Scalable Multimodal Machine Learning for Cervical Cancer Detection

Conference Paper · May 2024

DOI: 10.1109/AIIoT61789.2024.10578984

5 authors, including:

Jiblal Upadhya
Middle Tennessee State University
10 PUBLICATIONS **21** CITATIONS
SEE PROFILE

Samir Poudel
Middle Tennessee State University
16 PUBLICATIONS **46** CITATIONS
SEE PROFILE

Satish Wagle
Middle Tennessee State University
4 PUBLICATIONS **4** CITATIONS
SEE PROFILE

Scalable Multimodal Machine Learning for Cervical Cancer Detection

Thuan Nhan †, Jiblal Upadhya†, Samir Poudel*, Satish Wagle†,
Khem Poudel*
Department of Computer Science*, Computational and Data Science†
Middle Tennessee State University
Murfreesboro, TN, 37132
Email: {tnn2u, ju2i, sp2ai, sw8k}@mtmail.mtsu.edu,
{khem.poudel}@mtsu.edu

***Abstract*—Cervical cancer poses a significant global health concern, and machine learning techniques have shown great promise in early detection and diagnosis. However, the dominant focus on accuracy has overshadowed the crucial aspect of training time. This research paper investigates multimodal high-performance machine learning for cervical cancer detection, with a primary emphasis on scalability and training time optimization using Apache Spark. We conduct experiments on two distinct datasets – CSV and Images – analyzing the trade-offs between training time and accuracy. Moreover, we discuss the challenges encountered during the implementation of Spark. Our study helps to shed light on Spark's potential benefits and limitations in high-performance machine learning for cervical cancer detection. Our findings indicate that Spark ML showcases remarkable scalability potential compared to traditional platforms like SkLearn for text datasets. The majority of the ML algorithms explored showcased exceptional accuracy rates, with many surpassing the impressive threshold of 98 % in text data. In particular, we have shown that, Spark helps to reduce the time complexity more than three folds when the data size increases.**

***Index Terms*—Spark, Cervical Cancer, SparkML, SipakMed, Scalability**

I. Introduction

Cervical cancer is a type of cancer that predominantly affects the cervix in women. According to the World Health Organization (WHO) [1], it was the fourth most common cancer among women in 2020, with 90% of new cases and deaths occurring in low and middle-income countries. WHO recommends four different tests for detection, including HPV DNA testing, visual inspection of acetic acid, biopsy procedures, and liquid-based cytology [2]. However, these tests are expensive, time consuming, and unsuitable for large-scale testing. Consequently, there is a pressing need for a better and more efficient cervical cancer detection tool [3]. In the field of healthcare, machine learning has gained relevance and can be used to create advanced detection tools [4]. One of the key advantages of machine learning is its ability to rapidly deploy applications with a low barrier of entry for users, including healthcare professionals [5]. Furthermore, machine learning can greatly enhance accuracy by analyzing vast amounts of data from multiple resources [6], allowing the identification of patterns and trends that may not be discernible to the human eye alone.

Using the potential of machine learning, a new generation of cervical cancer detection tools can be developed [7]. These tools promise to provide a more efficient and accurate solution, significantly improving patient lives, reducing costs, decreasing mortality rates, and increasing the effectiveness of prevention programs. With seamless integration of machine learning, the limitations of current detection methods can be addressed, marking a significant step forward in the fight against cervical cancer. [8] tries to combine the usage of IoT devices and different machine learning models to reach results as high as 0.98% on average.

Multimodal deep learning has significant applications in cervical cancer detection and diagnosis, as demonstrated by various research studies [9], [10]. In this setting, a imaging modality with other clinical labels, biomarkers data, etc in the textual formats are analyzed which gives the complementary informations about the subjects and hence helps in capturing the more realistic informations than unimodals [11], [12]. Hence, by integrating different types of data in our research, we envision this will help in enhancing diagnostic accuracy for better reliability and efficiency of the medical conditions like cervical cancer detection.

However, most paper pay more attention to accuracy, which can be an oversight. In the era of Big Data, machine learning models must not only be accurate but also be able to handle large volumes of data in an efficient manner. This is a reasonable expectation as IoT devices are being used to improve accuracy and improve the life of users [8]. Since IoT devices can generate a large amount of data, other factors such as training time and scalability should also be considered.

In order to effectively handle data-rich data sets, parallelization is required [13]. However, implementing parallelization can be tricky and system dependent. Additionally, code portability can also be a concern for parallelized implementation. Spark can be a solution to this problem. Spark is a big data platform that has a massive performance advantage compared to other big data platforms such as Hadoop MapReduce [14]. SparkML, the Spark's machine learning component, provides a scalable and parallelized machine learning library. In this project, we explore the usage of SparkML for Cervical Cancer prognosis and evaluate its accuracy and performance compared

to nonparallelized platforms such as SkLearn. Our research makes following key contributions:

1. *Development of a Pipeline*: We introduce a pipeline that demonstrates how Spark can serve as an alternative to conventional machine learning tasks, showcasing its potential to enhance efficiency and scalability in cervical cancer prediction.

