**Journal of Engg. Research EMSME Special Issue pp. 17-25** DOI : 10.36909/jer.EMSME.13897

# A PSO Based Cloud Framework for Knowledge Extraction

Chaitanya Kanchibhotlaa*, Pruthvi Raj Venkatesh**, DVLN Somayajulu*, Radhakrishna P*

** National Institute of Technology, Warangal, India*
*** Ambedkar Institute of Technology, Bangalore, India*
*Corresponding Author : ckanchibhotla@gmail.com*

## ABSTRACT

Many industries, such as oil, construction, banking, and insurance, have substantial historical physical data. Companies store this data in physical warehouses that are geographically distributed and usually taken care of by record management companies. Storing large volumes of historical physical data poses many critical challenges, such as increased maintenance cost, high time for recovery, and unsearchable data. Many companies digitize this data and consolidate this data into cloud repositories as part of their Digital Transformation (DT) journey to address these challenges. This DT process introduces many other technical challenges while dealing with poor scans, huge file size, geographically distributed files, and confidential documents. Though there are options to resolve each of these limitations individually, there are no frameworks that deal with digitization and historical data storage in its entirety. Moreover, they cannot handle a large number of documents having variable file sizes. This paper presents a generic cloud-based high-performance computing framework for knowledge extraction, comprising document classification based on neural networks and particle swarm optimization (PSO), data extraction, metadata enrichment, image enhancement using image processing (IP) techniques, and high data availability to users using cloud-based search. The proposed framework is executed on two cloud providers, i.e., Azure and AWS, to test its efficacy.

**Keywords:**Neural networks; Classification. azure; A.W.S, Digitization; Digital Transformation; Particle swarm optimization

## INTRODUCTION

Many industries, such as oil, construction, banking, and insurance, have a substantial number of complex workflows that generate a considerable amount of data every day. This data can be in both physical and digital formats. Physical data includes paper documents, signed agreements, CDs, tape drives, floppy disks. Digital data will be in the form of structured databases (such as SQL, Oracle, NoSQL) and unstructured data (word documents, PDFs, Excel). This data is of great importance for the industry as it holds valuable information such as lessons learned, company proceedings, assets, business process, etc. As this data has grown over the years, companies are finding it challenging to manage vast storage of data as it poses many challenges, such as 1) Cost- for storing the data on-premises world, 2) High retrieval time- for finding the content of interest in the accumulated data corpus, 3) Data loss- as data stored in data warehouses also deteriorate over time, leading to a loss of critical data 4) Lost awareness- People who generate critical data may no longer be part of the organization in which the data is generated. Hence, there is no awareness of the existence of critical data. 5) Lost Knowledge- As the data is abundant and scattered, the knowledge is lost forever.

Many companies have embraced the digital transformation journey of moving data to the cloud by digitizing physical data and moving all the digital data to the cloud to reap the benefits such as decrease in physical storage costs, time data availability, disaster recovery, and information policies, etc. Digitization and extraction of domain-related metadata for data in the cloud will help users search for content of interest.Users can refer to the data that is extracted from these digitized documents for making informed decisions. The standard formats in which the data is digitized are TIFF for files (such as well logs) and PDF (for well reports). Even though the content is digitized and moved to the cloud, there are many problems such as huge File size, Searchability, and multiple versions. (Wu et

al.,2015) introduced a framework called PDFME which can extract multiple entities from PDF documents using PDFbox and TET. (Singh et al.,2016) developed a framework called OCR++ for metadata extraction from scholarly articles, identifying common patterns in writing and frame rules for information extraction.

This paper presents a practical cloud-based framework with implementation details on the two most popular cloud providers Amazon A.W.S. and Microsoft Azure. The proposed solution is a multi-step process in which files are uploaded to the cloud, classified, indexed, and presented through a search portal. The indexing process involves running complex metadata extraction modules like key metadata extraction, image processing, OCR, and thumbnail generation. Only images and PDF documents are considered in the current process.

The rest of the paper is ordered as follows. Section II details the concepts used in this paper, such as data classification, cloud service, image processing, etc. In section III, we introduce the proposed method. Section IV presents the proposed method performance along with the results, and Section V concludes the paper.

## BASIC CONCEPTS

This section introduces and briefly explains all the basic concepts used in this paper.

### Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is based on popular optimization techniques belonging to swarm intelligence (SI). It belongs to evolutionary algorithm in which the particles reiterate in solution space in search of the best solution. Each particle in the swarm iterates in the search space with both position and velocity. At each iteration, the velocity and position of each particle is updated using the following equation:

$$Vij^{t+1} = Vij^{t} + c_1 r_1 j^{t} * [pBest\ i^{t} - xij^{t}] + c_2 r_2 j^{t} * [gBest\ i^{t} - x_{ij}^{t}] \quad (1)$$

$$x_i = x_i + v_i \quad (2)$$

where $v_{ij}^{t}$ and , $x_{ij}^{t}$ are the velocity and position of particle i in dimension j at time t,pBest $i^{t}$ is the personal best position of particle i in dimension j found till time t, gBest $i^{t}$ is the global best position of particle i in dimension j found till time t, Constant c1, c2 are the acceleration coefficients which determines the influence of particle's personal and social experiences, $r_{1j}^{t}$ and $r_2^{t}$ are random numbers. The three terms $v_{ij}^{t}$, $c_1 r_{1j}^{t}$*[pBest $i^{t}$- $x_{ij}^{t}$ ] and $c_2 r_{2j}^{t}$ *[gBest $i^{t}$ - $x_{ij}^{t}$ ] represent inertial, cognitive, and social components, respectively, which significantly impact the particle's velocity.

### Data Classification

Data classification is one of the common activities in machine learning. It involves predicting the class of an item to which it can belong. Some of the important uses of classification are document classification, image classification, and handwriting recognition. In industries such as oil, construction, banking, and insurance, the number of physical documents is humungous. They need to be classified accordingly for further processing. Neural networks (NN) are one of the important algorithms used for data classification. NN belongs to the category of supervised learning. In supervised learning, the neural network is first trained with already known class labels. Predictions are made on the test data using the trained model. Due to its popularity, neural networks are used in image classification (Yim et.al. 2009), image processing and recognition (Browne et.al. 2008), object detection (Jiang et.al. 2019), machine translation (Han and Li 2020) etc. Training a neural network for better performance is an optimization problem (Smith, 2017).

'Weight' and 'learning rate' are important hyperparameters in neural networks that play a crucial role in maintaining the classification process's efficiency, transforming the input data to match the output. There are many optimization methods for setting a neural network's learning rate, such as gradient descent, stochastic gradient descent, mini-batch gradient descent, and so on. Each of these optimizers has its advantages and disadvantages. Among these, stochastic gradient descent is the most widely used optimization method. In general, if the learning rate is low, neural network training is more reliable. However, the optimization process takes more time because of

the small steps. If the learning rate is high, then the optimization process may not converge to its optimal solution because of the big steps. Hence, finding an optimal learning rate for the dataset is challenging.The difference between the actual value and the predicted value of a neural network is called the "loss" or "error" of a neural network. There are several ways to optimize the error of a neural network. Backpropagation with gradient descent is the most widely used algorithm to minimize the loss and train the neural network, which has many advantages and disadvantages. The major disadvantage is that it is extremely sensitive to the weights of neural networks and learning rate. Making a small change to these values impacts the performance of the neural network.

**Some Methods of Finding the Learning Rate**

There are multiple ways of finding the learning rate of a neural network. The primary approach is to set a random learning rate value and exponentially decrease it by calculating the network's loss with each value.

