

# Hybridizing AI and Domain Knowledge in Nanotechnology: the Example of Surface Roughness Effects on Wetting Behavior

Antonios Stellas  
Department of Mathematics and  
Computer Science,  
Technical University of Eindhoven  
Netherlands

George Giannakopoulos  
Institute of Informatics and  
Telecommunications,  
NCSR Demokritos  
and  
SciFY P.N.P.C.  
Greece

Vassilios Constantoudis  
Institute of Nanoscience and  
Nanotechnology,  
NCSR Demokritos  
and  
Nanometrisis P.C.  
Greece

## ABSTRACT

In this paper, we propose a scheme for the hybridization of domain modeling and theoretical knowledge in nanotechnology with Artificial Intelligence(AI) techniques and evaluate the success of its application to predict the relationship between nanosurface morphology and wettability. We utilize domain knowledge consisting of two parts. The first part is a mathematical modeling based on the inverse Fourier transform for the generation of rough surfaces with Gaussian or non-Gaussian height distributions, characterized by their first moments (Rms, skewness, kurtosis) and the correlation lengths along x and y-axes. The second part lies in the assumption that the Wenzel scenario for wetting of rough surfaces holds where the critical parameter for contact angle determination is the roughness ratio  $r$ , defined as the ratio of true (active) area of the solid surface to the apparent (projected) area. By creating different types of surfaces with a variety of input parameters, we create a database linking surface roughness parameters to the ratio  $r$ . This database is used to train Machine Learning (ML) models and validate them appropriately. Specifically, we train deep, feed-forward neural networks and random forest models and validate them on a separate (held-out) test dataset. We investigate systematically the amount of input data needed to get accurate predictions on the test data. We also evaluate the importance of different input roughness parameters with respect to their effects on surface wettability. To this end, we study the weights that the learning AI models assigned to roughness parameters through training and discuss the findings with respect to experimental expectations.

## KEYWORDS

Wetting roughness, mathematical modeling, Wenzel model, contact angle, Machine learning, Artificial Intelligence, Nanotechnology, Rough Surfaces

## 1 INTRODUCTION

Nanostructuring plays a fundamental role in nanotechnology since it enables new properties and functionalities of material surfaces. In order to provide a quantitative link between the geometry of nanostructure morphologies and the induced surface properties, we first need to find the proper mathematical tools to describe surface

morphology. To this end, several parameters and metrics have been proposed for the quantitative characterization of nanostructure morphology. Some of them are more closely linked to the fabrication process, while others more directly fit to the critical property of a targeted application. For example, surfaces with stochastic morphologies (rough surfaces) are widely used in the strong modification of the wetting behaviour of materials. According to the first scenario (Wenzel model) for the impact of surface roughness on wetting and contact angle, the critical parameter of surface nanoroughness is the roughness ratio  $r$  [1], defined as the ratio of true (active) area of the solid surface to the apparent (projected) area [2]. On the other side, the fabrication of surfaces is more usually related with the surface Rms, correlation length or other surface height parameters. Furthermore, the measurement of the latter is more straightforward and accurate with respect to the full active surface area [3]. Therefore, an estimated function connecting fabrication parameter of full active area (and thus, the roughness ratio  $r$  and contact angle) to roughness parameters (Rms,  $\xi$  and other moments), can significantly help the fabrication parameter selection process.

Up until now, theoretical modelling and experiments have been used to quantify these links [3–6]. However, these methods are time-consuming and their results are limited to the specific cases they investigate. Could integrating AI techniques improve the time-efficiency of these methods, making them more applicable in an industrial environment? If yes, how can we incorporate domain knowledge coming from both modeling and experimental results in AI models to achieve a hybridization of both, improve the accuracy of results and the success of the AI predictions? By the term domain knowledge, we refer to the specific scientific area of used data which, in our case is nanoscience/nanotechnology.

During the last decade, several nanotechnology areas have started to benefit from AI techniques aiming for example, to predict the properties of new nanomaterials, to enhance microscopy results, to accelerate simulations, to link manufacturing conditions with nanostructure morphology and then with the properties and performance of the nanostructured devices [7]. Due to the scientific and technological nature of data in these applications, there is an increasing need to devise ways to match AI methods with the domain knowledge, as reported in the relevant theoretical and experimental works.