2. *Performance Comparison* We conduct an analysis to understand the differences in performance—focusing on both training time and accuracy—between Spark and traditional machine learning models.

3. *Handling Image Data* We propose a methodology for processing image data within Spark. This approach allows for the integration of image-based analysis in Spark's ecosystem, expanding its applicability in medical imaging tasks.

4. *Multimodal Approach to Cervical Cancer Detection* Our study approaches cervical cancer detection from two angles: analyzing structured data in CSV format and unstructured image data. This multimodal strategy underscores the versatility of Spark in processing diverse data types for healthcare applications.

Our research aims to underscore Spark's capabilities within the realm of big data and machine learning, particularly focusing on classification tasks and applications in Cervical Cancer.

II. Literature Review

Most of the research on cervical cancer prediction has focused predominantly on the Cervical Cancer Risk Classification dataset [15], a CSV text file. However, limited exploration has been made in harnessing images of cervical cells for prediction. Although some research groups have developed proprietary image datasets, the only publicly available labeled dataset is SipakMed [16], introduced in 2018. Remarkably, this particular dataset remains largely underexplored in the research domain. Numerous papers have explored the application of machine learning in cervical cancer prediction, with a primary focus on the Cervical Cancer Risk Classification Dataset [15]. For example, in [17], the authors evaluated four machine learning models: Naive Bayes, Decision Tree, Logistic Regression and Random Forest using the aforementioned dataset. Among them, the Decision Tree model exhibited an impressive accuracy of 96.8%.

Expanding the scope, another study, [18], delved into various machine learning models, including Logistic Regression, Bagging with Decision Tree as the base, Random Forest, and XG-Boost (Extreme Gradient Boost). Each proposed model demonstrated accuracies greater than 95%, with XG-Boost attaining the highest accuracy of 97.08%.

Furthermore, in [19], researchers explored different machine learning algorithms, culminating in an impressive accuracy of 99.71% using Random Forest. Beyond accuracy, this study also focused on the importance of the characteristics and the explainability of the model. Employing SHAP (SHapley Additive exPlanations), the authors identified key influential features affecting the model's performance, such as Schiller, Hinselmann, Age, and Hormonal Contraceptive.

Despite achieving remarkable accuracy levels, research related to improving the scalability of machine learning algorithms for the prediction of cervical cancer remains scarce. One notable exception is the work by [20], which used a multi-objective genetic algorithm to reduce the dimensions of the characteristics while maintaining a high precision of 0.965. In line with addressing this gap, our research aims to prioritize scalability as a critical performance metric in the context of cervical cancer prediction. By exploring both traditional CSV text files and cell images, we aim to advance the accuracy and applicability of cervical cancer prediction models.

III. Methodology

A. MapReduce

MapReduce is a programming framework originally introduced by Google to handle large datasets [21]. MapReduce allows data to be processed across cluster of commodity hardware [22]. The MapReduce framework consists of 2 procedures, map and reduce. The map step is where the data is preprocessed. The reduce step is where the data are aggregated. In a typical MapReduce workflow, input is divided into smaller chunks. Then the small chunks of data are processed independently across different nodes in the cluster. The results from each node are then combined for the final output.

One of the most popular MapReduce implementations is Hadoop MapReduce. MapReduce is known for scalability and fault tolerance. MapReduce is used for batch processing and offline data processing jobs.

B. Spark and SparkML

Apache Spark address the limitations of Big Data processing frameworks such as MapReduces. Spark expands on the original MapReduce model by adding more feature sets and optimization [23]. Fig.1 shows some of the differences between Spark and MapReduce. Sparks offers caching itermediated data in RAM, which significantly increases performance compared to disk-based processing in MapReduce. Spark also provides support and integration with other programming languages and platforms. Spark also provided higher-level API and more expressive programming models thus lower the barrier of entry compared to MapReduce.

SparkML is a Spark library that offers a scalable and distributed platform for machine learning. It supports multiple programming languages and streamlines the development of machine learning pipelines [24]. Through SparkML, users can easily deploy parallelized machine learning models on extensive datasets. By providing a high-level API for a wide range of machine learning tasks, SparkML significantly reduces the complexity of implementing distributed machine learning solutions. Furthermore, it incorporates implementations of several well-known machine learning algorithms, enhancing the efficiency of model deployment.

Features	Apache Spark	MapReduce
Data Processing Model	Batch processing, real time streaming, iterative task	Batch processing
In-Memory computing (RAM)	Yes	No
Processing Speed	Generally faster (In-Memory computing)	Slower (Disk based processing)
Ease of use	Higher level API and more expressive programming models	Lower-level API, more complex and verbose programming
Fault Tolerance	Yes	Yes
Scalability	Highly scalable	Highly scalable
Language Support	Java, Scala, Python, R	Java, support for other language is available but limited
Libraries	Multiple different built in libraries – (SparkMLLib, GraphX, ...)	Limited
Interactive Analysis of Jobs	Yes	No
Stream Processing	Yes	Yes, but with external tool (Apache Flink)

Fig. 1. Comparison between Spark and MapReduce

C. Dataset

1) Cervical Cancer Risk Classification: This data set can be accessed on Kaggle [15]. The data set has 858 rows and 36 features. The features included are provided in figure 2 with "Biopsy" being the designated target class.

