

Secondly, we customize the architecture and parameters of artificial neural networks. We employ a fully-connected model with 5 hidden layers. The first hidden layer contains 100 neurons with the ReLU activation function, followed by a layer with 75 neurons and the sigmoid activation function. The last three hidden layers use the ReLU activation function and contain 64, 16, and 4 neurons, respectively. The model was trained for 1420 epochs, with the Adam optimizer [39] with the learning rate of 1×10^{-3}. By analogy with FCNN, various options for combining layers with different numbers of neurons were considered for CNN. We employ two convolutional layers containing 1 and 3 convolutions, respectively, followed by a pooling layer with the size of 2. Further, the same combination of layers was used with the difference that the number of kernels was set equal to 3 and 9, respectively. For all convolutional layers, the convolution size is 3, with the ReLU activation function. Further, 4 fully connected layers are used, containing 96, 64, 16, and 4 neurons with the following activation functions: Sigmoid, sigmoid, ReLU, and ReLU, respectively. The model was trained for 1520 epochs. We used the Adam optimizer with the learning rate of 3×10^{-4}.

Then, we employ the PCA method from the scikit-learn library [40]. We found that the first 5 principal components explain 98 percent of the variance in the original data. After that, we customize a fully connected neural network with 5 hidden layers. The first 3 layers contain 10 neurons, followed by 2 layers with 8 and 4 neurons, respectively. The ReLU activation functions are used. The neural network was trained for 2800 epochs. We

used the Adam optimizer with a learning rate of 6×10^{-4}. All models were trained in batches of 32 objects. For training, the mean absolute error was used. The Adam optimizer used the default parameters from the Keras framework [38], except for the learning rate parameter, the values of which are given above.

4.2.3. Final Comparison

The results of a comparative analysis of models created by machine learning methods for the optimal configurations of parameters are shown in Table 1. It turned out that in most cases, fine-tuning of the hyperparameters of the methods led to some increase in accuracy. At the same time, the achieved gain is not dramatic, which indicates that it is enough to choose the reasonable values of the parameters. Experiments have shown that artificial neural networks solve the problem of reconstructing the peak amplitude of a laser pulse with sufficiently higher accuracy. The SVM method loses out to deep learning methods by about a factor of two in terms of the average absolute and average relative error. The GBT method shows accuracy close to that of neural networks. However, unlike artificial neural networks, the GBT and SVM methods, with a small number of objects, can yield an error of 5–10%, which can be critical for practical use. The PCA method allowed us to reduce the size of the network and decrease the run time while maintaining a reasonable accuracy of the amplitude reconstruction. We applied this method to data for a fully connected neural network and selected 5 principal components, on which another fully connected neural network was trained. New features are not correlated, which also improves the neural network training procedure. The new data explains 98 percent of the variance in the original data.

Table 1. Accuracy of the fine-tuned machine learning methods for solving the peak amplitude reconstruction problem with 10,000 electrons for one feature vector.

Measure	SVM	GBT	FCNN	CNN	PCA+FCNN
Mean absolute error	4.050	2.453	1.784	1.827	2.000
Mean relative percentage error	1.062	0.661	0.512	0.496	0.709
Coefficient of determination	0.99930	0.99967	0.99993	0.99992	0.99991

Figure 5 shows the correlation between exact and predicted values for a fully connected neural network. The points are almost perfectly fitted by the linear function $y = x$, shown in red, which corresponds to the close to 1 value of the coefficient of determination. The rest of the methods show similar results (Table 1).

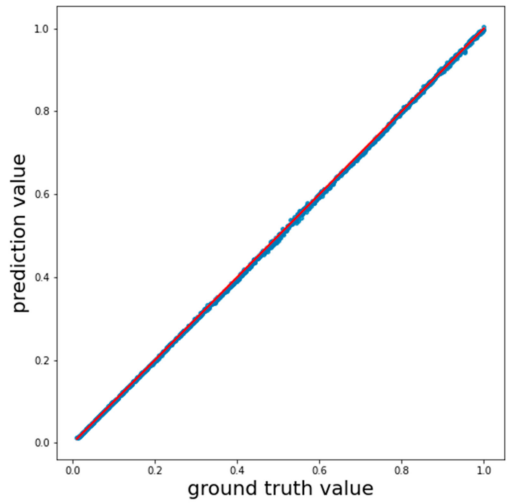


Figure 5. Correlation of the exact and predicted values when using the FCNN model. Points correspond to pairs of exact and predicted values. The red line is the linear function $y = x$.