The overall aim of this paper is to propose a hybridization scheme to facilitate the synergy of data-centric methods with domain modelling and theoretical knowledge concerning the link of the morphology of nanostructured surfaces with their properties and functionalities. The key ideas of this approach are: a) to use properly designed modeling results to train and validate AI methods along with experimental data when they are available and b) to exploit the ability of ML techniques to reverse input and output so that the targeted design of a nanostructured product dictates the choice of nanostructure geometry and manufacturing conditions. We will focus on the first idea and we will apply it to evaluate the success of AI techniques hybridized with domain modeling results to predict the relationship between nanosurface morphology and wettability. More specifically, the contributions of the paper can be outlined as follows:

- An implementation of domain modeling results in AI techniques is realized for training and validation.
- A comparison of different AI models is performed based on the physical modeling results.
- A study on the numerosity of required data (rough surfaces), to train sufficiently accurate AI methods.
- An estimate of the relative importance of input roughness parameters on wetting behavior, supported by a discussion to feed domain decisions and evaluations.

The paper is structured as follows. We begin, in section 2, with a presentation of the related recent work and the need to investigate further the predictive capability of AI techniques in the nanotechnology applications and specifically wetting behavior. The mathematical modeling methodology for the generation of rough surfaces to train AI models as well as the AI techniques used in the paper is the subject of the section 3. Section 4 presents the results of AI techniques and their comparison. The paper closes with the summary in the final section 5.

## 2 RELATED WORK

The applications of AI to physical problems and especially materials science have been studied in many contexts during the last decade, creating a novel research framework termed "data-driven materials science" (Teng Zhou et al., 2019 [8]). Teng Zhou highlighted the new opportunities that a data-driven approach can provide in the study of materials and named it the 4th paradigm in materials science. The previous three paradigms are assumed to be the empirical, theoretical and computational ones respectively. In the framework of this data-driven materials science, several studies have achieved to make significant contributions to the design and development of new materials (Sutton, C. et al., 2019 [9]), the prediction of material properties (electronic, mechanic, thermal,...) (Schütt, K. T. et al., 2014 [10]) or the evaluation of the importance of manufacturing and structure parameters on material surface functionalities such as wetting behavior (Amir Kordijazi et al., 2020 [11]). Furthermore, a critical question in this framework has been the incorporation of the domain knowledge of materials science (theoretical concepts and laws, modeling and simulation results) coming from previous paradigms in the data-driven algorithms, to avoid significant errors in the provided predictions. To this end, Sutton, C. et al., (2019) and Schütt, K. T. et al., (2014) applied the data-driven science paradigm

scheme, using simulation data to train a ML algorithm to predict faster solid-state properties. On the other hand, in the work of M Aziar Raissi and George Karniadakis et al. in 2019 [12], we find an elaborated methodology in which, physical modeling assists the AI research by exploration of physics-informed neural network algorithms.

Regarding the prediction of wetting behaviour, Amir Kordijazi et al., (2020) used ML techniques to predict the water contact angle on surfaces of ductile iron. They used a set of experimental measurements with input parameters the material composition, droplet size, the surface grit size and roughness and the time of the exposure to the liquid. The authors also evaluated the importance of each input parameter on the value of contact angle and they justified the primary role of surface roughness determined by the grit size. However, it was not specified which aspect of surface roughness is more critical. Given that the roughness of surfaces is a complex multifaceted phenomenon characterized by a plethora of parameters, it is worth questioning the relative importance of roughness parameters on surface wetting behavior. In literature, one can find interesting results coming from both experimental and computational approaches exploring the impact of surface roughness parameters on contact angle and hysteresis [4].

In our work, we follow the data-driven approach of the 4th paradigm of science endowed by theoretical and computational modelling knowledge of the 2nd and 3rd paradigms. The aim is to investigate the prediction performance of these AI models on the effects of roughness parameters on the wetting behavior of solid surfaces. We assume that the Wenzel model assumption holds: the contact angle of droplets posed on rough surfaces is determined by the roughness ratio  $r$  and especially the full active surface area. A hybridization framework is implemented, in which simulated rough surfaces with a wide spectrum of parameters and appearances are used to train and evaluate the AI models and explore their performance versus the simulation cost. We also use the capability of the developed AI models to reveal each roughness parameter importance on the observed wetting behavior.

