Performance Analysis of Machine Learning Algorithms for Cervical Cancer Detection

Sanjay Kumar Singh, I.K. Gujral Punjab Technical University, Kapurthala, Punjab, India

Anjali Goyal, Department of Computer Applications, GNIMT, Ludhiana, Punjab, India

ABSTRACT

Cervical cancer is second most prevailing cancer in women all over the world and the Pap smear is one of the most popular techniques used to diagnosis cervical cancer at an early stage. Developing countries like India has to face the challenges in order to handle more cases day by day. In this article, various online and offline machine learning algorithms has been applied on benchmarked data sets to detect cervical cancer. This article also addresses the problem of segmentation with hybrid techniques and optimizes the number of features using extra tree classifiers. Accuracy, precision score, recall score, and F1 score are increasing in the proportion of data for training and attained up to 100% by some algorithms. Algorithm like logistic regression with L1 regularization has an accuracy of 100%, but it is too much costly in terms of CPU time in comparison to some of the algorithms which obtain 99% accuracy with less CPU time. The key finding in this article is the selection of the best machine learning algorithm with the highest accuracy. Cost effectiveness in terms of CPU time is also analysed.

KEYWORDS

Cervical Cancer Detection, Decision Tree, Gaussian Naïve Bayes, Gradient Boosting, K-Nearest Neighbour, Logistic Regression, Machine Learning, Pap-Smear Classification, Perceptron, Random Forest

1. INTRODUCTION

Cervical cancer is the one of most common cancer found in women for the last 30 years or more, and almost every woman is at risk (Office on women's health, U.S. Department of Health and Human Services, 2018). The best way to prevent cervical cancer is to get an HPV vaccine, get regular pap tests, and be monogamous. In cervical cancer, malignant (cancer) cells form in the tissues of the cervix and Human papillomavirus (HPV) contamination is the real hazard factor for cervical disease. Pap test is one of the methods used to detect and diagnose cervical cancer. In this test, piece of cotton and a brush is used to collect cells from the surface of cervix and vagina. The cells are examined under a microscope to check for abnormality, this procedure is sometimes called as pap smear (National institutes of health, 2018). It is a challenging task to detect cervical cancer because there are no symptoms found in women at early stages of cervical cancer (American cancer society, 2018). A well-demonstrated approach to prevent cervical cancer is to have screening to discover pre-tumours before they can transform into obtrusive growth.

DOI: 10.4018/IJHISI.2020040101

This article, originally published under IGI Global's copyright on December 20, 2019 will proceed with publication as an Open Access article starting on January 14, 2021 in the gold Open Access journal, International Journal of Healthcare Information Systems and Informatics (converted to gold Open Access January 1, 2021), and will be distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

According to (international agency for research on cancer, 2018), number of new cancer cases predicted in 2020 for all age groups are shown in Figure 1. It is found that there is approx. 15.48% increase in patients of all age in world.

Hence it is recommended to have a reliable computer-based system for detection of cervical cancer in the world, as number of patients is increasing day by day. To implement such system, machine learning algorithms can be most suitable, and they can easily detect cervical cancer at its early stage.

Machine learning algorithms can be used at two different stages for detecting cervical cancer. First one is when it is used after extracting meaningful features. In this case, machine learning algorithms can be used to learn these features that will further predict and detect the cancer. Second one is when machine learning algorithms are used to extract important features, learning of features followed by prediction and detection of cervical cancer. In this paper, online and offline machine learning algorithms are also used which are experimented upon benchmarked data sets. Online machine learning algorithms have the ability to learn data sets one by one, also called incremental learning whereas offline machine learning algorithms also called batch learning learn through batches or groups of data at a time. Online machine learning algorithms are relatively fast as compared to offline machine learning algorithms.

Cervical cell classification can be used to classify the problem in major 2 class, 3 class, 4 class and 7 class problems. 2-class problem differentiates normal cells from abnormal cells. The 3-class problem differentiates cells among normal, low grade squamous intra-epithelial lesion (LSIL) and high grade squamous intra-epithelial lesion (HSIL). The 4-class problem differentiates between normal cells and 3 categories of abnormal cells mainly mild, moderate and severe level of dysplastic abnormal cells. The 7-class problem relates to differentiation of each class of cells. Minimum distance classifiers and ant colony optimization for classification of these cells in seven classes (Jantzen & Dounias, 2006). A metaheuristic algorithm with genetic algorithm for feature selection has been used for classification task for Pap smear images into two or more classes (Marinakakis, Dounias & Jantzen, 2009). Various complete intelligent techniques such as clustering algorithms, inductive machine learning, genetic programming, ANFIS and second order neural networks for cervical cells classification in two or more classes (Dounias et al., 2006; Sing & Pandey, 2014). It has been observed that the classification rate of 2-class problem is better than 7-class problem. These algorithms have been used along with feature selection on extracted features.

An efficient pap smear classifier has been proposed, which is based on 2nd order neural network learning algorithms (Ampazis, Dounias & Jantzen, 2004). For diagnostic of cervical

Figure 1. World cervix uteri number of new cancers in 2020, includes all ages

cancer, hierarchical neural network has been used which improves the classification accuracy with a fewer number of features (Mustafa, Mat-Isa, Manshor & Othman, 2008). Fuzzy based techniques are also used for segmentation of Pap smear images into nucleus, cytoplasm and background (Chankong, Theera-Umpun & Auephanwiriyakul, 2014). In his paper, 2-class classification is considered for experimentation.