2) Sipakmed: Sipakmed, introduced in 2018, is one of the first publicly available datasets for labeled Pap smear images [16]. Its development aimed to standardize and increase the availability of Pap smear images. The data set encompasses five distinct classes, annotated based on the cytomorphological characteristics of the cells as follows:

- Normal Cells
  - Superficial-Intermediate Cells
  - Parabasal Cells
- Abnormal Cells
  - Koilocytotic Cells
  - Dyskeratotic Cells
- Begnign Cells
  - Metaplastic Cells

Sipakmed has 966 images, including 4049 cells in total. The specific distribution is shown in Figure 3. For the purpose of this project, only the photo of each individual cell is considered for training and testing purposes.

D. PreProcessing

1) Cervical Cancer Risk Classfication: Before the training, several basic steps were taken to make sure the data were ready for machine learning models. First, all "?" values are replaced with the median of their respective columns. Additionally, all duplicated data was removed, resulting in a reduction in the number of rows from 858 to 835 rows.

The data set was heavily skewed. Specifically, most of the data set was within the age group of 21 to 32 (Figure 4). Other features are also skewed. However, no modification was done to ensure balance on the data set apart from using ADASYN (a super sampling technique) on the "Biopsy" target class. This was necessary due to the imbalanced ratio of 781 to 54

Age	STDs (number)	STDs: HPV
Number of sexual partners	STDs: condylomatosis	STDs: Number of diagnosis
First sexual intercourse	STDs:cervical condylomatosis	STDs: Time since first diagnosis
Num of pregnancies	STDs: vaginal condylomatosis	STDs: Time since last diagnosis
Smokes	STDs: vulvo-perineal condylomatosis	Dx:Cancer
Smokes (years)	STDs: syphilis	Dx:CIN
Smokes (packs/year)	STDs: pelvic inflammatory disease	Dx:HPV
Hormonal Contraceptives	STDs: genital herpes	Dx
Hormonal Contraceptives (years)	STDs: molluscum contagiosum	Hinselmann
IUD	STDs: AIDS	Schiller
IUD (years)	STDs: HIV	Citology
STDs	STDs: Hepatitis B	total_std
total_tests	Biopsy	

Fig. 2. All features in Cervical Cancer Risk Classification dataset

Category	Num of Images	Num of Cells
Superficial/Intermediate	126	813
Parabasal	108	787
Koilocytotic	238	825
Metaplastic	271	793
Dyskeratotic	223	813
Total	966	4049

Fig. 3. Sipakmed's Distribution

between the values of 0 and 1 for the class of 'biopsy', as demonstrated in Figure 5.

2) Sipakmed: For the Sipakmed dataset, the initial step involves compressing the images to alleviate the computational load. In this process, the images are resized to a squared dimension of 77x77 pixels. Although it is common to convert images to grayscale to further mitigate computational demands, it was found that grayscale conversion resulted in a 20% decrease in accuracy during testing. As a result, the

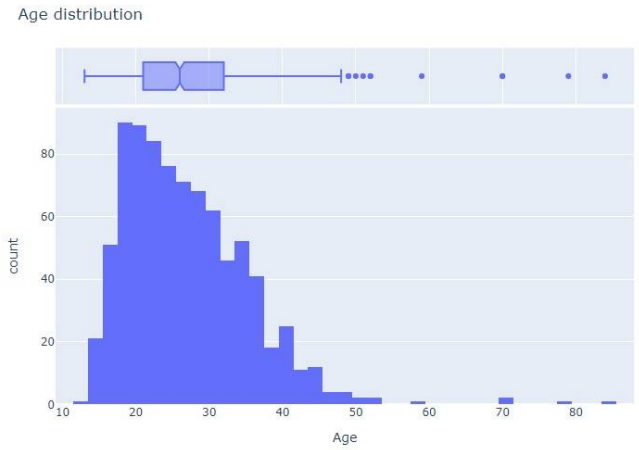

Fig. 4. Age distribution

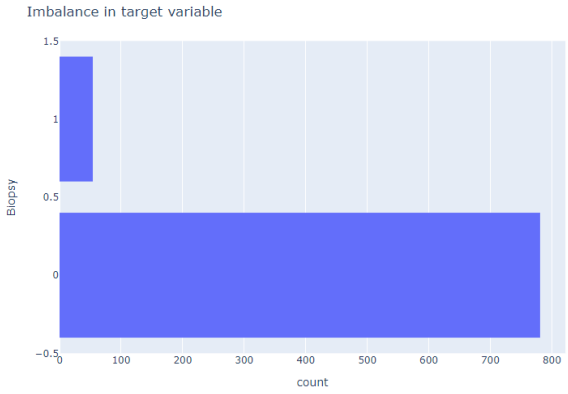

Fig. 5. Imbalance of target variable

decision was made to retain the images' original RGB values without converting to grayscale. Subsequently, the compressed images are transformed into a dataframe structure, where each feature corresponds to the color value of a specific pixel.