## 3 METHODOLOGY

### 3.1 Mathematical modelling of rough surface generation.

In this section, we describe the methodology we used for generating simulated rough surfaces with similar characteristics with a large variety of experimental ones. These surfaces will be used to enrich the dataset for training and validating the ML models. We begin by describing the methodology for generating Gaussian and non-Gaussian surfaces with controlled spatial correlations.

*Gaussian surfaces:* We produce three-dimensional Gaussian surfaces by inputting the Rms of the height distribution and the correlation lengths along the x and y axes ( $\xi_x$  and  $\xi_y$ ) of the surface (Table 1). (Figure 1a and 1b). The heights of the generated surfaces are calculated on a square lattice  $N \times N$  points and area  $rL \times rL$ . Therefore, the spacing in x and y direction corresponds the ratio:  $\frac{rL}{N-1}$ . The methodology for simulating the Gaussian surfaces is based on the work of Garcia et.al. [13]. First, we produce a white

Surface Type	Input Parameters	Output Parameter
Gaussian	Rms, $\xi_x$ , $\xi_y$	active area
Non-Gaussian	Rms, $\xi_x$ , $\xi_y$ , skewness, kurtosis	active area

**Table 1: Input parameters used for generating the active area of Gaussian and non-Gaussian surfaces. The input (roughness) parameters for the Gaussian consist of the Rms heights and the correlation lengths ( $\xi$ ) in x and y directions. For the Non-Gaussian surfaces the inputs include additionally the skewness and kurtosis.**

noise  $Z_{WG}$  distribution with mean value zero and standard deviation equal to the input Rms value. By applying the Gaussian filter  $F_G$ , described in the following equation (Eq. 1) to the distribution we add the desired correlations along x and y axis.

$$F_G = \exp\left(-2\frac{x^2}{\xi_x^2} + 2\frac{y^2}{\xi_y^2}\right) \quad (1)$$

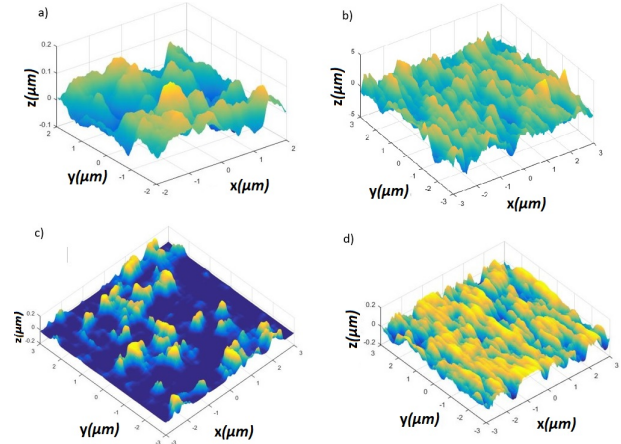
Then, we take the Inverse Fourier Transform of the product of Fourier transforms of  $Z_{WG}$  and  $F_G$  and multiply with normalization factors to generate correlated isotropic and anisotropic Gaussian surfaces. After producing the surface, their Rms,  $\xi_x$  and  $\xi_y$  are compared with the inputs to check possible divergence. The divergence is related with limitations imposed by the discrete sampling and finite range of surfaces. In such case, the algorithm of the surface generation is repeated until the input parameters are converged.