Lastly, we run a *t*-test to compare ML models in their optimal configurations in terms of accuracy. To do this, we combined training and test samples for the fine-tuned ML models. Next, we randomly divided the obtained data into new training and test samples 10 times and calculated the accuracy of the models. The results are presented below (Figure 6). The ML models were further sorted by accuracy and compared by the paired *t*-test (the most accurate model was compared with the second one, the second model with the third one, and so on). As a null hypothesis, it was assumed that the methods are indistinguishable in accuracy. The *p*-value was equal to 0.05. The *t*-test results showed that FCNN, CNN methods work better on this problem than GBT, and SVM shows the worst result.

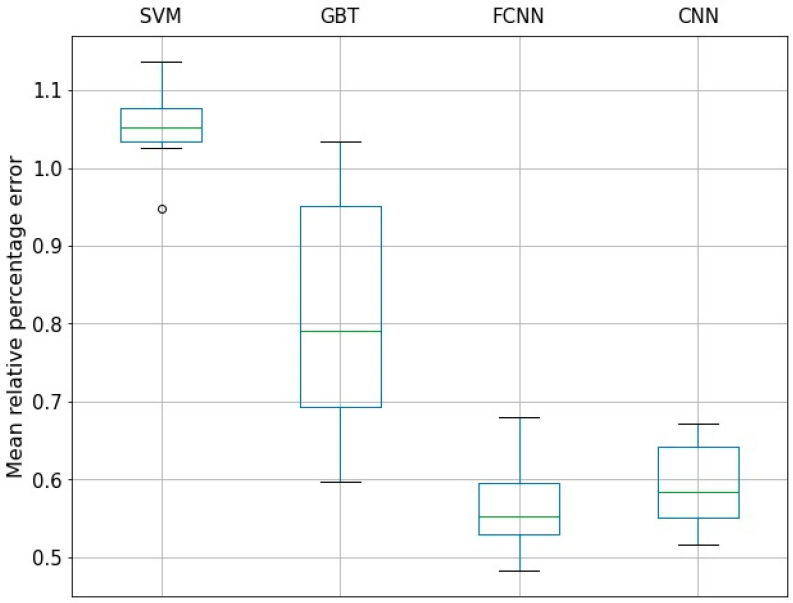


Figure 6. Distribution of mean relative percentage errors for 4 ML methods: SVM, GBT, FCNN, CNN.

5. Conclusions

In this work, we considered the effect of binning strategy on the accuracy of several ML models applied to a test problem that models the needs of the upcoming experiments on the SFQED effects. We varied the size of bins used for the construction of the input vector from the energy spectra that can presumably be measured with high resolution. The limit of small bins (i.e., large input vectors) corresponds to a high level of noise, whereas the use of large bins (i.e., small input vectors) implies the loss of information. The results indicate that SVM and GBT are more sensitive to the choice of the bin size than FCNN and CNN, but all the considered ML models can be configured to achieve a reasonably good accuracy in our tests. The studies carried out do not guarantee the success of solving more complex problems. However, they show the prospects for continuing work in this direction. In the future, we plan to consider problems closer to state-of-the-art physical experiments based on the experience gained.

One of the potential directions for further development is the use of new approaches to dimensionality reduction, in particular, non-linear PCA options based on principal manifolds [41]. We also plan to pay special attention to the issues of reliability and explainability of the results obtained using artificial neural networks. We believe that these questions are extremely important for planning future experiments. In the model problem considered in this paper, we see that FCNN shows good accuracy with an appropriate binning strategy in a wide range of parameters. However, the question of whether this effect will persist in more complex problems remains open.

Supplementary Materials: The Hi-Chi project is available online at https://github.com/hi-chi/pyHiChi. The data and scripts required to reproduce the numerical results may be downloaded from https://github.com/hi-chi/Machine-Learning (the relevant examples are located in the "Amplitude Reconstruction" folder).

Author Contributions: Conceptualization, E.E., I.M. and A.G.; methodology, A.G., I.M., A.P.; software, Y.R., V.V.; validation, Y.R., A.P.; formal analysis, A.P., A.G.; investigation, Y.R., A.G., A.P., E.P.; resources, I.M., A.G.; data curation, Y.R.; writing—original draft preparation, Y.R., E.E., E.P., V.V., A.P., I.M., A.G.; writing—review and editing, Y.R., E.E., V.V., A.P., I.M., A.G.; visualization, Y.R.; supervision, A.G.; project administration, I.M.; funding acquisition, I.M., A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Higher Education of the Russian Federation, agreement number 075-15-2020-808.