Given the considerable number of pixels present in the images, Principal Component Analysis (PCA) is conducted to project the dataset into a lower-dimensional space. SkLearn is employed for these pre-processing stages, a choice that will be rationalized in a subsequent explanation. Following PCA, the resulting dataset is saved as a CSV file and input into Apache Spark for subsequent training and testing phases.

E. PCA

Principal Component Analysis (PCA) is a fundamental technique used to reduce the dimensionality of a dataset while retaining crucial information. In the context of this paper, PCA serves as a vital tool for dimensionality reduction in pap smear cell images. Given an image dataset X comprising n samples, where each sample x_i represents a vector in a high-dimensional space, PCA aims to identify a set of orthogonal vectors (principal components) that capture the maximum variance present in the data.

PC	explained_ratio	explained_ratio_sum
0	5.594232e-01	0.559423
1	1.328089e-01	0.692232
2	4.171268e-02	0.733945
3	3.804913e-02	0.771994
4	3.212373e-02	0.804118
...	...	...
995	9.494518e-07	0.999483
996	9.395253e-07	0.999484
997	9.382255e-07	0.999485
998	9.353575e-07	0.999486
999	9.305075e-07	0.999487

Fig. 6. PCA explained variance

Mathematically, the PCA process involves calculating the covariance matrix [25]

$$Sigma = (1/n) * X^T * X, \tag{1}$$

followed by determining its eigenvectors and eigenvalues. Subsequently, the eigenvectors are ordered according to their associated eigenvalues, signifying the directions of maximum variance within the dataset. Through projection onto these principal components, an alternative, lower-dimensional representation of the original data is generated, which effectively retains the predominant variability encapsulated within the dataset. By integrating PCA into our methodology, we can effectively transform patterns inherent in pap smear cell images into a more manageable and informative feature space, thereby facilitating subsequent machine learning tasks.

In this case, the images dataframe after compression has dimensions of 4049 by 17787. This indicates that the dataframe represents 4049 photos, each with 17787 features. The result from PCA analysis demonstrates that 1000 features are sufficient to explain more than 99% of the original dataframe variance (see Figure 6). This substantial reduction in the number of features significantly alleviates the computational overhead required for the classification task.

F. Model chosen

1) Logistic Regression: Logistic regression is a linear classification algorithm that models the relationship between input variables and a binary output using the logistic function. It is widely used due to its simplicity and interpretability. Logistic Regression performs well when the relationship between features and the target class is linear. It is computationally

efficient and works well with large datasets. However, Logistic Regression may struggle with complex relationships and non-linear data. It assumes linearity and may not capture intricate interactions between variables.

2) Decision Tree: Decision Tree Classifier is a non-parametric algorithm that builds a tree-like model of decisions and their possible consequences. It is intuitive and easy to understand. Decision trees can handle both numerical and categorical data, require minimal data preprocessing, and can capture complex interactions between variables. However, decision trees tend to be prone to overfitting, especially with deep trees. They can be sensitive to small changes in the data, and their interpretability decreases as the tree grows larger.

3) Random Forest: Random Forest Classifier is an ensemble method that combines multiple decision trees to reduce overfitting and improve generalization. It constructs an ensemble of decision trees, where each tree is trained on a random subset of the data and a random subset of the features. Random Forest can handle large datasets, high-dimensional data, and capture complex relationships between variables. However, Random Forest models can be challenging to interpret compared to single decision trees. Training time and memory requirements may increase with a large number of trees in the forest.

4) Gradient Boosted Classifier: Gradient Boosted Classifier is an ensemble method that sequentially builds models, each improving upon the mistakes of the previous ones. It combines weak learners, typically decision trees, in a stage-wise manner. Gradient Boosting can handle complex relationships, works well with heterogeneous data, and can capture feature interactions. It is especially effective in dealing with imbalanced datasets. However, Gradient Boosting can be computationally expensive and may require careful tuning of hyperparameters. It may also be prone to overfitting if the number of boosting iterations is too high.

5) Linear Support Vector Machine: Linear Support Vector Machine (SVM) is a popular classification algorithm that finds the hyperplane that maximally separates the classes in the feature space. SVMs are effective in high-dimensional spaces and can handle both linear and non-linear data by using different kernel functions. They provide robustness against outliers and perform well with small to medium-sized datasets. However, SVMs can be computationally expensive for large datasets and require proper scaling of features. The choice of the kernel function and regularization parameter can significantly impact the performance of SVMs.

6) Naives Bayes: Naive Bayes is a probabilistic classification algorithm that applies Bayes' theorem with the "naive" assumption of feature independence. It is simple, fast, and performs well with high-dimensional datasets. Naive Bayes can handle categorical features and works well with small training sets. It is robust to irrelevant features. However, Naive Bayes assumes feature independence, which may not hold true in real-world scenarios. It may struggle with capturing complex relationships and interactions between variables.