Artificial neural networks, support vector machine based supervised learning and nearest neighbour based unsupervised learning are used for classification of cervical cells among 2-class or more class classification. Recently convolutional neural networks are used (Zhang, Summers, Kebebew & Yao, 2018; Takkar, Singh & Pandey, 2017) to extract features and also for detecting cervical cancers. In tumour growth prediction, deep convolution networks can be used for cell invasion and mass-effect. Deep convolutional networks to extract features automatically by deep learning and do the classification of cervical cells efficiently (Zhang et al., 2017; Bora, Chowdhury, Mahanta, Kundu & Das, 2016). This is one of segmentation free methods for classification tasks.

In this paper, the proposed methodology selects the best features using extra tree classifier which reduces the training time and helps oncologists to detect the cervical cancer with fast response time. Enormous experiments have been performed to find out the best machine learning algorithm in terms of accuracy and responsiveness for cervical cancer detection. The rest of paper is divided as: Section 2 explains proposed methodology to detect cervical cancer using various machine learning algorithms along with hybrid segmentation and feature selection methods. Section 3 discusses experiments and results followed by Section 4 comprising of conclusion and future works.

2. PROPOSED METHODOLOGY

To detect the cervical cancer, the proposed methodology is broadly divided into following subtasks as shown in Figure 2 data base collection, pre-processing, segmentation using hybrid segmentation techniques, extraction of shape and texture-based features, feature selection using extra tree classifier and finally classify using various machine learning algorithms.

Figure 2. Proposed methodology

2.1. Pap Smear Database

The authors used the DTU/Herlev pap smear database freely available on website for research work (Dounias, 2017). The Pap smear data base contains 917 images including 242 normal and 675 abnormal smears. Figure 3 displays some samples of images for both normal and abnormal smears.

2.2. Pre-Processing of Images

Pre-processing is one of the tasks used in image processing that reduces the noise available in an image for better use. In this procedure, all images are passed through median filter of size 3×3 for smoothening of images.

2.3. Segmentation of Pap Smear Images

This step is used to extract background, cytoplasm and nucleus part of single cell Pap smear images. For this purpose, we propose a new hybrid segmentation technique as shown in Figure 4.

The authors used watershed segmentation for nucleus detection and multilevel image thresholding for cytoplasm detection. After that both images are overlapped for full image segmentation as shown in Figure 5.

2.4. Feature Extraction

After segmentation, few images have been discarded as they are not properly segmented. As a result of this, 803 images are properly segmented (230 normal image and 573 abnormal images) out of 917 images which are considered for feature extraction. Total of 25 features are extracted out of which 20

Figure 3. Sample image data sets of normal and abnormal smears

Figure 4. Proposed hybrid segmentation approach

Figure 5. Segmented image

are shape based and rest 5 are texture-based features. All 25 features are calculated using MATLAB, at personal computer of 64-bit CPU.

(Byriel, 1999; Jantzen & Dounias, 2006; Chen et al., 2014) uses shape-based features for nucleus and cytoplasm in his research work are:

1. **Nucleus area, cytoplasm area:** It will be computed as number of pixels in nucleus and cytoplasm respectively;
2. **N/c ratio:** It is ratio of nucleus area to nucleus and cytoplasm area;
3. **Nucleus brightness and cytoplasm brightness:** It is the average intensity of nucleus and cytoplasm respectively;
4. **Nucleus perimeter and cytoplasm perimeter:** It is the boundary cover nucleus and cytoplasm part of cell;
5. **Nucleus maxima and nucleus minima:** Number of pixels found minimum then their surroundings in window size 3×3 by sliding over grayscale image of nucleus and cytoplasm respectively;
6. **Cytoplasm maxima and cytoplasm minima:** Number of pixels found maximum then their surroundings in window size 3×3 by sliding over grayscale image of nucleus and cytoplasm respectively;
7. **Nucleus relative positions:** Relative place of nucleus inside cytoplasm;
8. **Nucleus shortest diameter, nucleus longest diameter:** It is the shortest diameter perpendicular to longest diameter of nucleus and cytoplasm;
9. **Cytoplasm shortest diameter, cytoplasm longest diameter:** It is the distance between farthest points on perimeter of nucleus and cytoplasm;
10. **Nucleus elongation, cytoplasm elongation:** It is the ratio of shortest diameter to longest diameter of nucleus and cytoplasm respectively;
11. **Nucleus roundness, cytoplasm roundness:** It is the ratio of actual area to area of minimum surrounding circles for nucleus and cytoplasm respectively.

Let $P(i,j)$ is the probability of a pixel with grey value i having a pixel with grey-level j adjacent to it. Texture based features explained in (Chen et al., 2014) are calculated as:

- **Entropy:** Statistical measure of randomness in an image:

$$entropy = -\sum_i \sum_j P(i, j) \log P(i, j) \quad (1)$$

- **Contrast:** Measure of intensity contrast between a pixel and its neighbour over complete image:

$$contrast = \sum_i \sum_j \|i - j\|^2 P(i, j) \quad (2)$$

- **Correlation:** Measure of how correlated a pixel and its neighbour over complete image:

$$correlation = \sum_i \sum_j \frac{(i - \mu_i)(j - \mu_j) P(i, j)}{\sigma_i \sigma_j} \quad (3)$$