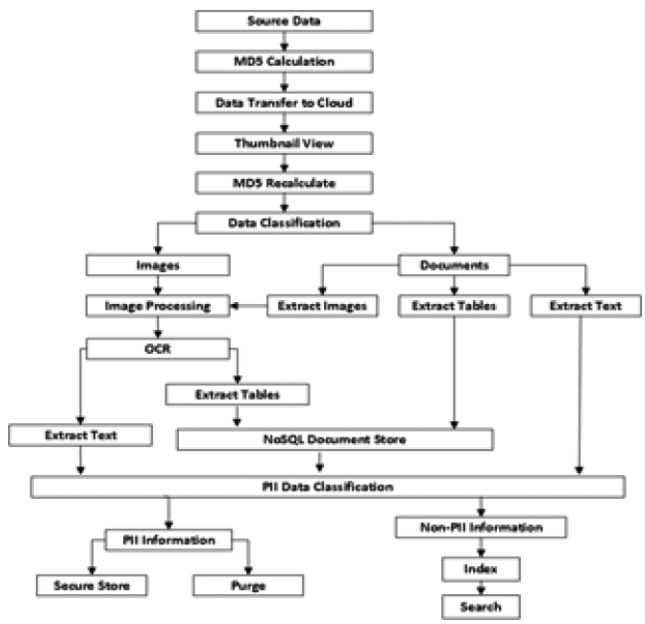



**Figure 1.** Process Overview

The value for which the loss is minimal can be considered as the learning rate of the network. For instance, initially, the learning rate value of 0.1 can be considered, and later, lower values such as 0.01 and 0.001 can be considered. Multiple optimization algorithms can be used to change the learning rate of the network. Some of the important algorithms are gradient descent (Ruder, 2016) stochastic gradient descent (Ruder, 2016) , mini-batch gradient descent (Ruder, 2016) Gradient descent is the most widely used algorithm typically used in classification and regression problems. The loss is transferred from one layer to another in gradient descent. The weights are modified according to the learning loss value. Stochastic gradient descent is an optimization algorithm that calculates the error gradient for each record in the training set and updates the model weights using the backpropagation of errors algorithm (Ruder, 2016). Figure 1 shows the process overview diagram describing the sequence of steps in the framework

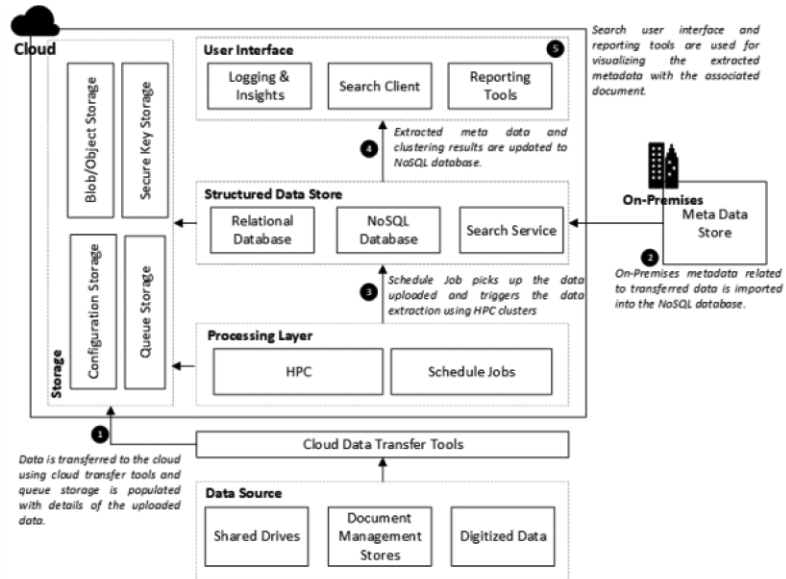



**Figure 2.** Solution Overview

## PROPOSED APPROACH

This section presents our approach to digital media knowledge extraction, which includes the novel data classification approach.

### Solution Overview

Figure 2 presents the technical components and the flow of events in the overall solution. The diagram is numbered to display the flow of the events in the framework

### Proposed Method of Finding Learning Rate

Motivation behind the proposed learning rate:

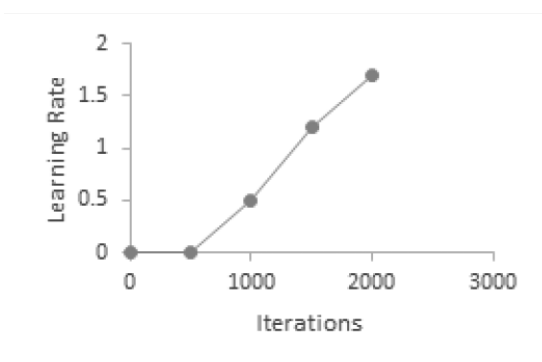



**Figure 3.** Learning rate for iterations

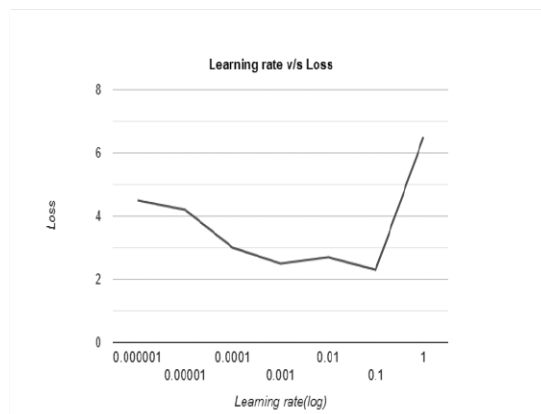



**Figure 4.** Learning rate versus loss values

In stochastic gradient descent, the parameters, i.e., weights (θ), are updated by equation

