On Breast Cancer Detection: An Application of Machine Learning Algorithms on the Wisconsin Diagnostic Dataset

Abien Fred M. Agarap
abienfred.agarap@gmail.com

ABSTRACT

This paper presents a comparison of six machine learning (ML) algorithms: GRU-SVM[4], Linear Regression, Multilayer Perceptron (MLP), Nearest Neighbor (NN) search, Softmax Regression, and Support Vector Machine (SVM) on the Wisconsin Diagnostic Breast Cancer (WDBC) dataset[20] by measuring their classification test accuracy, and their sensitivity and specificity values. The said dataset consists of features which were computed from digitized images of FNA tests on a breast mass[20]. For the implementation of the ML algorithms, the dataset was partitioned in the following fashion: 70% for training phase, and 30% for the testing phase. The hyper-parameters used for all the classifiers were manually assigned. Results show that all the presented ML algorithms performed well (all exceeded 90% test accuracy) on the classification task. The MLP algorithm stands out among the implemented algorithms with a test accuracy of ≈99.04%.

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning by classification**; **Supervised learning by regression**; **Support vector machines**; **Neural networks**;

KEYWORDS

artificial intelligence; artificial neural networks; classification; linear regression; machine learning; multilayer perceptron; nearest neighbors; softmax regression; supervised learning; support vector machine; wisconsin diagnostic breast cancer dataset

ACM Reference Format:
Abien Fred M. Agarap. 2018. On Breast Cancer Detection: An Application of Machine Learning Algorithms on the Wisconsin Diagnostic Dataset. In *ICMLSC 2018: ICMLSC 2018, The 2nd International Conference on Machine Learning and Soft Computing, February 2–4, 2018, Phu Quoc Island, Viet Nam.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3184066.3184080

1 INTRODUCTION

Breast cancer is one of the most common cancer along with lung and bronchus cancer, prostate cancer, colon cancer, and pancreatic cancer among others[2]. Representing 15% of all new cancer cases in the United States alone[1], it is a topic of research with great value.

The utilization of data science and machine learning approaches in medical fields proves to be prolific as such approaches may be considered of great assistance in the decision making process of medical practitioners. With an unfortunate increasing trend of breast cancer cases[1], comes also a big deal of data which is of significant use in furthering clinical and medical research, and much more to the application of data science and machine learning in the aforementioned domain.

Prior studies have seen the importance of the same research topic[17, 21], where they proposed the use of machine learning (ML) algorithms for the classification of breast cancer using the Wisconsin Diagnostic Breast Cancer (WDBC) dataset[20], and eventually had significant results.

This paper presents yet another study on the said topic, but with the introduction of our recently-proposed GRU-SVM model[4]. The said ML algorithm combines a type of recurrent neural network (RNN), the gated recurrent unit (GRU)[8] with the support vector machine (SVM)[9]. Along with the GRU-SVM model, a number of ML algorithms is presented in Section 2.4, which were all applied on breast cancer classification with the aid of WDBC[20].

2 METHODOLOGY

2.1 Machine Intelligence Library

Google TensorFlow[3] was used to implement the machine learning algorithms in this study, with the aid of other scientific computing libraries: matplotlib[12], numpy[19], and scikit-learn[15].

2.2 The Dataset

The machine learning algorithms were trained to detect breast cancer using the Wisconsin Diagnostic Breast Cancer (WDBC) dataset[20]. According to [20], the dataset consists of features which were computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. The said features describe the characteristics of the cell nuclei found in the image[20].

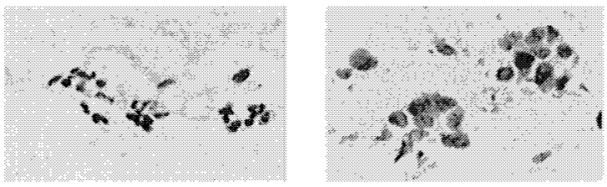

Figure 1: Image from [20] as cited by [21]. Digitized images of FNA: (a) Benign, (b) Malignant.