- **Energy:** Sum of square element in grey level co-occurrence matrix:

$$energy = \sum_i \sum_j P(i, j)^2 \quad (4)$$

- **Homogeneity:** Measure the closeness of distribution of elements in grey level co-occurrence matrix to its diagonal elements:

$$homogeneity = -\sum_i \sum_j \frac{P(i, j)}{1 + |i - j|} \quad (5)$$

2.5. Standardization

Standardization is one of the processes used to bringing all features at the same scale. Standardization for a particular sample $X^{(i)}$ with mean μ_x and standard deviation σ_x for corresponding sample calculated (Raschka, 2015) as:

$$X_{stand}^{(i)} = \frac{X^{(i)} - \mu_x}{\sigma_x} \quad (6)$$

2.6. Feature Selection

After the extraction of all 25 features, we have used extra tree classifiers to measure the importance of each feature that will help to build least weight classifiers. To compute feature importance using extra tree classifier open source Python-based package Scikitlearn is used (Pedregosa et al., 2011). It differs from simple decision tree in way to choose best split to separate samples in two groups at a node, using randomly selected features. Extremely randomized trees as proposed by (Geurts, Ernst, & Wehenkel, 2006), is an ensemble method for classification and regression in supervised learning. The purpose of randomization is used for tuning the best parameter and in addition to this it is computationally inexpensive.

The relative rank of features used as decision node in a tree to assess relative importance of that feature to predict the target output. Feature used at top nodes of tree contributes to final decision of a large fraction of the input examples. The relative importance of features can be used as expected fractions of samples they contribute to final prediction decision. The importance of all features is shown in Table 1 and represented graphically in Figure 6.

The result of extra-tree classifier clearly shows that contrast-based features are less important as compared to shape based features. As a result of this experiment, we have used only top 15 features out of 25 features for machine learning algorithms.

2.7. Machine Learning Algorithms

After the feature selection, the next task is to train and test machine learning algorithms on 15 selected features out of 25, suggested by feature selection method. For this purpose, we have used open source Python package Scikitlearn (Pedregosa et al., 2011).

Table 1. Feature ranking

Rank	Feature No.	Feature Name	Feature Importance
1	6	Nucleus Longest Diameter	0.105630
2	13	Nucleus Perimeter	0.094532
3	0	Nucleus Area	0.089053
4	2	N/C ratio	0.085828
5	1	Cytoplasm Area	0.059664
6	16	Nucleus maxima	0.058665
7	5	Nucleus shortest diameter	0.058234
8	18	Cytoplasm maxima	0.046058
9	19	Cytoplasm Minima	0.041601
10	9	Cytoplasm shortest diameter	0.041268
11	14	Cytoplasm perimeter	0.038186
12	17	Minimum in nucleus	0.038090
13	3	Nucleus Brightness	0.025289
14	10	Cytoplasm longest diameter	0.034507
15	4	Cytoplasm brightness	0.024231
16	12	Cytoplasm roundness	0.022669
17	15	Nucleus relative position	0.022652
18	11	Cytoplasm elongation	0.017863
19	22	Correlation	0.014712
20	7	Nucleus elongation	0.014374
21	8	Nucleus roundness	0.013968
22	20	Energy	0.012616
23	21	Contrast	0.010222
24	24	Homogeneity	0.010083
25	23	Energy	0.010004

Figure 6. Features importance

The following online and offline machine learning algorithms have been used during this research.

2.7.1. Linear Classifiers With Stochastic Gradient Descent Learning (SGD)

Gradient descent is the process to minimize the cost function $J(\theta)$ with respect to model operator $\theta \in \mathbb{R}^d$ by updating its parameter in reverse direction of the gradient of objective function $\nabla_{\theta} J(\theta)$ with respect to θ (Ruder, 2016).

SGD performs one update at a time as compared to batch gradient descent algorithm. The parameter θ , update in SGD algorithm for each training sample $x^{(i)}$, and corresponding label $y^{(i)}$ is defined as:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i)}; y^{(i)}) \quad (7)$$

where η is a learning rate in between 0 to 1, is used to define step size.

2.7.2. Perceptron

Perceptron is a binary classifier for linear separable patterns. Updation of weight vector for perceptron has been explained by (Haykin, 2005) in his book as:

$$W_{n+1} = W_n + \eta \cdot (D_n - Y_n) \cdot X_n \quad (8)$$

where $D_n - Y_n = \text{error}_n$, η is learning rate and X_n is input vector.

2.7.3. Passive Aggressive Classifier With Loss Function Hinge (PA- I)

Hinge loss function defined in (Crammer et al., 2006), for a weight vector W and input vector X as:

$$l(W; (X, y)) = \begin{cases} 0 & \text{if } y(W \cdot X) \geq 1 \\ 1 - y(W \cdot X) & \text{otherwise} \end{cases} \quad (9)$$

The loss is equal to zero if margin exceeds to 1, otherwise it is the difference between margin value and 1.

The weight of passive – aggressive algorithm I explained in (Crammer et al., 2006) is:

$$W_{t+1} = W_t + \zeta_t \cdot Y_t \cdot X_t, \text{ where } \zeta_t = \min\{C, l_t / \|X_t\|^2\} \quad (10)$$

Here C is aggressiveness parameter, l_t is loss function, X_t is input vector, W_t is weight vector at round t .

2.7.4. Passive Aggressive Classifier With Loss Function Squared Hinge (PA- II)

The weight of passive – aggressive algorithm II explained in (Crammer et al., 2006) in his research work is updated as:

$$W_{t+1} = W_t + \zeta_t \cdot Y_t \cdot X_t, \text{ where } \zeta_t = \frac{l_t}{\|X_t\|^2 + \frac{1}{2C}} \quad (11)$$

and $l_t = \max\{0, 1 - Y_t(W_t \cdot X_t)\}$ is loss function, X_t is input vector, W_t is weight vector at round t .