7) Factorization Machine Classifier: Factorization Machine Classifier is a versatile algorithm that models feature interactions in a flexible and scalable manner. It captures complex interactions between features and can handle high-dimensional and sparse data. Factorization Machines are robust to overfitting and require less memory compared to other models. They work well with large-scale datasets and have been successful in various applications, including recommendation systems, click-through rate prediction, and text classification.

G. Experimental Setup and Analysis

The project has two specific goals:

- **Demonstrating Spark's Viability as an Alternative Platform**: One of the key aims is to establish Spark as a compelling alternative platform in comparison to traditional machine learning frameworks such as SKLearn or CNN.
- **Comparing Spark's Scalability**: Additionally, the project strives to conduct a comparison of Spark's scalability in contrast to other non-parallelized platforms, with a particular focus on SKLearn.

To address the first objective, the preprocessed data undergoes shuffling and is then divided into a 70-30 training-testing split. Subsequently, the training data is fed into each of the aforementioned machine learning algorithms. This process is repeated ten times, and the resulting outcomes are averaged over these iterations. The default implementation of each algorithm provided by Spark is employed. The methodology is also replicated for the Sipakmed dataset. To evaluate the model performance, a range of performance metrics including accuracy, precision, F1 score, and recall are computed. These results are subsequently compared with those from SkLearn (for CSV files) and CNN (for images) to provide an overview that supports the claim of Spark's viability as a platform.

In relation to the second objective, given the challenges encountered during image classification in Spark, the assessment of scalability focuses exclusively on the Cervical Cancer Risk Classification dataset. The initial step involves recording the training time for each machine learning algorithm. Subsequently, the algorithm with the longest training time is selected for further analysis. This chosen algorithm is implemented both in SkLearn and Spark. The chosen algorithm is then applied to a CSV file, which is subsequently duplicated in size with each iteration of the loop. Starting at an initial file size of 268KB, the file grows to a substantial 217MB following ten iterations. Within this context, the singular evaluation metric considered is training time, given that the duplicated data points introduce an element of artificiality. These methodological steps collectively contribute to an exhaustive assessment of Spark's potential as a scalable and viable alternative platform for the context of cervical cancer prediction.

TABLE I
SKLEARN'S ACCURACY ON CERVICAL CANCER RISK CLASSIFICATION

Name	Accuracy	Precision	Recall	F1
Logistic Regression	0.93	0.93	0.93	0.93
Random Forest	0.85	0.86	0.85	0.85
K Nearest Neighbour	0.93	0.94	0.93	0.93
Support Vector Classifier	0.97	0.97	0.97	0.97

TABLE II
SPARK'S ACCURACY ON CERVICAL CANCER RISK CLASSIFICATION

Name	Accuracy	Precision	Recall	F1
Decision Tree	0.93	0.98	0.94	0.94
FM Classifier	0.89	0.97	0.92	0.91
GBT Classifier	0.94	0.97	0.97	0.94
Linear SVC	0.95	0.99	0.96	0.96
Logistic Regression	0.93	0.98	0.95	0.94
Naive Bayes	0.82	0.98	0.83	0.86
Random Forest	0.96	0.99	0.97	0.96

IV. RESULTS AND DISCUSSION

A. Accuracy

1) Cervical Cancer Risk Classification: As illustrated in Table II, the Spark-implemented algorithms demonstrate robust performance. Notably, except for Naive Bayes, all other algorithms exhibit strong performance, consistently maintaining accuracy rates above 90

In Table I, the accuracy results when implemented using Scikit-learn are illustrated. While the accuracy rates exhibit a comparable range to those observed in Spark's implementation, the Support Vector Classifier stands out with the best results, achieving accuracy scores exceeding 96% for all metrics. However, Random Forest performs comparatively less favorably, with accuracy scores hovering around 85% across all metrics.

For Spark's Implementation, the Random Forest algorithm emerges as a promising solution for the given problem. This is an interesting observation, as the Scikit-learn implementation of the same algorithm yields a much lower result. The difference most likely comes from default parameters of each platform, as we used the default parameters for both platforms. As shown in Figure 7, this algorithm boasts an average training time of 1.18 seconds, ranking as the third most efficient. Remarkably, despite its efficient training, the Random Forest algorithm maintains consistently high performance levels (above 96%) across all other metrics, further affirming its suitability for the task at hand.

2) Sipakmed Dataset: In Table. III, the results for the SipakMed dataset regarding Spark accuracy are presented. Overall, the outcomes are suboptimal, with no model surpassing the 80% mark for any metrics. The most favorable performance is observed in the Decision Tree classifier, achieving scores above 72% for all metrics. On the other end of the spectrum, Random Forest performs the least effectively, with a recall score of 57%, although other metrics achieve scores above 65%. While this performance surpasses random guessing, considering the classification task's complexity across five classes, the results remain far from desirable.