There are 569 data points in the dataset: 212 – Malignant, 357 – Benign. Accordingly, the dataset features are as follows: (1) radius,

ICMLSC 2018, February 2–4, 2018, Phu Quoc Island, Viet Nam

ACM ISBN 978-1-4503-6336-5/18/02…$15.00
https://doi.org/10.1145/3184066.3184080

(2) texture, (3) perimeter, (4) area, (5) smoothness, (6) compactness, (7) concavity, (8) concave points, (9) symmetry, and (10) fractal dimension. With each feature having three information[20]: (1) mean, (2) standard error, and (3) "worst" or largest (mean of the three largest values) computed. Thus, having a total of 30 dataset features.

2.3 Dataset Preprocessing

To avoid inappropriate assignment of relevance, the dataset was standardized using Eq. 1.

$$z = \frac{X - \mu}{\sigma} \tag{1}$$

where X is the feature to be standardized, μ is the mean value of the feature, and σ is the standard deviation of the feature. The standardization was implemented using `StandardScaler().fit_transform()` of `scikit-learn`[15].

2.4 Machine Learning (ML) Algorithms

This section presents the machine learning (ML) algorithms used in the study. The Stochastic Gradient Descent (SGD) learning algorithm was used for all the ML algorithms presented in this section except for GRU-SVM, Nearest Neighbor search, and Support Vector Machine. The code implementations may be found online at https://github.com/AFAgarap/wisconsin-breast-cancer.

2.4.1 GRU-SVM. We proposed a neural network architecture[4] combining the gated recurrent unit (GRU) variant of recurrent neural network (RNN) and the support vector machine (SVM), for the purpose of binary classification.

$$z = \sigma(\mathbf{W}_z \cdot [h_{t-1}, x_t]) \tag{2}$$

$$r = \sigma(\mathbf{W}_r \cdot [h_{t-1}, x_t]) \tag{3}$$

$$\tilde{h}_t = tanh(\mathbf{W} \cdot [r_t * h_{t-1}, x_t]) \tag{4}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{5}$$

where z and r are the *update gate* and *reset gate* of a GRU-RNN respectively, $\tilde{h}_t$ is the candidate value, and h_t is the new RNN cell state value[8]. In turn, the h_t is used as the predictor variable $\mathbf{x}$ in the L2-SVM predictor function (given by $sign(\mathbf{wx} + b)$) of the network instead of the conventional Softmax classifier.

The learning parameter $\mathbf{W}$ of the GRU-RNN is learned by the L2-SVM using the loss function given by Eq. 20. The computed loss is then minimized through Adam[13] optimization. The same optimization algorithm was used for Softmax Regression (Section 2.4.5) and SVM (Section 2.4.6). Then, the decision function $f(x) = sign(\mathbf{wx} + b)$ produces a vector of scores for each cancer diagnosis: -1 for benign, and +1 for malignant. In order to get the predicted labels y for a given data $\mathbf{x}$, the *argmax* function is used (see Eq. 6).

$$\mathbf{y}' = argmax\big(sign(\mathbf{wx} + b)\big) \tag{6}$$

The *argmax* function shall return the indices of the highest scores across the vector of predicted classes $sign(\mathbf{wx} + b)$.

2.4.2 Linear Regression. Despite an algorithm for regression problem, linear regression (see Eq. 7) was used as a classifier for this study. This was done by applying a threshold for the output of Eq. 7, i.e. subjecting the value of the regressand to Eq. 8.

$$h_\theta(x) = \sum_{i=0}^{n} \theta_i \cdot x_i + b \tag{7}$$

$$f\big(h_\theta(x)\big) = \begin{cases} 1 & h_\theta(x) \geq 0.5 \\ 0 & h_\theta(x) < 0.5 \end{cases} \tag{8}$$

To measure the loss of the model, the mean squared error (MSE) was used (see Eq. 9).

$$L(y, \theta, x) = \frac{1}{N} \sum_{i=0}^{N} \big(y_i - (\theta_i \cdot x_i + b)\big)^2 \tag{9}$$

where y represents the actual class, and $(\theta \cdot \mathbf{x} + b)$ represents the predicted class. This loss is minimized using the SGD algorithm, which learns the parameters θ of Eq. 7. The same method of loss minimization was used for MLP and Softmax Regression.