*Non-Gaussian surfaces:* The method we used to model non-Gaussian surfaces is based on the work of Yang et al. [14] where, the Johnson and Pearson transformations systems are used to transform random Gaussian noise with specified average height and Rms into non-Gaussian noise with user-defined skewness, and kurtosis (Table 1). The steps of the method are as follows: First, we generate a random two-dimensional non-Gaussian noise via Johnson transform system giving as input parameters the first four statistical moments (mean, Rms, skewness and kurtosis). If the distribution parameters cannot converge, we use the Pearson transformation system. Then, we measure the skewness and kurtosis of the surface to satisfy the chosen precision conditions. If the conditions are not met, we repeat the generation of the surface and validation. Finally, the surface becomes correlated by reconstructing and rearranging the height sequence in the x and y directions imitating a known Gaussian correlated surface with correlation lengths  $\xi_x$  and  $\xi_y$  along axes x and y respectively. Yang's method is characterized by its efficiency as different internal fitting methods are used for convergence. Thus, we can create surfaces with skewness and kurtosis inputs that can successfully (with low error) cover every point in the skewness-kurtosis plane  $Ku - Sk^2 - 1 \geq 0$ .

Subsequently, the active area was measured by integrating the secant of the angle  $\gamma$  between the surface normal and its z-direction normal.

$$\int \int dA = \int \int \sec(\gamma) dx dy \quad (2)$$

Where,  $\gamma$  is defined as the angle that the z-axis makes with the normal vector of the differential surface  $dA$ .



**Figure 1: Examples of simulated rough nano-surfaces**  
**a) Isotropic Gaussian surface** N= 600 points, rL=4  $\mu\text{m}$ ,  $\xi_x=0.5$   $\mu\text{m}$ ,  $\xi_y=0.5$   $\mu\text{m}$ , Rms=0.1  $\mu\text{m}$   
**b) Anisotropic Gaussian surface** N= 600 points, rL=6  $\mu\text{m}$ ,  $\xi_x=0.$   $\mu\text{m}$ ,  $\xi_y=0.6$   $\mu\text{m}$ , Rms=0.1  $\mu\text{m}$   
**c) Isotropic non-Gaussian surface** N= 600 points, rL=6  $\mu\text{m}$ ,  $\xi_x=0.5$   $\mu\text{m}$ ,  $\xi_y=0.5$   $\mu\text{m}$ , Rms=0.05  $\mu\text{m}$  Sk=2, Ku=6  
**d) Anisotropic non-Gaussian surface** N= 600 points, rL=6  $\mu\text{m}$ ,  $\xi_x=0.2$   $\mu\text{m}$ ,  $\xi_y=0.6$   $\mu\text{m}$ , Rms=0.05  $\mu\text{m}$  Sk=-0.4, Ku=2

### 3.2 AI techniques

In this section, we outline the AI methods we utilized in this work. We begin by over-viewing the main points describing the methods and then we align these descriptions with our work.

A physical model is a domain-driven model that is created through functions that follow underlying physical laws to predict a property. When using those models, the relation between the input parameters and the output value is already known and used for predictions. A ML model is a data-driven model that is used to find an appropriate (originally unknown) function that reflects the connection of an input (given) property to an output property that we want to predict. Even though the actual relation is unknown, given sufficient values for input and output values from experiments, we can create a ML (statistical) model that approximates the relation. Along with pre-existing human expertise, the approximations could add value to the manufacturer that aims to predict physical properties, without the need to exhaustively perform experiments, given the cost of such experiments in time and money.

In this work, we are using three ML models: 1) Linear Regression 2) Random Forests 3) Neural Networks. We try these different families of learning, since this problem has no prior indication of the underlying input-output relation, which may affect the method selection. For example, linear regression models will assume a linear relationship between the input and the output. On the other hand, Random forests and Neural network models use different methodologies to approximate/learn non-linear relationships between (even high dimensional) input parameters and a predicted output property. We stress that there is no single, overall better

method for all estimation problems. This statement is better known as the "no-free-lunch theorem" [15].

Linear regression models [16] (or in our case a multiple linear regression model) approximate the relationship between the input space and the output variable by fitting a linear equation. Those models are characterized by their simplicity and speed and are oftentimes used as a benchmark, to allow comparison to other, more complex models.

Random Forests [17] are ensemble methods that use multiple decision trees. A decision tree partitions the input space into sub-spaces and maps each subspace to a predicted output value. The division of the space is done using training (example) data from the dataset. The algorithm searches for an appropriate partitioning, which reduces the error of the predicted value across all training examples. Normally, each decision tree is fed by all the available training data. However, in the Random forest approach, the learning creates several trees, each applied on a randomly selected subset of the full training data. The model predicts the output by considering all of the predictions of each decision tree. This approach has been shown to be more effective and also generalize better with respect to unseen examples [18].