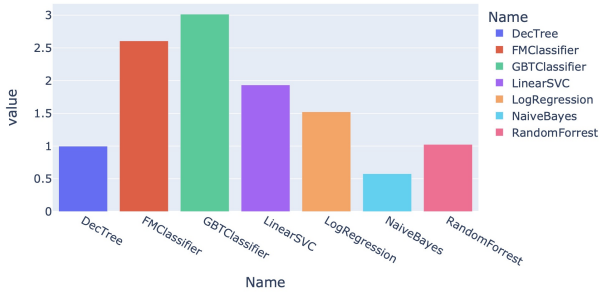

Fig. 7. Spark's Training Time (Cerivical Cancer Classification)

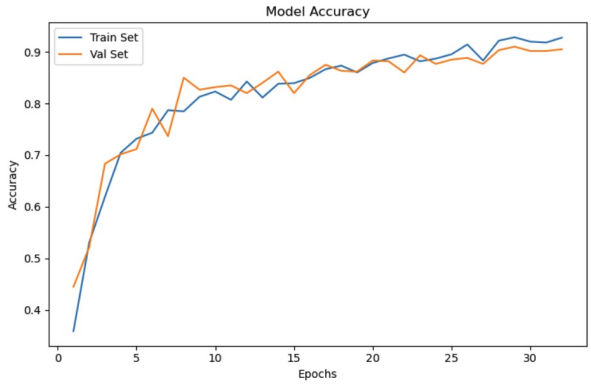

Fig. 8. CNN's Accuracy (SipakMed)

Historically, Convolutional Neural Networks (CNNs) have excelled in image classification tasks [26]. Figure 8 showcases the significant superiority of CNN performance compared to the Spark implementation in this paper as in Figure 9. Despite being designed with 32 epochs in mind, CNN outperforms all of Spark's implemented algorithms as early as epoch 10. By the 32nd epoch, CNN attains an accuracy of 92.75% on the same image dataset.

This result highlights the fact that image classification is not an area where Spark demonstrates strength, especially in comparison to specialized deep learning frameworks like TensorFlow or PyTorch that leverage GPU acceleration. While Spark ML serves as a versatile tool for distributed computing tasks, its architecture and default configurations are not optimized for the intricacies of image processing and deep learning.

B. Scalability

Figure 7 illustrates the training times of various algorithms, guiding the selection of the Gradient Boosted Tree algorithm for scalability testing due to its longer training duration.

As shown in the Fig. 10, SkLearn's fitting time doubles with each iteration, reflecting its lack of parallelization. This behavior aligns with expectations, considering the proportional

TABLE III
SPARK'S ACCURACY (SIPAKMED)

Name	Accuracy	Precision	Recall	F1
Decision Tree	0.73	0.73	0.77	0.73
Logistic Regression	0.66	0.63	0.67	0.66
Random Forest	0.65	0.70	0.57	0.64

Fig. 9. Spark's Training time (SipakMed)

increase in training time with growing file size. In contrast, Spark exhibits a distinct pattern. While the initial iteration involves additional setup time for parallelization, subsequent iterations demonstrate a significant performance advantage over SkLearn. Notably, Spark maintains consistent and comparatively faster training times from the second to the fourth iteration, hovering around 1.6 seconds.

As shown in the Table IV, Spark's training time, after the initial setup in the first iteration, remains relatively steady across iterations 2, 3, and 4. Remarkably, starting from the seventh iteration, Spark exhibits a substantial performance improvement, achieving considerably faster training times.

While SkLearn showcases a relatively faster training time in percentage terms before the sixth iteration, the practical difference between instant completion and a 2-second interval remains minimal. However, as the iterations progress, Spark's lead becomes evident, resulting in a 16-second difference in training time by the tenth iteration. This means, the scalability

TABLE IV
SCABILITY OF SKLEARN AND SPARK

Size	SkLearn (s)	Spark (s)
1	0.110263	5.197456
2	0.126359	1.664117
3	0.217357	1.639954
4	0.378548	1.661266
5	0.700354	1.991977
6	1.388288	2.140520
7	2.694753	2.308474
8	5.468835	3.224648
9	11.219578	5.624002
10	23.706132	7.079738

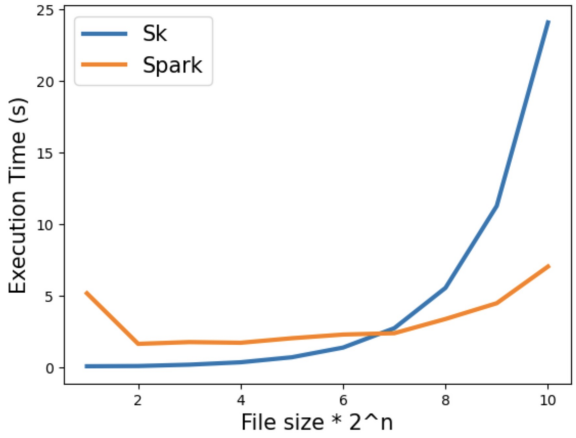

Fig. 10. Scalability (Chart)

of spark grows three times than the SkLearn in the 10th iterations, reducing the time complexity. This emphasizes Spark's superiority in terms of scalability.