2.7.5. Logistic Regression Classifier With Stochastic Average Gradient Learning (SAG)

For each iteration, SAG randomly selects any sample from training data, calculates the gradient of this sample, stores and updates corresponding item in N^D vector space. So each element in vector is computed in different iteration (Chen et al., 2017). SAG uses given formula to update its weight as:

$$W^{k+1} = W^k - \frac{h}{N} \sum_{j=1}^C \phi_k^j \quad (12)$$

and:

$$j_k = \begin{cases} G_j(W_k) & \text{if } j = j_k \\ \phi_k^{j-1} & \text{otherwise} \end{cases} \quad (13)$$

where parameter h is step size, N is data size, C is number of classes.

2.7.6. Logistic Regression With L1 Regularization

Regularization is the way to reduce complexity of model, avoid overtraining and for better generalization of model. A generalized model can be trained on some data set and perform better on unknown dataset. The cost function of logistic regression by adding L1 regularization term is defined in (Raschka, 2015) as:

$$aJ(W) = C \left[\sum_{j=1}^m \left(-\log(\phi(Z^{(i)})) + (1 - Y^{(i)}) \left(-\log(1 - \phi(Z^{(i)})) \right) \right) \right] + \sum_{j=1}^m |W_j| \quad (14)$$

where $Y^{(i)}$ is target for input sample $X^{(i)}$ and $\phi(Z) = 1/(1+e)$ is logistic function or activation function used to compute expected output.

2.7.7. Support Vector Machine With Linear Kernel (svmLK)

Support vector machine uses support vectors to draw an optimal hyperplane between patterns having maximum margin from both classes (Khamparia, & Pandey, 2018). Dual form for constrained optimization of support vector machine explained in (Haykin, 2005) as the following.

Find the values of Lagrange multipliers $\{\alpha_i\}_{i=1}^N$ that maximize the objective function for given training samples $\{x_i, d_i\}_{i=1}^N$:

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j K(X_i, X_j) \quad (15)$$

Subjected to the constraints:

$$\sum_{i=1}^N \alpha_i d_i = 0 \text{ and } 0 \leq \alpha_i \leq C$$

where C is positive parameter and $K(X_i, X_j)$ is called kernel of machine. The linear kernel machine can be designed by setting power p of polynomial learning machine equals to 1, as the following.

Polynomial learning machine kernel, $K(X, X_i) = (1 + X^T X_i)^p$, where p is specified a priori by user and X^T represents transpose of column vector X . If $p = 1$ called linear kernel machine.

2.7.8. Support Vector Machine With Radial Basis Function Kernel (svmRBF)

Radial basis function kernel can be defined in (Haykin, 2005) as:

$$K(X, X_i) = e^{-\frac{\|X - X_i\|^2}{2\sigma^2}} \quad (16)$$

where width σ^2 is specified a priori by the user.

2.7.9. K-Nearest Neighbour Classifier (KNN)

KNN is a typical lazy learner that will not learn the pattern but it will memorize it. It will compute the similarity in term of distance for the patterns to its K neighbours and it will classify according frequencies that matches to number of neighbours. For this research, we have used 5 nearest neighbour with similarity as Minkowski distance (Raschka, 2015) is:

$$d(X^i, X^j) = \sqrt[p]{\left(X_k^i - X_k^j\right)^p} \quad (17)$$

where $p=1$ the similarity is known as Manhattan distance and $p=2$ it is called Euclidean distance.

2.7.10. Decision Tree Classifier (DTree)

Decision tree classifiers act as a model that breaks our data by making a decision, the breaking points are called node and all leaf nodes are labelled class. In decision tree, the objective function is to maximize information gain at each split or breaking defined in (Raschka, 2015) as:

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^m \frac{N_j}{N_p} I(D_j) \quad (18)$$

Here, f is a feature to perform the split, D_p and D_j are the dataset of parent and j^{th} child node, I is impurity measure. N_p and N_j is the number of samples at a particular node and in j^{th} child node.

2.7.11. Random Forest Classifier (Rforest)

Random forest classifier can be considered as ensemble of decision trees, most of time trained with bagging method. Bagging is a method that reduce variance of a model by drawing random bootstrap samples from training sets and combine individually trained classifier via majority vote (Raschka, 2015).

2.7.12. Gaussian Naïve Bayes Classifier (GNB)

Naïve Bayes classifier based on decision rule that choose the hypothesis which is most preferable. The target class output from naïve Bayes classifier to assign class C_k for some k is defined (Mitchell, 1997) as:

$$v_{nb} = \underset{k \in \{1,2,\dots,K\}}{\operatorname{argmax}} P(C_k) \prod_{i=1}^N P(X_i | C_k) \quad (19)$$

Here, $P(C_k)$ is probability of k^{th} class, N is total number of samples and K is total number of class.

2.7.13. Gradient Boosting Classifier (GBT)

Gradient boosting classifier can be used for binary or multiclass classifier, has following form defined explained by (Buitinck, 2013) as:

$$F(x) = \sum_{m=1}^M \gamma_m h_m(x) \quad (20)$$

Here $h_m(x)$ represent basis function usually a weak learners. Gradient boosting classifiers uses decision tree of fixed size as weak learners.