Neural Networks (NNs) are models that learn complex functions, combined in a non-linear manner over several layers. As such, NNs have been shown to be excellent approximators of many families of functions [19], meaning that they can mimic many underlying functions very efficiently, when given enough training data. NNs include neurons (or layers of nodes) that are combined to solve complex problems. A NN consists of an input layer, one or more hidden layers and an output layer. The input layer is fed a representation of the independent variables that (we expect) define the output. The output layer is expected to deliver the estimation of the output, dependent variable. Intermediate layers of nodes take as input the output of previous layers and transform it (i.e. apply a function on it). In essence, each node of a NN is a linear function of its input, passed through a non-linear operator. The intermediate layers end up forming a (sometimes very complex) function that connects input to output.

NN models with many layers are called Deep Neural Networks. Such a network may contain millions of parameters, identifying the function the network represents. No matter whether we have a deep network or not, these parameters are optimized to minimize the estimation error. In other words, we take each training example, provide it as input to the network and get its output prediction. Based on the real value it should have output, we change (i.e. optimize) the parameters of the network to minimize the prediction error. A number of optimization methods can be used to infer these parameters from the training data, the most well-known being back-propagation.

There is no a-priori best way to create a NN. However, standard practices can be applied to create such an architecture [20]. Generally, a Neural Network can be made deeper by adding more hidden layers or more nodes, that however may lead the function to over-fitting, i.e. reducing generalization ability to unseen input, which in turns increases the error when using the network for prediction. Thus, the definition of an architecture can be a challenge in itself. However, techniques such as dropout [21] have been proved to be effective.

Linear regression models are faster than both Neural networks and Random Forests but have less accuracy in cases where non-linear inner dependencies appear in data. Both, Neural Networks (NN) and Random Forests offer good levels of performance in different application areas. However, different methods offer different potential for learning and approximating, coupled with different processing requirements (in time and memory). Random forests training costs less time (when compared to NNs in a generic setting) and after training, they can be more interpretable than the average NN. On the other hand, the accuracy a neural network can reach is higher, if we have access to the required volume and diversity of data (more data is usually needed to tune more parameters). Essentially, the selection of training data is a defining factor for ML, beyond the algorithms and the corresponding architecture. These data should satisfy three basic requirements in order to make a good approximation through a ML model. Those requirements are related to the: 1) quantity 2) diversity 3) quality of data. Regarding our work, we aimed to satisfy these three requirements while creating the training and testing datasets before using the models.

## 4 RESULTS

### 4.1 Database Characteristics

By applying the methodology described in section 3.1, we generated two databases for the training and the validation of the ML models respectively. The databases consist of surfaces with diverse combinations of roughness parameters (Rms, correlation lengths, Skewness and Kurtosis) and the corresponding functional parameter (active area). The distribution of the training and validation datasets are shown in Figure 2 a) and b):

The total volume of training data-surfaces has been 3000 surfaces while for validating reached 15000. We trained each model with different percentages of surfaces from the train dataset to examine the effects of the training data size on model success. Also, for every percentage of the training set, we applied the training procedure 10 times with randomly selected surfaces from the database.