2.4.3 Multilayer Perceptron. The perceptron model was developed by Rosenblatt (1958)[16] based on the neuron model by McCulloch & Pitts (1943)[14]. The multilayer perceptron (MLP)[7] consists of hidden layers (composed by a number of perceptrons) that enable the approximation of any functions, that is, through activation functions such as *tanh* or *sigmoid* σ.

$$h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i + b \tag{10}$$

$$f\big(h_\theta(x)\big) = \mathbf{h}_\theta(\mathbf{x})^+ = max(0, \mathbf{h}_\theta(\mathbf{x})) \tag{11}$$

For this study, the activation function used for MLP was ReLU[11] (see Eq. 11), while there were three hidden layers that each consists of 500 nodes (500-500-500 architecture). As for the loss, it was computed using the cross entropy function (see Eq. 15).

2.4.4 Nearest Neighbor. This is a form of an optimization problem that seeks to find the closest point $p_i \in \mathbf{p}$ to a query point $q_i \in \mathbf{q}$. In this study, both the L1 (Manhattan, see Eq. 12) and L2 (Euclidean, see Eq. 13) norm were used to measure the distance between $\mathbf{p}$ and $\mathbf{q}$.

$$\|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i| \tag{12}$$

$$\|\mathbf{p} - \mathbf{q}\|_2 = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{13}$$

The code implementation was based on the work of Damien (2017)[10] in GitHub. A learning algorithm such as SGD and Adam[13] is not applicable to Nearest Neighbor search, as it is practically a geometric approach for classification.

2.4.5 Softmax Regression. This is a classification model generalizing logistic regression to multinomial problems. But unlike linear regression (Section 2.4.2) that produces raw scores for the classes, softmax regression produces a probability distribution for the classes. This is accomplished using the Softmax function (see Eq. 14).

$$P(\hat{\mathbf{y}} \mid \mathbf{x}) = \frac{e^{\hat{y}_i}}{\sum_{i=0}^{n} e^{\hat{y}_i}} \tag{14}$$

$$L(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=0}^{n} y_i \cdot log(\hat{y}_i) \tag{15}$$

The loss is measured by using the cross entropy function (see Eq. 15), where $\mathbf{y}$ represents the actual class, and $\hat{\mathbf{y}}$ represents the predicted class.

2.4.6 Support Vector Machine. Developed by Vapnik[9], the support vector machine (SVM) was primarily intended for binary classification. Its main objective is to determine the optimal hyperplane $f(w, x) = \mathbf{w} \cdot \mathbf{x} + b$ separating two classes in a given dataset having input features $\mathbf{x} \in \mathbb{R}^p$, and labels $y \in \{-1, +1\}$.

SVM learns by solving the following constrained optimization problem:

$$min \frac{1}{p} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^{p} \xi_i \tag{16}$$

$$s.t\ y_i'(\mathbf{w} \cdot \mathbf{x} + b) \geq 1 - \xi_i \tag{17}$$

$$\xi_i \geq 0, i = 1, ..., p \tag{18}$$

where $\mathbf{w}^T \mathbf{w}$ is the Manhattan norm, ξ is a cost function, and C is the penalty parameter (may be an arbitrary value or a selected value using hyper-parameter tuning). Its corresponding unconstrained optimization problem is the following:

$$min \frac{1}{p} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^{p} max\big(0, 1 - y_i'(w_i x_i + b)\big) \tag{19}$$

where $\mathbf{w}\mathbf{x} + b$ is the predictor function. The objective of Eq. 19 is known as the primal form problem of L1-SVM, with the standard hinge loss. The problem with L1-SVM is the fact that it is not differentiable[18], as opposed to its variation, the L2-SVM:

$$min \frac{1}{p} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^{p} max\big(0, 1 - y_i'(w_i x_i + b)\big)^2 \tag{20}$$

The L2-SVM is differentiable and provides more stable results than its L1 counterpart[18].