3. RESULT AND FINDINGS

All the experiments have been done in language python on personal computer with 4 GB RAM, Intel® core™ i3-7310U CPU @ 2.70 GHZ. The benchmarked Pap smear dataset has been classified in 2 class namely normal and abnormal cells using machine learning algorithms. The evaluation of classifiers has been done on various performance metrics and CPU time.

To measure generalized performance of the algorithms, we have used holdout cross-validation techniques. For this purpose, training and test samples are randomly divided in 9 different sets containing training size {10, 20, 30, 40, 50, 60, 70, 80, 90} % of original data and vice versa for test

size for 100 times. We have calculated the performance metrics along with CPU time for each holdout proportion. Algorithm used to compute performance matrices and CPU time is shown in Table 2.

3.1. Performance Metrics

To compare the performance of machine learning algorithms as discussed in Section 2.7, the authors has used Accuracy, Precision Score, Recall score and F1 score metrics. (Raschka, 2015). Open source python packages has been used by authors to calculate and plot these performance metrics.

Performance metrics such as accuracy, precision score, recall score and F1 score are computed using algorithm discussed in Table 2 for every holdout proportion for training size {10, 20, 30, 40, 50, 60, 70, 80, 90}% of original data and vice versa for test size are shown in Table 3, Table 4, Table 5 and Table 6 respectively.

Accuracy: Accuracy of a classifier is the measure of percentage of data accurately classified in correct class. E.g. out of 500 samples, if a classifier predicts 200 correct positive class and 200 correct negative class remaining 100 are misclassified, then the accuracy is 80%.

For a given classifier if:

TP: True Positive, means positive class data classified in positive class by classifier.

TN: True Negative, means negative class data classified in negative class by classifier.

FP: False Positive, means negative class data classified in positive class by classifier.

FN: False Negative, means positive class data classified in negative class by classifier.

Table 2. Algorithm to compute performance metrics and CPU time

Step 1: initialize number of round (round), number of algorithm (num_algo), training size (TS) round =100 num_algo=13 TS=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
Step 2: initialize for loop (to deal with all 13 algorithms) for k=1 to num_algo
Step 3: initialize inner for loop of step 2 (to deal with each training size) for training_size in TS
Step 4: initialize inner for loop of step 3 (to deal with each rounds) for r=1 to round
Step 5: randomly split samples to training samples of size equal to TS, and testing sample of size = 1 – TS
Step 6: initialize timer as START (to measure CPU execution time for training and testing)
Step 7: train k^{th} classifier
Step 8: compute output of k^{th} classifiers
Step 9: STOP timer and compute CPU TIME=STOP TIME-START TIME
Step 10: compute accuracy, precision score, recall score and F1 score
Step 11: END of for loop of step 4
Step 12: compute the average of all accuracy, precision, recall, F1 score and execution_time (save the values for corresponding to each training size)
Step 13: END of for loop of step 3
Step 14: END of for loop of step 2
Step 15: END of program

Table 3. Accuracy (in %) of machine learning algorithms

Algorithms	Training Size								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
SGD	93.34	94.01	93.77	94.92	94.99	95.1	95.65	96.74	97.83
Perceptron	92.74	94.01	94.39	94.01	95.64	95.64	95.29	95.11	98.91
PA-I	93.7	94.14	94.55	94.56	95.42	94.55	96.01	96.74	98.91
PA-II	94.07	94.41	94.39	94.74	94.55	94.55	96.74	96.74	97.83
SAG	93.46	94.41	94.24	95.1	96.08	95.64	96.01	98.37	100
LR	93.34	92.92	94.08	94.92	95.42	95.91	96.38	97.28	100
svmLK	93.58	94.55	94.7	95.28	96.51	96.19	96.74	97.28	100
svmRBF	92.74	94.55	94.86	95.1	96.08	95.64	97.1	97.83	97.83
KNN	92.01	93.6	93.46	94.37	94.34	95.1	96.38	96.2	98.91
Dtree	91.89	92.64	93.46	93.83	94.99	95.1	95.65	95.65	98.91
Rforest	93.22	93.87	95.02	94.92	95.42	96.46	98.19	97.83	98.91
GNB	92.13	92.64	92.37	93.65	93.25	94.28	94.57	96.74	98.91
GBC	92.25	92.92	93.61	94.01	94.12	95.37	94.93	96.2	97.83

Table 4. Precision score (in %) of algorithms

Algorithms	Training Size								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
SGD	97.07	98.26	98.18	97.64	99.31	98.16	99.02	99.25	100
Perceptron	97.1	98.62	97.81	98.28	98.39	98.02	98.86	100	100
PA-I	96.88	98.47	98.58	98.16	98.63	99.1	98.41	99.21	100
PA-II	97.79	98.29	97.57	98.92	98.44	99.58	99.49	100	100
SAG	96.66	96.41	96.4	96.85	97.14	96.73	97.6	98.6	100
LR	96.68	97.18	96.94	97.2	97.52	99.23	97.98	97.92	100
svmLK	97.02	97.05	96.94	96.8	96.98	97.16	97.21	99.28	100
svmRBF	95.77	95.46	96.67	96.13	96.5	96.47	97.61	97.92	100
KNN	92.63	93.84	94.59	95.56	94.81	96.07	97.61	96.03	100
Dtree	95.08	95.57	96.15	96.55	97.19	98.15	98.11	100	100
Rforest	96.61	95.78	95.85	96.31	96.3	97.16	98.58	99.25	100
GNB	95.94	94.31	94.26	95.56	93.67	93.84	94.76	95.83	98.57
GBC	96.58	96.38	97.99	96.78	96.9	97.42	97.97	97.87	98.63