$$\theta t = \theta t - 1 - \varepsilon t \frac{\delta L}{\delta \theta} \quad (1)$$

**Table 1.** Cloud Service – Solution Component Mapping

| No | Cloud Service | Azure Offering | AWS Offering |
|---|---|---|---|
| 1. | Configuration Storage | Azure Table Storage, Azure Cosmos Database | Amazon DynamoDB |
| 2. | Blob Storage | Azure Blob Storage | Amazon Simple Storage Service(Amazon S3) |
| 3. | Logging & Insights | Application Insights | Amazon Cloud Watch |
| 4. | HPC Batch Service | Azure Batch Service | AWS Batch |
| 5. | Graph Database | Azure Cosmos Database | Amazon Neptune |
| 6. | NoSQL Database | Azure Cosmos Database | Amazon DynamoDB |
| 7. | Search Service | Azure Cognitive Search | Amazon CloudSearch |
| 8. | Secret store | Azure KeyVault | AWS Secrets Manager |

Where 'L' is the loss function, and εt is the learning rate. If L is too small, the algorithm converges very slowly. If L is too high, the training process converges very quickly near the sub-optimal solution. Hence there is a need to need to find out and experiment with various learning rates and schedules. The basic idea is to calculate the learning rate and training loss at each predefined step of neural network execution. Figure 3 shows the plot of values of both values. The x-axis shows the learning rate, and the y-axis shows the training loss. Choosing a point that has the minimum rate of change is reasonable(it is 0.01 from the figure) because, at this point, the objective function must have reached its local minima during the optimization process.

Our proposed method for finding the learning rate is two-phased. The entire process is outlined in algorithm, Figure 5. In the first phase, the neural network is initialized with the parameters such as batch size, minimum learning rate, maximum learning rate. batch_step_count is the number of steps for which the learning rate should be captured. The minimum learning rate is the starting value of the learning rate with which the neural network is initialized. The maximum learning rate is the value of the learning rate at which the program execution can be halted. Typically, the minimum learning rate and maximum learning rate values provide the range of learning rate values between which the values are captured. The input dataset is divided into test and train sets. The neural network is initialized with the parameter values. In the training phase, the neural network is optimized with the stochastic gradient descent method. We record the learning rate when the condition for batch_step_count is satisfied. We continue the same process for all epochs. In the second phase, the PSO algorithm is executed with predefined parameters. Until the stopping criteria are met, we execute the PSO algorithm with the objective function presented in equation (2). The objective function calculates the rate of change of loss.

Table 1 below captures the mapping between the solution components detailed in the solution overview section (refer to section 2.7) and the corresponding cloud service components in Azure and AWS.

**Algorithm 1:Algorithm to find the learning rate**

**Inputs**: Neural network parameters and PSO parameters.

**Outputs**: learning rate value

*%%Phase_1%%*

1) Initialize parameters for the neural network: No_of_epochs, batch_step_count, Minimum learning rate, maximum learning rate, and parameters for PSO: number of particles, number of iterations
2) Initialize neural network with the parameters.
3) Read dataset
4) Split the data into training data and test data.
5) Start the training phase for the neural network with training data.
6) Initialize optimizer with stochastic gradient descent method with learning rate.
7) If current_epoch% batch_step_count ==0:
    a. Record learning rate
    b. Record training loss

*%%Phase_2%%*

8) Initialize PSO with predefined parameters

**Figure 5.** Alorithm

## RESULTS AND DISCUSSIONS

### Data Classification Results

This section implements the newly proposed process on the imagenette dataset to find the optimal learning rate and check its validation accuracy. The imagenette dataset is a collection of 10 classes and is a minified version of the imagenet dataset. ImageNet dataset contains more than 14 million images belonging to 20,000 categories. It required a considerable CPU, GPU power to execute and is mainly used in compute vision tasks such as object identification, image classification, etc. FastAI developed the Imagenette dataset to execute and test any new algorithms without computing resources as required by ImageNet. The dataset contains 9,469 records in the training set and 3,925 records in the validation set.