These findings show that Spark has significant advantage over SkLearn when managing larger datasets and achieving efficient parallel processing. Spark's capability to deliver consistent and fast performance, even as data size increases, solidifies its position as the preferred platform for scalable text based machine learning tasks.

V. SPARK'S IMPLEMENTATION ISSUES

Despite Spark's demonstrated scalability with larger text data and satisfactory performance with smaller datasets, certain issues arise in its implementation:

- **Performance Discrepancies Between Platforms**: In this research scope, significant performance variations were observed between Spark instances running on different hardware setups. For example, Spark exhibited notably faster execution on an M1 chip compared to a Windows machine equipped with an i7-12700 processor. While the training of the smallest model on the Windows machine took at least 50 seconds, the M1 chip completed the same task, including setup time, in just 5.3 seconds.
- **Popularity and Algorithm Availability**: Unlike Scikit-learn, Spark has a smaller user base, resulting in fewer available algorithms. While the SparkML library offers a diverse range of algorithms, it may not provide as extensive a selection as Scikit-learn. Users may find a wider range of readily accessible algorithms within the Scikit-learn ecosystem.

Furthermore, while not directly related to Spark as a library, developing custom algorithms within the Spark framework can be significantly more complex compared to Scikit-learn. To fully leverage Spark's capabilities, users need to understand additional concepts such as MapReduce and Resilient Distributed Datasets (RDDs). This steep learning curve may require users to invest more time and effort to extract the

maximum benefits from Spark. We encountered this first-hand when attempting image classification tasks, where the traditional workflow involving image compression, grayscale transformation, and subsequent PCA proved challenging. Despite having 32GB of RAM, loading images and applying PCA resulted in memory overflow. Upgrading to a system with 128GB of RAM allowed PCA processing to begin, but after approximately 15 minutes, Spark's PCA implementation unexpectedly reduced the dataset's dimensionality, contrary to PCA's intended function. This issue stemmed from both code implementations and potential limitations in Spark's PCA implementation. However, we addressed this issue by using PCA from Scikit-learn, saving the output as a CSV, and then feeding it into Spark. Nonetheless, the inability to perform all tasks solely within the Spark ecosystem remains a constraint.2

Overall, Spark offers significant scalability benefits and a comprehensive machine learning library suitable for diverse computational environments, including those primarily utilizing CPUs. Nevertheless, it is imperative to acknowledge potential performance discrepancies based on hardware configurations, algorithm availability with respect to SkLearn, and the additional expertise required to effectively wield Spark. Furthermore, Spark may not be as well-suited for certain machine learning tasks, such as image classification, which can benefit from specialized hardware like GPUs."

VI. Conclusion

Our research has revealed that Spark is a viable option for the medical sector, particularly in the area of cervical cancer prediction. All algorithms tested for CSV file showed excellent results, with most achieving an accuracy of 98% or higher. Furthermore, Spark was able to efficiently manage large datasets, which is essential given the growing availability of data from IoT devices that can be used to improve machine learning model performance. It is important to note that Spark is not a universal solution for all machine learning problems. There are many difficulties encounter when trying to make image recognition works on Spark. But in the context of big data, the field of machine learning should be moving towards increased parallelization. Spark's capabilities are well-suited to this trend, making it a valuable tool for taking advantage of the potential of big data in medical applications and beyond. A key aspect of the research is its multimodal approach to cervical cancer detection, analyzing structured data in CSV format and unstructured image data. By addressing the detection challenge from multiple angles, the study underscores the potential of Spark in processing various data types for healthcare applications, paving the way for more comprehensive and effective diagnostic solutions. In future works, we plan to explore about image classification in general settings using Spark implementations for other modalities, that might help in improvements in scalability issues.

References

[1] World Health Organization. World health organization. https://www.who.int/health-topics/cervical-cancer#tab=tab_1. Accessed on: 27th March 2024.

[2] Manika Jha, Richa Gupta, and Rajiv Saxena. Cervical cancer risk prediction using xgboost classifier. In *2021 7th International Conference on Signal Processing and Communication (ICSC)*, pages 133–136, 2021.

[3] Kyaw Phyo Win, Yada Kitjaidure, Kohei Hamamoto, and Thet Myo Aung. Computer-assisted screening for cervical cancer using digital image processing of pap smear images. *Applied Sciences*, 10(5):1800, 2020.

[4] Keshav Allawadi, Mayank Kumar Singh, and Charvi Vij. Using machine learning to improve healthcare: A disease prediction and management system. In *2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, pages 281–285, 2023.

[5] Dongmin Lee and Sung-Nyun Yoon. Application of artificial intelligence-based technologies in the healthcare industry: Opportunities and challenges. *International Journal of Environmental Research and Public Health*, 18(1):271, 2021.