Table 5. Recall score (in %) of algorithms

Algorithms	Training Size								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
SGD	99.01	99.63	99.37	99.51	99.4	100	100	100	100
Perceptron	98.66	99.26	99.37	99.76	99.11	100	98.99	100	100
PA-I	99.83	99.63	99.36	99.75	99.1	99.63	100	100	100
PA-II	99.02	99.63	99.79	99.75	99.4	99.63	99.52	100	100
SAG	99.17	99.81	99.37	99.01	99.11	98.92	99.53	100	100
LR	97.84	97.4	97.17	98.51	97.61	98.92	99	99.28	100
svmLK	99.67	99.45	99.15	98.75	99.12	99.28	99.51	100	100
svmRBF	100	100	100	99.75	99.71	99.63	100	100	100
KNN	100	100	99.79	100	99.7	100	100	100	100
Dtree	99.01	97.41	98.29	96.86	98.18	97.84	98.06	98.58	100
Rforest	99.83	99.44	98.75	99.5	99.41	99.63	99.53	100	100
GNB	99.18	99.26	99.15	99.5	99.7	99.64	99.53	100	100
GBC	98.36	98.88	98.11	98.5	98.45	98.13	98.13	99.26	100

Table 6. F1 scores (in %) of algorithms

Algorithms	Training Size								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
SGD	95.52	95.96	95.85	96.53	96.67	96.68	97.03	97.93	98.59
Perceptron	95.16	95.94	96.2	96.02	97.03	97.15	96.81	96.91	99.29
PA-I	95.67	96.13	96.38	96.46	96.8	96.26	97.39	97.93	99.28
PA-II	96.03	96.35	96.27	96.5	96.3	96.42	97.79	97.83	98.48
SAG	95.6	96.3	96.21	96.74	97.42	97.15	97.39	98.95	100
LR	95.52	95.2	96	96.55	96.89	97.34	97.52	98.22	100
svmLK	95.73	96.29	96.52	96.86	97.74	97.51	97.89	98.17	100
svmRBF	95.14	96.37	96.58	96.78	97.41	97.13	98.08	98.5	98.68
KNN	94.88	95.78	95.77	96.32	96.39	96.76	97.61	97.51	99.28
Dtree	94.6	95	95.68	95.93	96.57	96.8	97.2	97.24	99.34
Rforest	95.45	95.9	96.77	96.62	97.05	97.63	98.82	98.6	99.3
GNB	94.87	95.15	94.96	95.77	95.69	96.31	96.54	97.87	99.28
GBC	94.82	95.22	95.8	95.99	96.1	96.95	96.55	97.58	98.61

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} * 100 \quad (21)$$

Table 3 highlights that accuracy of the algorithms is getting better as increase in training size. The best accuracy achieved by logistic regression with stochastic average gradient learning and with L1 regularization and support vector machine with linear kernel algorithm at 90% training size and 10% testing size of 803 samples. Accuracy of machine learning algorithms as shown in Table 3 is compared in Figure 7. For better understanding the author has compared algorithms at training size 10% i.e. minimum sample size and training size at 70%, 80% and 90% having better accuracies.

3.1.1. Precision Score

Precision score is the rate of true positive prediction of a classifier versus total positive prediction. For example, if a classifier predicts 100 positive samples but out of which only 90 are true positive, the precision score is 90%:

$$Precision\ Score = \frac{TP}{TP + FP} * 100 \quad (22)$$

Perceptron and decision tree algorithm have precision score of 100% for training size greater than equal to 80% whereas except gaussian naïve Bayes and gradient boosting classifier algorithm all algorithms have achieved 100% precision at training size equals to 90% of 803 samples. For simplicity, Figure 8 compares the precision scores of various algorithms at training size 10%, 70%, 80% and 90% of original samples.

Figure 7. Accuracy graph of algorithms

Figure 8. Precision score of algorithms

3.1.2. Recall Score

Recall score is the rate of true positive prediction of a classifier versus total positive given target class. For example, if a classifier was given a data with 200 positive samples but only 190 are correctly classified as positive, then the recall score is 95%:

$$Recall\ Score = \frac{TP}{TP + FN} * 100 \quad (23)$$

All algorithms except decision tree algorithm achieve recall score approximately 100% if training data size is 80% or more. Decision tree algorithm also achieves 100% recall score on training size of 90% of samples. Figure 9 compares the recall scores of various algorithms. For simplicity, the authors has compared the scores at only training size equal to 10%, 70%, 80% and 90% of 803 samples.

3.1.3. F1 Score

F1 score is the combination of both precision score and recall score:

$$F1\ Score = 2 * \frac{Precision\ Score * Recall\ Score}{Precision\ Score + Recall\ Score} * 100 = \frac{2 * TP}{2 * TP + FP + FN} * 100 \quad (24)$$

All algorithms have better F1 score when training size is increased. Logistic regression with stochastic average gradient learning and with L1 regularization and support vector machine with linear kernel has better F1 score than remaining algorithms at training size equal to 90%. For simplicity, the authors compared F1 scores of various algorithms in Figure 10 at training size 10%, 70%, 80% and 90% of 803 samples.

Figure 9. Recall score of algorithms

Figure 10. F1-score of algorithms

3.2. CPU Computational Time

CPU time for all algorithms has been calculated using pseudo code as discussed in Table 2. CPU time includes training and classification time in seconds. CPU computational time is machine dependent, which may vary time to time depend on CPU processing speeds. To compute CPU time, we have used open source python time package. Figure 11 compares mean CPU time of all algorithms.