2.5 Data Analysis

There were two phases of experiment for this study: (1) training phase, and (2) test phase. The dataset was partitioned by 70% (training phase) / 30% (testing phase). The parameters considered in the experiments were as follows: (1) Test Accuracy, (2) Epochs, (3) Number of data points, (4) False Positive Rate (FPR), (5) False Negative Rate (FNR), (6) True Positive Rate (TPR), and (7) True Negative Rate (TNR).

3 RESULTS AND DISCUSSION

All experiments in this study were conducted on a laptop computer with Intel Core(TM) i5-6300HQ CPU @ 2.30GHz x 4, 16GB of DDR3 RAM, and NVIDIA GeForce GTX 960M 4GB DDR5 GPU. Table 1 shows the manually-assigned hyper-parameters used for the ML algorithms. Table 2 summarizes the experiment results. In addition to the reported results, the result from [21] was put into comparison.

[21] implemented the SVM with Gaussian Radial Basis Function (RBF) as its kernel for classification on WDBC. Their experiment revealed that their SVM had its highest test accuracy of 89.28% with its free parameter $\sigma = 0.6$. However, their experiment was based on a 60/40 partition (training/testing respectively). Hence, we would not be able to draw a fair comparison between the current study and [21]. Comparing the results of this study on an intuitive sense may perhaps be close to a fair comparison, recalling that the partition done in this study was 70/30.

With a test accuracy of ≈96.09%, the L2-SVM in this study bares superiority against the findings of [21] (SVM with Gaussian RBF, having a test accuracy of 89.28%). But then again, it was based on a higher training data of 10% (70% vs 60%).

Figure 2 shows the training accuracy of the ML algorithms: (1)

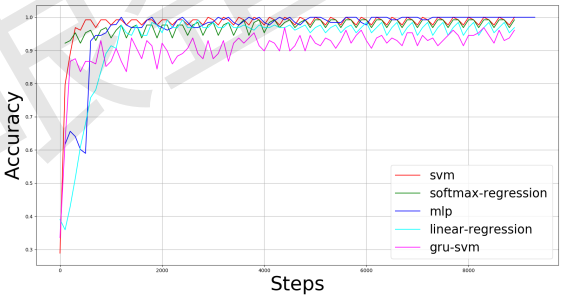

Figure 2: Plotted using `matplotlib`[12]. Training accuracy of the ML algorithms on breast cancer detection using WDBC.

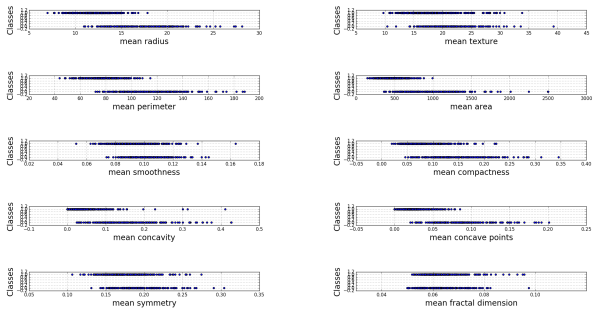

Figure 3: Plotted using `matplotlib`[12]. Scatter plot of *mean* features ($x_0 - x_9$) in the WDBC.

GRU-SVM finished its training in 2 minutes and 54 seconds with

Table 1: Hyper-parameters used for the ML algorithms.

Hyper-parameters	GRU-SVM	Linear Regression	MLP	Nearest Neighbor	Softmax Regression	SVM
Batch Size	128	128	128	N/A	128	128
Cell Size	128	N/A	[500, 500, 500]	N/A	N/A	N/A
Dropout Rate	0.5	N/A	None	N/A	N/A	N/A
Epochs	3000	3000	3000	1	3000	3000
Learning Rate	1e-3	1e-3	1e-2	N/A	1e-3	1e-3
Norm	L2	N/A	N/A	L1, L2	N/A	L2
SVM C	5	N/A	N/A	N/A	N/A	5

Table 2: Summary of experiment results on the ML algorithms.