[6] Juan M Górriz, Javier Ramírez, Andrés Ortíz, Francisco J Martínez-Murcia, Fermín Segovia, John Suckling, Michael Leming, Yong-Dong Zhang, José R Álvarez-Sánchez, Guido Bologna, Pierluigi Bonomini, Francisco E Casado, Daniel Charte, Francisco Charte, Ricardo Contreras, Alfredo Cuesta-Infante, Rodolfo J Duro, Antonio Fernández-Caballero, Eduardo Fernández-Jover, Pedro Gómez-Vilda, Manuel Graña, Francisco Herrera, Roberto Iglesias, Anna Lekova, J de Lope, Eloy López-Rubio, Raúl Martínez-Tomás, Miguel A Molina-Cabello, Andrés S Montemayor, Paulo Novais, Daniel Palacios-Alonso, Juan J Pantrigo, Brian R Payne, Francisco de la Peña López, Martin A Pinninghoff, María Rincón, Juan Santos, Karl Thurnhofer-Hemsi, Athanasios Tsanas, Roberto Varela, and José M Ferrández. Artificial intelligence within the interplay between natural and artificial computation: Advances in data science, trends and applications. *Neurocomputing*, 410:237–270, 2020.

[7] Hormuzd A Katki, Sholom Wacholder, Diane Solomon, Philip E Castle, and Mark Schiffman. Risk estimation for the next generation of prevention programmes for cervical cancer. *The Lancet Oncology*, 10(11):1022–1023, 2009.

[8] Sanjay J P, Tummalapalli Naga Deepak, and Manimozhi M. Prediction of health problems and recommendation system using machine learning and iot. *2021 Innovations in Power and Advanced Computing Technologies (i-PACT), Innovations in Power and Advanced Computing Technologies (i-PACT), 2021*, pages 1 – 8, 2021.

[9] Yue Ming, Xiying Dong, Jihuai Zhao, Zefu Chen, Hao Wang, and Nan Wu. Deep learning-based multimodal image analysis for cervical cancer detection. *Methods*, 205:46–52, 2022.

[10] Tao Xu, Han Zhang, Xiaolei Huang, Shaoting Zhang, and Dimitris N Metaxas. Multimodal deep learning for cervical dysplasia diagnosis. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*, pages 115–123. Springer, 2016.

[11] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

[12] Jiblal Upadhya, Jorge Vargas, Khem Poudel, and Jaishree Ranganathan. Improving the efficiency of multimodal approach for chest x-ray. In *International Conference on Advances in Computing Research*, pages 47–59. Springer, 2024.

[13] Ciprian Dobre and Fatos Xhafa. Parallel programming paradigms and frameworks in big data era. *International Journal of Parallel Programming*, 42(5):710–738, 2014.

[14] Satish Gopalani and Rohan Arora. Comparing apache spark and map reduce with performance analysis using k-means. *International journal of computer applications*, 113(1), 2015.

[15] Gokagglers. Cervical cancer risk classification, Aug 2017.

[16] Marina E. Plissiti, P. Dimitrakopoulos, G. Sfikas, Christophoros Nikou, O. Krikoni, and A. Charchanti. Sipakmed: A new dataset for feature and image based classification of normal and pathological cervical cells in pap smear images. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3144–3148, 2018.

[17] Somya Srivastav, Kalpna Guleria, and Shagun Sharma. Predictive machine learning approaches for cervical cancer detection: An analytical comparison. In *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 951–956, 2023.

[18] Priyanshu Rawat, Madhvan Bajaj, Shreshtha Mehta, Vikrant Sharma, and Satvik Vats. A study on cervical cancer prediction using various machine learning approaches. In *2023 International Conference on In-*

novative Data Communication Technologies and Application (ICIDCA), pages 1101–1107, 2023.

[19] Mahmudul Hasan, Priyanka Roy, and Adiba Mahjabin Nitu. Cervical cancer classification using machine learning with feature importance and model explainability. In *2022 4th International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, pages 1–4, 2022.

[20] Madhusree Kuanr, Puspanjali Mohapatra, and Jayshree Piri. Health recommender system for cervical cancer prognosis in women. *2021 6th International Conference on Inventive Computation Technologies (ICICT), Inventive Computation Technologies (ICICT), 2021 6th International Conference on*, pages 673 – 679, 2021.

[21] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113 – 113, 2008.

[22] Howard Karloff, Siddharth Suri, and Sergei Vassilvitskii. A model of computation for mapreduce. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 938–948. Society for Industrial and Applied Mathematics, 2010.

[23] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. In *HotCloud'10: Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, pages 10–10, 2010.

[24] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, Doris Xin, Reynold Xin, Michael J. Franklin, Reza Zadeh, Matei Zaharia, and Ameet Talwalkar. Mllib: Machine learning in apache spark. 2015.

[25] S. Mishra, U. Sarkar, S. Taraphder, S. Datta, D. Swain, R. Saikhom, and et al. Multivariate statistical data analysis- principal component analysis (pca). *International Journal of Livestock Research*, 7(5):60–78, 2017.

[26] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. 2013.