The authors have computed CPU time for each algorithms for every training proportion, as shown in Table 7. CPU time taken by these machine learning algorithms includes training time, testing time and an algorithm to predict cervical cell as normal or abnormal.

Figure 11. Average CPU running time

Table 7. CPU time and mean CPU time taken by algorithms

Algorithms	Training Size									
	10%	20%	30%	40%	50%	60%	70%	80%	90%	Mean CPU Time
SGD	0.13	0.11	0.11	0.15	0.18	0.17	0.16	0.18	0.19	0.16
Perceptron	0.13	0.11	0.11	0.12	0.13	0.13	0.15	0.13	0.17	0.14
PA-I	0.1	0.11	0.13	0.11	0.13	0.15	0.15	0.16	0.16	0.14
PA-II	0.1	0.11	0.13	0.11	0.13	0.15	0.16	0.15	0.16	0.14
SAG	0.15	0.16	0.19	0.19	0.21	0.26	0.26	0.26	0.29	0.22
LR	0.76	15.18	23.03	14.03	14.38	15.64	15.72	17.86	20.21	15.21
svmLK	0.18	0.37	0.4	0.52	0.7	0.83	0.99	1.15	1.32	0.72
sbmRBF	0.32	0.54	0.59	0.91	0.82	1.06	1.34	1.41	1.59	0.96
KNN	0.58	0.93	1.32	1.27	1.54	1.81	1.11	0.92	0.67	1.13
Dtree	0.25	0.4	0.5	0.77	0.87	1.16	1.28	1.64	2.14	1.01
Rforest	14.37	16.94	19.26	21.78	24.13	27.96	31.26	33.49	36.88	25.12
GNB	0.18	0.17	0.13	0.14	0.15	0.15	0.2	0.19	0.16	0.17
GBC	2.39	3.55	4.33	5.21	5.44	6.17	6.89	7.58	8.12	5.52

Logistic regression with L1 regularization and random forest classifiers have taken more computation time than remaining algorithms.

3.3. Discussion and Findings

In this article, the authors made a performance analysis of various machine learning algorithms to detect the cervical cancer and some of the important findings are as follows:

- Algorithm like logistic regression with L1 regularization has achieved accuracy up to 100% but it is more expensive in term of CPU time in comparison to some of the algorithms having accuracy near about 99% with less CPU time;
- Random forest classifier having maximum accuracy near about 99%, but it is most expensive algorithm in term of CPU time;
- The online algorithms like perceptron, passive aggressive - I having accuracy near about 99% and computed in very less time as compared to other algorithms;
- Other online algorithm i.e. stochastic average gradient (SAG) and offline algorithm i.e. linear support vector machine have accuracy up to 100% while online algorithm has less CPU time as compared to offline algorithm;
- Precision score of almost all algorithms except gaussian naïve Bayes and gradient boosting classifier is up to 100% that means false positive (FP) rate equals to zero which indicates these algorithms are suitable for cervical cancer detection;
- Recall score of all algorithms achieved up to 100% that means false negative (FN) rate equals to zero for all algorithms.

4. CONCLUSION AND SCOPE OF FUTURE WORK

Cervical cancer is second most prevailing cancer in women all over the world and pap smear is one of the most popular techniques used to diagnosis cervical cancer at an early stage. It is a challenging task for the doctors to diagnose cases of cervical cancer in fixed prescribed time with limited resources. In this paper 2-class problem of cervical cancer diagnosis is experimented using online and offline machine learning algorithms. Major focus has been laid on hybrid segmentation techniques used for segmentation of Pap smear images in nucleus, cytoplasm and background, selection of most important features using extra tree classifier. Various machine learning algorithms have been evaluated on performance metrics and CPU time. The best accuracy achieved is 100% which indicates the importance of machine learning algorithms for cervical cancer detection. Moreover, the precision score, recall score and F1 score of some algorithms are up to 100%, which indicates reliability of this work.

Various other datasets can be tested by taking hybrid features. Other machine learning techniques can also be applied in future to observe the performance of classifier to detect the cervical cancer for 2-class, 4-class and 7-class classification problem.

ACKNOWLEDGMENT

The authors thank to MDE LAB to provide data sets available online on website. We also thank to office on women's health, national institute of health, American cancer society, and international agency for research on cancer to provide facts on cervical cancer which motivates authors to do research work.