For the sake of experimentation, No_of_epochs is taken as 2000, and the batch_step_count is taken as 100. The minimum learning rate is taken as 0.001, and the maximum learning rate is taken as 0.005. Figure 4 shows the value of the learning rate captured across all the iterations. Figure 5 shows the learning rate and training loss for all the iterations, and figure 6 shows the rate of change of loss and the learning rate. From the figures, it can be found that the optimal learning rate for the network is 0.01. Next, we define the neural network architecture. The number of input nodes considered for the experiment is 4, the number of hidden nodes is 20, and the output nodes are 3. One hot encoding is carried out to calculate the categorical cross-entropy loss. A new binary value is added for each unique integer value in one-hot encoding, and a unique integer vector is assigned to each class. Categorical loss entropy is used as the loss function for the input. The softmax function is used to calculate the probability of each class from logits. In the forward pass of the neura l network, we calculate the error between actual and predicted labels. We then return the error to PSO to optimize the error. Figure 7 shows the accuracy plot for the model. It can be observed that the model has performed better in terms of training, testing, and validation accuracies. All the scores have reached 100 % accuracy at the later stages, which shows that the model has performed better on the provided training data. Also, from figure 7, it can be seen that the model has comparable performance on the compared dataset. For PSO, the number of particles is taken as 100, and inertia weight is taken as 0.9, the constants c1 and c2 are taken as 0.5, 0.3. The number of dimensions is taken as 163, which is according to the below formula

*(input_nodes * hidden_nodes) + hidden_nodes + (hidden_nodes * output_nodes) + output_nodes*

The same procedure is applied for the source dataset downloaded from the Bureau of Safety and Environmental Enforcement which belongs to the oil and gas domain. The total number of images in the dataset is 2943, and the size of the dataset is 1.4 GB. The dataset files are all PDF files, and pages are extracted as images from the dataset

using the python library. The dataset consists of files related to six classes: well logs, well reports, seismic sections, seismic reports, general reports, and maps. They all belong to the Gulf of Mexico(GOM) region.

**Results of Image Enhancement Techniques**

Figure 8 shows the image processing results for a sample image from the source dataset.

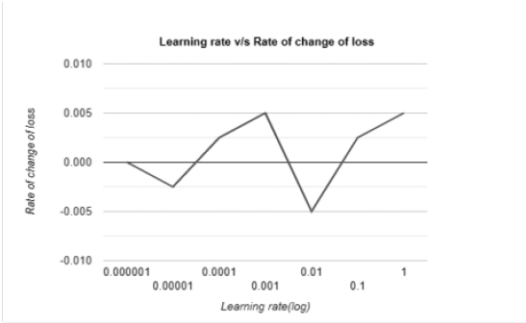



**Figure 6.** Rate Of Change of Loss

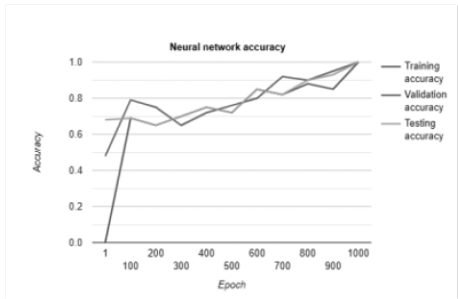
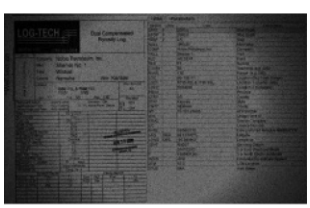



**Figure 7.** Neural Network Accurancy

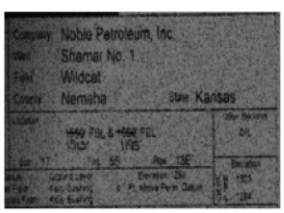
The input image is presented in figure 8(b). Figure 8(b) presents the subsection of the input image. 8(c) shows the sharpened image, and 8(d) shows the adaptive thresholding phase's output. Finally, figure 8(e) shows the smoothened image, and figure 8(f) shows the complete final resultant image after all the image processing steps.