Parameter	GRU-SVM	Linear Regression	MLP	L1-NN	L2-NN	Softmax Regression	SVM
Accuracy	93.75%	96.09375%	99.038449585420729%	93.567252%	94.736844%	97.65625%	96.09375%
Data points	384000	384000	512896	171	171	384000	384000
Epochs	3000	3000	3000	1	1	3000	3000
FPR	16.666667%	10.204082%	1.267042%	6.25%	9.375%	5.769231%	6.382979%
FNR	0	0	0.786157%	6.542056%	2.803738%	0	2.469136%
TPR	100%	100%	99.213843%	93.457944%	97.196262%	100%	97.530864%
TNR	83.333333%	89.795918%	98.732958%	93.75%	90.625%	94.230769%	93.617021%

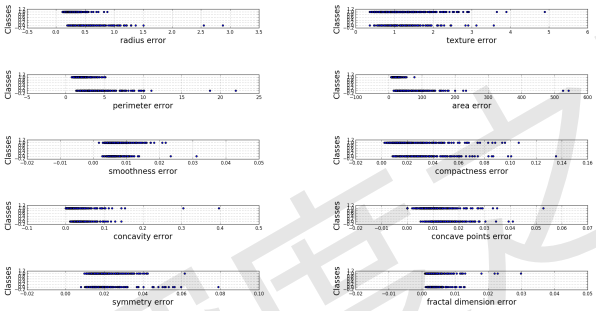

Figure 4: Plotted using `matplotlib`[12]. Scatter plot of *error* features ($x_{10} - x_{19}$) in the WDBC.

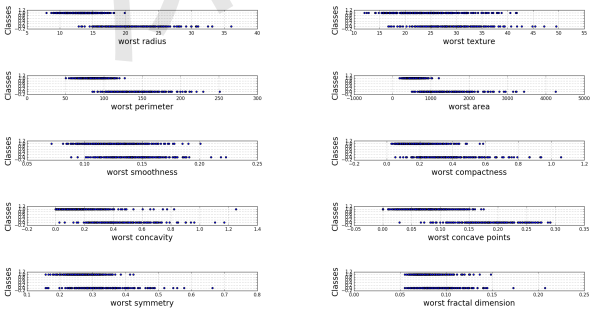

Figure 5: Plotted using `matplotlib`[12]. Scatter plot of *worst* features ($x_{20} - x_{29}$) in the WDBC.

an average training accuracy of 90.6857639%, (2) Linear Regression finished its training in 35 seconds with an average training accuracy of 92.8906257%, (3) MLP finished its training in 28 seconds with an average training accuracy of 96.9286785%, (4) Softmax Regression finished its training in 25 seconds with an average training accuracy of 97.366573%, and (5) L2-SVM finished its training in 14 seconds with an average training accuracy of 97.734375%. There was no recorded training accuracy for Nearest Neighbor search since it does not require any training, as the norm equations (Eq. 12 and Eq. 13) are directly applied on the dataset to determine the "nearest neighbor" of a given data point $p_i \in \mathbf{p}$.

The empirical evidence presented in this section draws a qualitative comparability with, and corroborates the findings of [21]. Hence, a testament to the effectiveness of ML algorithms on the diagnosis of breast cancer. While the experiment results are all commendable, the performance of the GRU-SVM model[4] warrants a discussion. The mid-level performance of GRU-SVM with a test accuracy of 93.75% is hypothetically attributed to the following information: (1) the non-linearities introduced by the GRU model[8] through its gating mechanism (see Eq. 2, Eq. 3, and Eq. 4) to its output may be the cause of a difficulty in generalizing on a linearly-separable data such as the WDBC dataset, and (2) the sensitivity of RNNs to weight initialization[5]. Since the weights of the GRU-SVM model are assigned with arbitrary values, it will also prove limited capability of result reproducibility, even when using an identical configuration[5].