Data Availability Statement: The data generated and analyzed in this study are publicly available in https://github.com/hi-chi/Machine-Learning (the relevant examples are located in the "Amplitude Reconstruction" folder).

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

References

1. Mehta, P.; Bukov, M.; Wang, C.H.; Day, A.G.; Richardson, C.; Fisher, C.K.; Schwab, D.J. A high-bias, low-variance introduction to machine learning for physicists. *Phys. Rep.* **2019**, *810*, 1–124. [CrossRef] [PubMed]
2. Carleo, G.; Cirac, I.; Cranmer, K.; Daudet, L.; Schuld, M.; Tishby, N.; Vogt-Maranto, L.; Zdeborová, L. Machine learning and the physical sciences. *Rev. Mod. Phys.* **2019**, *91*, 045002. [CrossRef]
3. Gonoskov, A.; Wallin, E.; Polovinkin, A.; Meyerov, I. Employing machine learning for theory validation and identification of experimental conditions in laser-plasma physics. *Sci. Rep.* **2019**, *9*, 7043. [CrossRef] [PubMed]
4. Rubin, D.B. Bayesianly justifiable and relevant frequency calculations for the applies statistician. *Ann. Stat.* **1984**, *12*, 1151–1172. [CrossRef]
5. Beaumont, M.A.; Zhang, W.; Balding, D.J. Approximate Bayesian computation in population genetics. *Genetics* **2002**, *162*, 2025–2035. [PubMed]
6. Marjoram, P.; Molitor, J.; Plagnol, V.; Tavaré, S. Markov chain Monte Carlo without likelihoods. *Proc. Natl. Acad. Sci. USA* **2003**, *100*, 15324–15328. [CrossRef] [PubMed]
7. Sisson, S.A.; Fan, Y.; Beaumont, M. *Handbook of Approximate Bayesian Computation*; Sisson, S.A., Fan, Y., Beaumont, M.A., Eds.; CRC Press: Boca-Raton, FL, USA, 2019.
8. Alsing, J.; Wandelt, B.; Feeney, S. Massive optimal data compression and density estimation for scalable, likelihood-free inference in cosmology. *MNRAS* **2018**, *477*, 2874–2885. [CrossRef]
9. Charnock, T.; Lavaux, G.; Wandelt, B.D. Automatic physical inference with information maximizing neural networks. *Phys. Rev. D* **2018**, *97*, 083004. [CrossRef]
10. di Piazza, A.; Müller, C.; Hatsagortsyan, K.Z.; Keitel, C.H. Extremely high-intensity laser interactions with fundamental quantum systems. *Rev. Mod. Phys.* **2012**, *84*, 1177. [CrossRef]
11. Cole, J.M.; Behm, K.T.; Gerstmayr, E.; Blackburn, T.G.; Wood, J.C.; Baird, C.D.; Duff, M.J.; Harvey, C.; Ilderton, A.; Joglekar, A.S.; et al. Experimental evidence of radiation reaction in the collision of a high-intensity laser pulse with a laser-wakefield accelerated electron beam. *Phys. Rev. X* **2018**, *8*, 011020. [CrossRef]
12. Poder, K.; Tamburini, M.; Sarri, G.; di Piazza, A.; Kuschel, S.; Baird, C.D.; Behm, K.; Bohlen, S.; Cole, J.M.; Corvan, D.J.; et al. Experimental signatures of the quantum nature of radiation reaction in the field of an ultraintense laser. *Phys. Rev. X* **2018**, *8*, 031004. [CrossRef]
13. Harvey, C.N.; Gonoskov, A.; Ilderton, A.; Marklund, M. Quantum quenching of radiation losses in short laser pulses. *Phys. Rev. Lett.* **2017**, *118*, 105004. [CrossRef] [PubMed]
14. Kim, Y.J.; Lee, M.; Lee, H.J. Machine learning analysis for the soliton formation in resonant nonlinear three-wave interactions. *J. Korean Phys. Soc.* **2019**, *75*, 909–916. [CrossRef]
15. Gonoskov, A.; Bastrakov, S.; Efimenko, E.; Ilderton, A.; Marklund, M.; Meyerov, I.; Muraviev, A.; Sergeev, A.; Surmin, I.; Wallin, E. Extended particle-in-cell schemes for physics in ultrastrong laser fields: Review and developments. *Phys. Rev. E* **2015**, *92*, 023305. [CrossRef] [PubMed]
16. Arran, C.; Cole, J.M.; Gerstmayr, E.; Blackburn, T.G.; Mangles, S.P.D.; Ridgers, C.P. Optimal parameters for radiation reaction experiments. *Plasma Phys. Control. Fusion* **2019**, *61*, 074009. [CrossRef]
17. Hi-Chi Project. Available online: https://github.com/hi-chi/pyHiChi (accessed on 5 December 2020).
18. Taflove, A.; Hagness, S.C. *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, 3rd ed.; Artech house: Boston, MA, USA, 2005.