REFERENCES

- American cancer society. (2018), *Signs and symptoms of cervical cancer*. Retrieved from <https://www.cancer.org/cancer/cervical-cancer/detection-diagnosis-staging/signs-symptoms.html>
- Ampazis, N., Dounias, G., & Jantzen, J. (2004). *Efficient second order neural network training algorithms for the construction of a pap-smear classifier*.
- Bora, K., Chowdhury, M., Mahanta, L. B., Kundu, M. K., & Das, A. K. (2016). Pap smear image classification using convolutional neural network. In *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP '16)*. New York: ACM. doi:10.1145/3009977.3010068
- Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project.
- Byriel, J. (1999). Neuro-fuzzy classification of cells in cervical smears [MSc Thesis]. Technical University of Denmark.
- Chankong, T., Theera-Umpon, N., & Auephanwiriyakul, S. (2014). Automatic cervical cell segmentation and classification in pap smears. *Computer Methods and Programs in Biomedicine*, 113(2), 539–556. doi:10.1016/j.cmpb.2013.12.012 PMID:24433758
- Chen, A., Chen, B., Chai, X., Bian, R., & Li, H. (2017). A novel stochastic stratified average gradient method: convergence rate and its complexity. *CoRR*, abs/1710.07783.
- Chen, Y. F., Huang, P. C., Lin, K. C., Lin, H. H., Wang, L. E., Cheng, C. C., & Chiang, J. Y. et al. (2014). Semi-automatic segmentation and classification of pap smear cells. *IEEE Journal of Biomedical and Health Informatics*, 18(1), 94–108. doi:10.1109/JBHI.2013.2250984 PMID:24403407
- Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online passive-aggressive algorithm. *Journal of Machine Learning Research*, 7, 551–585.
- Dounias, G. (2017, February 6). Pap-smear (DTU/HERLEV) databases & related studies [Data file]. Retrieved from <http://mde-lab.aegean.gr/downloads>
- Dounias, G., Bjerregaard, B., Jantzen, J., Tsakonas, A., Ampazis, N., Panagi, G., & Panourgias, E. (2006). Automated identification of cancerous smears using various competitive intelligent techniques. *Oncology Reports Journal*, 15(4), 1001–1006. PMID:16525690
- Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1), 3–42. doi:10.1007/s10994-006-6226-1
- Haykin, S. (2005). *Neural networks a comprehensive foundation* (2nd ed.). Singapore: Pearson education.
- International Agency for research on Cancer (IARC). (2018), *GLOBOCAN 2012: Estimated Cancer Incidence, Mortality and Prevalence Worldwide in 2012*. Retrieved from http://globocan.iarc.fr/Pages/burden_sel.aspx
- Jantzen, J., & Dounias, G. (2006). Analysis of pap-smear data. In *Proceedings of the European Symposium on Nature-inspired Smart Information Systems (NiSIS 2006)*, Puerto de la Cruz, Tenerife, Spain.
- Khamparia, A., & Pandey, B. (2018). SVM and PCA Based Learning Feature Classification Approaches for E-Learning System. *International Journal of Web-Based Learning and Teaching Technologies*, 13(2), 32–45. doi:10.4018/IJWLTT.2018040103
- Marinakakis, Y., Dounias, G., & Jantzen, J. (2009). Pap smear diagnosis using a hybrid intelligent scheme focusing on genetic algorithm based feature selection and nearest neighbor classification. *Computers in Biology and Medicine*, 39(1), 69–78. doi:10.1016/j.combiomed.2008.11.006 PMID:19147127
- Mitchell, T. M. (1997). *Machine learning*. New York: McGraw-Hill education.
- Mustafa, N., Mat-Isa, N. A., Manshor, M. Y., & Othman, N. H. (2008). Capability of new features of cervical cells for cervical cancer diagnostic system using hierarchical neural network. *International Journal of Simulation: System. Science and Technology, IJSSST*, 9(2), 56–64.

National institutes of health. (2018). *Tests that examine the cervix are used to detect (find) and diagnose cervical cancer*. Retrieved from <https://www.cancer.gov/types/cervical/patient/cervical-treatment-pdq>

Office on women's health, U.S. Department of Health and Human Services. (2018). *Cervical Cancer*. Retrieved from <https://www.womenshealth.gov/cancer/cervical-cancer>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, E. et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12, 2825–2830.

Raschka, R. (2015). *Python machine learning*. Retrieved from www.it-ebooks.info

Ruder, S. (2016). An overview of gradient descent optimization algorithms.

Singh, A., & Pandey, B. (2014). Intelligent techniques and applications in liver disorders: a survey. *International Journal of Biomedical Engineering and Technology*, 16(1), 27-70. doi:065638 doi:10.1504/IJBET.2014

Takkar, S., Singh, A., & Pandey, B. (2017). *Application of machine learning algorithms to a well defined clinical problem-liver disease*. *International Journal of E-Health and Medical Communications*, 8(4). doi:10.4018/IJEHMC.2017100103

Zhang, L., Lu, L., Nogues, I., Summers, R. M., Liu, S., & Yao, J. (2017). DeepPap: Deep convolutional networks for cervical cells classification. *IEEE Journal of Biomedical and Health Informatics*, 21(6), 1633–1643. doi:10.1109/JBHI.2017.2705583 PMID:28541229

Zhang, L., Lu, L., Summers, R. M., Kebebew, E., & Yao, J. (2018). Convolutional invasion and expansion networks for tumor growth prediction. *IEEE Transactions on Medical Imaging*, 37(2), 638–648. doi:10.1109/TMI.2017.2774044 PMID:29408791

Sanjay Kumar Singh received B. Tech and M. tech degrees from ABV-Indian Institute of Information Technology and Management, Gwalior in Information Technology in 2011. He is pursuing his PhD from I.K. Gujral Punjab Technical University, Punjab. His research interests include medical image processing, speech signal processing, and intelligent systems.

Anjali Goyal has received her bachelor's degree in electronics in 1993 from Kurukshetra University and a master's degree in computer applications in 1996 from Panjab University, Chandigarh. She received her PhD degree from Punjab Technical University, Jalandhar, India in 2013. Her academic achievement includes a University Merit position in graduation. Presently, she is working as an Assistant Professor in the Department of Computer Application at the Guru Nanak Institute of Management and Technology, Ludhiana affiliated to PTU. Earlier, she served in various colleges of Kurukshetra University and Punjab Technical University. She has 18 years of teaching experience. Her research interests include content base image retrieval, digital watermarking and pattern recognition. She has a number of international journal and conference publications to her credit. She is reviewer of many reputed international journals. She is also guiding many PhD students.