Despite the given arguments, it does not necessarily revoke the fact that GRU-SVM is comparable with the presented ML algorithms, as what the results have shown. In addition, it was a expected that the upper hand goes to the linear classifiers (Linear Regression and SVM) as the utilized dataset was linearly separable. The linear separability of the WDBC dataset is shown in a

naive method of visualization (see Figure 3, Figure 4, and Figure 5). Visually speaking, it is palpable that the scattered features in the mentioned figures may be easily separated by a linear function.

4 CONCLUSION AND RECOMMENDATION

This paper presents an application of different machine learning algorithms, including the proposed GRU-SVM model in [4], for the diagnosis of breast cancer. All presented ML algorithms exhibited high performance on the binary classification of breast cancer, i.e. determining whether benign tumor or malignant tumor. Consequently, the statistical measures on the classification problem were also satisfactory.

To further substantiate the results of this study, a CV technique such as k-fold cross validation should be employed. The application of such a technique will not only provide a more accurate measure of model prediction performance, but it will also assist in determining the most optimal hyper-parameters for the ML algorithms[6].

5 ACKNOWLEDGMENT

Deep appreciation is given to the family and friends of the author (in arbitrary order): Myra M. Maranan, Faisal E. Montilla, Corazon Fabreag-Agarap, Crystal Love Fabreag-Agarap, Michaelangelo Milo L. Lim, Liberato F. Ramos, Hyacinth Gasmin, Rhea Jude Ferrer, Ma. Pauline de Ocampo, and Abqary Alon.

REFERENCES

[1] [n. d.]. ([n. d.]). https://seer.cancer.gov/statfacts/html/breast.html
[2] 2017. Cancer Statistics. (Mar 2017). https://www.cancer.gov/about-cancer/understanding/statistics
[3] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (2015). http://tensorflow.org/ Software available from tensorflow.org.
[4] Abien Fred Agarap. 2017. A Neural Network Architecture Combining Gated Recurrent Unit (GRU) and Support Vector Machine (SVM) for Intrusion Detection in Network Traffic Data. *arXiv preprint arXiv:1709.03082* (2017).
[5] Abdulrahman Alalshekmubarak and Leslie S Smith. 2013. A novel approach combining recurrent neural network and support vector machines for time series classification. In *Innovations in Information Technology (IIT), 2013 9th International Conference on*. IEEE, 42–47.
[6] Yoshua Bengio, Ian J Goodfellow, and Aaron Courville. 2015. Deep learning. *Nature* 521 (2015), 436–444.
[7] Christopher M Bishop. 1995. *Neural networks for pattern recognition*. Oxford university press.
[8] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).
[9] C. Cortes and V. Vapnik. 1995. Support-vector Networks. *Machine Learning* 20.3 (1995), 273–297. https://doi.org/10.1007/BF00994018
[10] Aymeric Damien. 2017, August 29. (2017, August 29). https://github.com/aymericdamien/TensorFlow-Examples/blob/master/examples/2_BasicModels/nearest_neighbor.py Accessed: November 17, 2017.
[11] Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. 2000. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature* 405, 6789 (2000), 947–951.
[12] J. D. Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering* 9, 3 (2007), 90–95. https://doi.org/10.1109/MCSE.2007.55
[13] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[14] Warren S McCulloch and Walter Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* 5, 4 (1943), 115–133.
[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[16] Frank Rosenblatt. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review* 65, 6 (1958), 386.
[17] Gouda I Salama, M Abdelhalim, and Magdy Abd-elghany Zeid. 2012. Breast cancer diagnosis on three different datasets using multi-classifiers. *Breast Cancer (WDBC)* 32, 569 (2012), 2.
[18] Yichuan Tang. 2013. Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239* (2013).
[19] Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. 2011. The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering* 13, 2 (2011), 22–30.
[20] William H Wolberg, W Nick Street, and Olvi L Mangasarian. 1992. Breast cancer Wisconsin (diagnostic) data set. *UCI Machine Learning Repository [http://archive.ics.uci.edu/ml/]* (1992).
[21] Elias Zafiropoulos, Ilias Maglogiannis, and Ioannis Anagnostopoulos. 2006. A support vector machine approach to breast cancer diagnosis and prognosis. *Artificial Intelligence Applications and Innovations* (2006), 500–507.