19. Liu, Q.H. The PSTD algorithm: A time-domain method requiring only two cells per wavelength. *Microw. Opt. Technol. Lett.* **1997**, *15*, 158–165. [CrossRef]
20. Haber, I.; Lee, R.; Klein, H.; Boris, J. Advances in electromagnetic simulation techniques. In Proceedings of the Sixth Conference on Numerical Simulation of Plasmas, Berkeley, CA, USA, 16–18 July 1973; pp. 46–48.
21. Vay, J.L.; Haber, I.; Godfrey, B.B. A domain decomposition method for pseudo-spectral electromagnetic simulations of plasmas. *J. Comput. Phys.* **2013**, *243*, 260–268. [CrossRef]
22. Lehé, R.; Vay, J.L. Review of spectral maxwell solvers for electromagnetic particle-in-cell: Algorithms and advantages. In Proceedings of the 13th International Computational Accelerator Physics Conference, Key West, FL, USA, 20–24 October 2018; pp. 345–349.
23. Muraviev, A.; Bashinov, A.; Efimenko, E.; Volokitin, V.; Meyerov, I.; Gonoskov, A. Strategies for particle resampling in PIC simulations. *arXiv* **2020**, arXiv:2006.08593.
24. Surmin, I.A.; Bastrakov, S.I.; Efimenko, E.S.; Gonoskov, A.A.; Korzhimanov, A.V.; Meyerov, I.B. Particle-in-Cell laser-plasma simulation on Xeon Phi coprocessors. *Comput. Phys. Commun.* **2016**, *202*, 204–210. [CrossRef]
25. Surmin, I.; Bastrakov, S.; Matveev, Z.; Efimenko, E.; Gonoskov, A.; Meyerov, I. Co-design of a particle-in-cell plasma simulation code for Intel Xeon Phi: A first look at Knights Landing. In *Lecture Notes in Computer Science, Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing, Granada, Spain, 14–16 December 2016*; Springer: Cham, Switzerland, 2016; Volume 10049, pp. 319–329. [CrossRef]
26. Hager, G.; Wellein, G. *Introduction to High Performance Computing for Scientists and Engineers*; CRC Press: Boca-Raton, FL, USA, 2010.
27. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, New York, NY, USA, 27–29 July 1992; pp. 144–152.
28. Drucker, H.; Burges, C.J.; Kaufman, L.; Smola, A.; Vapnik, V. Support vector regression machines. *Adv. Neural Inf. Process. Syst.* **1996**, *9*, 155–161.
29. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
30. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*; Wadsworth, Inc.: Monterey, CA, USA, 1984.
31. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control. Syst.* **1992**, *5*, 455. [CrossRef]
32. Lu, Z.; Pu, H.; Wang, F.; Hu, Z.; Wang, L. The expressive power of neural networks: A view from the width. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6231–6239.
33. Krizhevsky, A.; Sutskever, I.; Hinton, G. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
34. XGBoost Documentation. Available online: https://xgboost.readthedocs.io/ (accessed on 5 December 2020).
35. Scikit-Learn Documentation. Available online: https://scikit-learn.org/ (accessed on 5 December 2020).
36. XGBoost Documentation. Python API. Available online: https://xgboost.readthedocs.io/en/latest/python/python_api.html (accessed on 21 December 2020).
37. Scikit-Learn Documentation. Python API (SVR). Available online: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html (accessed on 21 December 2020).
38. Keras Documentation. Available online: https://keras.io/ (accessed on 5 December 2020).
39. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
40. Scikit-Learn Documentation. Python API (PCA). Available online: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html (accessed on 5 December 2020).
41. Gorban, A.; Kégl, B.; Wunsch, D.; Zinovyev, A. Principal manifolds for data visualization and dimension reduction. *Lect. Notes Comput. Sci. Eng.* **2008**, *58*. [CrossRef]