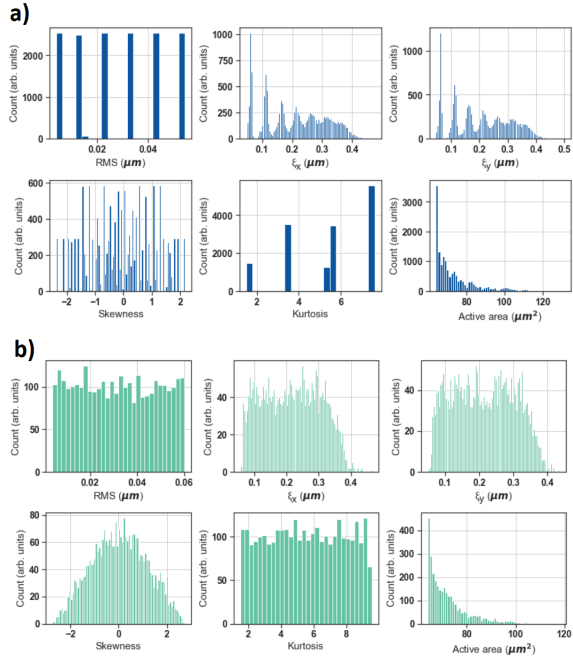
### 4.2 Model evaluation metric

For the validation of the predictability of models, we evaluated the RMSRE (Root Mean Square Relative Error). While the RMSE (Root Mean Square Error) can indicate successfully the appearance of outliers, the relative value of RMSE has no units. RMSE is the squared root error of the average predicted active area  $A_p$  from the average actual active area  $A_r$ .

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (A_p - A_r)^2}{N}} \quad (3)$$

To produce the RMSRE, the RMSE is normalized by the projected area of our surfaces. In our case, the projected area is  $A_{projected} = rL^2 = 64\mu m^2$ . Thus, for N surfaces the RMSRE would take the following formation:

$$RMSRE = \frac{RMSE}{A_{projected}} \cdot 100 \quad (4)$$



**Figure 2: a) Database of the surface roughness parameters used to train the models. b) Database of the roughness parameters used to validate the models. Each box represents a histogram for the specific parameters (Rms,  $\xi_x$ ,  $\xi_y$ , skewness, kurtosis) and the functional parameter of active area.**

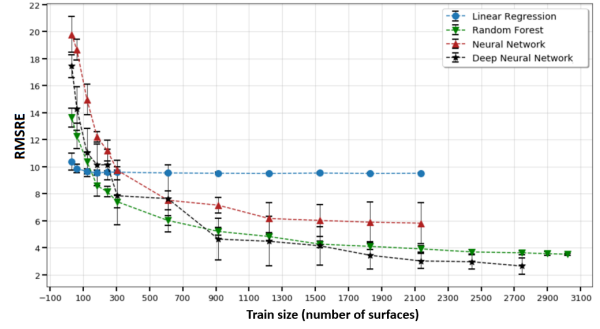
### 4.3 Machine Learning Results

#### *Linear Regression.*

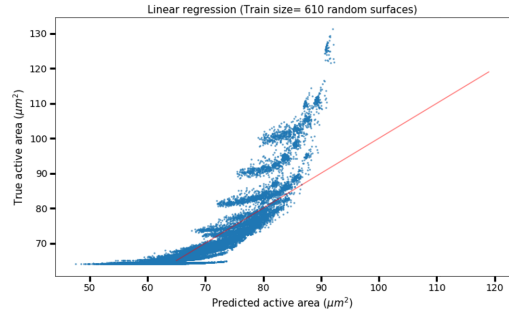
We trained a set of Linear Regression models as a basis comparison with the rest of models. Figure 3 shows the RMSRE of the trained models within a volume range of training data (surfaces). The linear models reached a plateau of RMSRE 9.6% using the validation dataset after approximately 100 training data points-surfaces. The model predictions diverge from the true values for high and low active area values, as seen in Figure 4.

#### *Random Forests.*

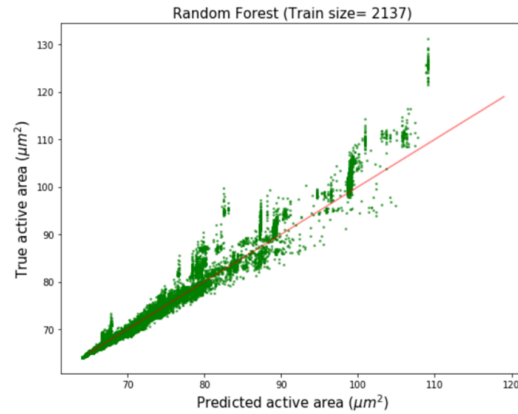
We used random forest estimators fitting 25 regression decision trees each. Each tree averages the results to improve the predictive accuracy and control over-fitting. The decision trees used for training had a maximum depth of 25 branches with a minimum number of internal node splits of two and a mean squared error criterion for splitting. Random Forest models outperforms the Linear Regression models after approximately 120 data-surfaces (Figure 3) and achieved less than 4.0% RMSRE after 2100 training data-surfaces. The standard deviation of the RMSRE decreases as the training dataset becomes larger. Figure 5 shows that the models were able to predict the high and low active areas of the surfaces with less error as compared to the the linear regression models. However, there is still a significant amount of error for high active areas.



**Figure 3: RMSRE vs the volume of the training data for Linear Regression, Random Forests, Neural Networks and Deep Neural Networks**



**Figure 4: True actual active versus predicted active area for the linear regression model. This model was trained with 610 random data-surfaces**



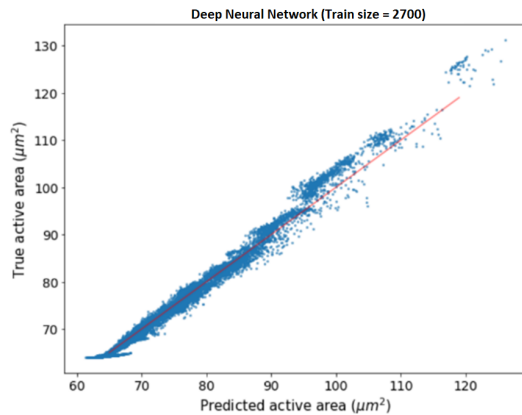
**Figure 5: True actual active versus predicted active area from a random forest model that was trained with 2137 random data-surfaces**

#### *Neural Networks and Deep neural networks.*

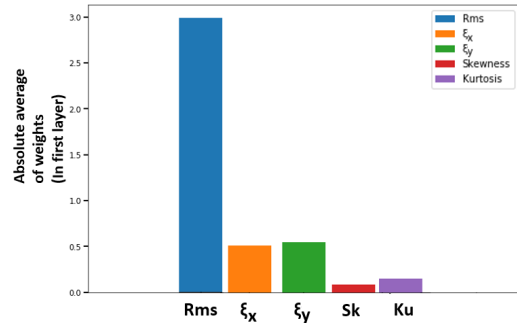
We used a set of neural network (NN) and a deep neural network (DNN) models for the prediction of active areas. NN models were

trained using the adam [22] solver with a five-layer architecture consisting of 15,25,40,25,15 nodes respectively. The DNN models were trained through rmsprop optimizer with a two-layer architecture consisting of 200 nodes each. To overcome any overfitting tendency, a 50% dropout was established between the two layers and an additive zero-centered Gaussian noise of  $std = 1$  was added to the last layer. Both models had activation function the Rectified Linear unit (ReLU).

For training datasets with a size of less than 300 surfaces, the NNs performed worse as compared to the linear regression models (Figure 3). After 300 surfaces, they outperformed the linear regression models reaching a  $RMSRE = 6\%$ . It is noted that in this case as well, the standard deviation increases with the size of the training data. The performance of DNNs is comparable to the Random forest model. For 2700 surfaces of training data, the DNN models show an average error of  $RMSRE = 2\%$  (Figure 6). Therefore, the DNNs generally show a significantly lower error compared to the NNs. By taking the average of the absolute value of the weights between the input layer and the first layer of the DNN, we can identify the most important features of the model. Figure 7 shows that Rms has the highest weight intensity, followed by the correlation lengths in x and y axis ( $\xi_x$  and  $\xi_y$  resp). This results is in harmony with previous findings based on computational analysis and experimental measurements [4] [3]



**Figure 6: True actual active versus predicted active area from a Deep Neural Network model that was trained with 2700 random data-surfaces**



**Figure 7: Absolute value of the weights between the input and the first layer of a Deep Neural Network that was trained with 2700 random data-surfaces**

## 5 SUMMARY

In our work, we proposed a hybridization scheme that combines data-driven methodologies (4th paradigm [8]) with domain theoretical and modeling knowledge as an alternative (2nd and 3rd paradigm) link between the configuration of nanosurface rough morphology and prediction of wetting behaviour.

In particular, we trained Linear Regression, Random Forest, Neural Network and Deep Neural Network models with simulated nanosurfaces in order to predict the true (active) area of the surface, a critical parameter for wetting when Wenzel model is assumed. We then compared their performances in relation to the required data (simulation cost to produce rough surfaces). Random Forests and Deep Neural Networks showed the highest performance reaching 4 % of RMSRE after 1000 training data-surfaces. The models and particularly the Deep Neural Networks indicate that Rms has the highest importance in wetting behavior. The correlation lengths in the x and y axis showed lower but significant importance as well whereas skewness and kurtosis play a minor though detectable role.

## REFERENCES

- [1] Y.-P. Zhao, *Characterization of Amorphous and Crystalline Rough Surface: Principles and Applications*. San Diego, CA: Elsevier, 2000.
- [2] R. N. Wenzel, "Resistance of solid surfaces to wetting by water," *Industrial & Engineering Chemistry*, vol. 28, no. 8, pp. 988–994, 1936.
- [3] L. Lai and E. A. Irene, "Area evaluation of microscopically rough surfaces," *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures Processing, Measurement, and Phenomena*, vol. 17, no. 1, pp. 33–39, 1999.
- [4] F. Foadi, G. H. ten Brink, M. R. Mohammadizadeh, and G. Palasantzas, "Roughness dependent wettability of sputtered copper thin films: The effect of the local surface slope," *Journal of Applied Physics*, vol. 125, no. 24, p. 244307, 2019.
- [5] M. B. Stearns, "Magnetization and structure of mn-ni and mn-co layered magnetic thin films," *Journal of Applied Physics*, vol. 53, no. 3, pp. 2436–2438, 1982.
- [6] Y. Zhang, *The effect of surface roughness parameters on contact and wettability of solid surfaces*. PhD thesis, Iowa State University, 2007.
- [7] G. M. Sacha and P. Varona, "Artificial intelligence in nanotechnology," *Nanotechnology*, vol. 24, p. 452002, oct 2013.
- [8] T. Zhou, Z. Song, and K. Sundmacher, "Big data creates new opportunities for materials research: A review on methods and applications of machine learning for materials design," *Engineering*, vol. 5, no. 6, pp. 1017 – 1026, 2019.
- [9] C. Sutton, "Crowd-sourcing materials-science challenges with the nomad 2018 kaggle competition," *npj Computational Materials*, vol. 5, 2019.
- [10] K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K. R. Müller, and E. K. U. Gross, "How to represent crystal structures for machine learning: Towards fast prediction of electronic properties," *Physical Review B*, vol. 89, May 2014.
- [11] A. Kordijazi, H. M. Roshan, A. Dhingra, M. Povoio, P. K. Rohatgi, and M. Nosonovsky, "Machine-learning methods to predict the wetting properties of

- iron-based composites," *Surface Innovations*, pp. 1–9, 2020.
- [12] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations," 2017.
- [13] N. Garcia and E. Stoll, "Monte carlo calculation for electromagnetic-wave scattering from random rough surfaces," *Phys. Rev. Lett.*, vol. 52, pp. 1798–1801, May 1984.
- [14] G. Yang, B. Li, Y. Wang, and J. Hong, "Numerical simulation of 3d rough surfaces and analysis of interfacial contact characteristics," *CMES - Computer Modeling in Engineering and Sciences*, vol. 103, pp. 251–279, 12 2014.
- [15] D. H. Wolpert, "The lack of a priori distinctions between learning algorithms," *Neural Computation*, vol. 8, no. 7, pp. 1341–1390, 1996.
- [16] D. W. Gareth James and H. Tibshirani, *An Introduction to Statistical Learning pp 59-126 and pp*. New York, NY: Springer, 2013.
- [17] Tin Kam Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, pp. 278–282 vol.1, 1995.
- [18] Tin Kam Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.
- [19] M. A. Nielsen, "Neural networks and deep learning," 2018.
- [20] P. Benardos and G.-C. Vosniakos, "Optimizing feedforward artificial neural network architecture," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 3, pp. 365 – 382, 2007.
- [21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, p. 1929–1958, Jan. 2014.
- [22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.