**Performance Metrics**

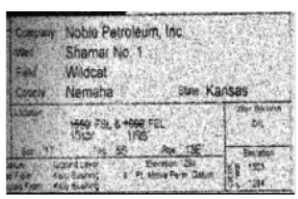
Table 2 shows the average execution time in minutes for all the steps in the process. This time is calculated in both Azure and AWS.



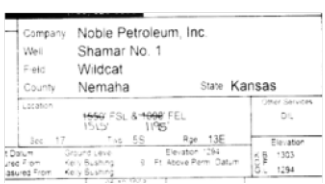
(a) Input Image



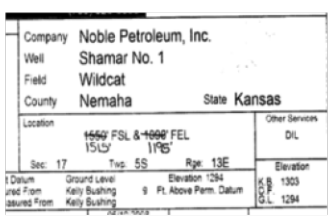

(b) Subsection of Image




(c) Image Sharpening




(d) Adaptive Thresholding




(e) Image Smoothening

(f) Output Image

**Figure 8.** Output of image processing techniques

It can be noted that the execution time is similar across Azure and AWS. The experiment was carried out with 30 files consisting of 15 image files and 15 PDF files. The average size of the files is 12.45 MB. Data transfer is done through a computer with an average data upload and download speed of 10 Mbps. Azure service components are set up in the North Europe data center, and AWS service components are set up in the Ireland data center. It can be noted that the majority of the time is consumed during data transfer to the cloud and search indexing.

**Keyword Extraction Metrics**

We used FlashText library to extract keywords such as country names, well names, basin names, etc. Figure 9, 10 shows the results captured as part of the experiment performed on the Standard_A4m_v2 machine. The Python flash text module was configured to extract country names and basin names from text documents with a variable number of terms. A Dictionary consisting of country names and basin names was provided as input to the flash text module. The results indicate that the flash text performance is significantly better than the traditional regular expression-based search.

**Batch Processing Metrics**

This section presents the execution results of batch service in both Azure and AWS cloud providers. The experiment involved the execution of series of python modules that performed MD5 calculation, data classification, image processing, data extraction from image and PDF, table data extraction, and populating to the data store. The batch service initially consisted of five virtual machines, which are incremented by additional five virtual machines in subsequent iterations. Figure 11 shows the total execution time in both Azure and AWS. Y-axis shows the total execution time, and X-Axis shows the total number of servers that are added in each iteration. Experiments indicate that the total execution time is consistent across the batch service in AWS and Azure.

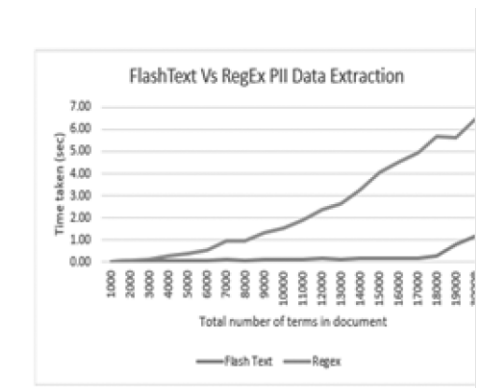



**Figure 9.** PII Data extraction times

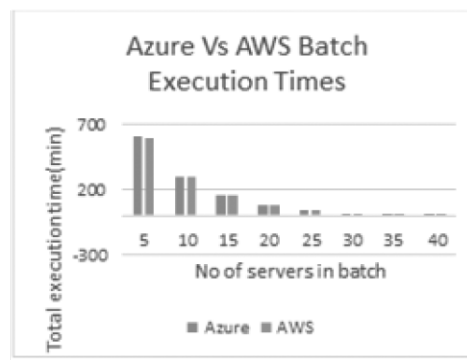



**Figure 10.** PII Data extraction times

**Overall Cost**

This subsection shows the overall cost in both Azure and AWS across all the proposed knowledge extraction framework phases. The cost has been calculated for hosting 2000 documents downloaded from the BSEE website. The hosting cost is calculated for a month usage of 1 TB blob storage and one time execution of the HPC cluster. It can be observed that AWS costs are relatively cheaper when the base configurations are considered for the setup

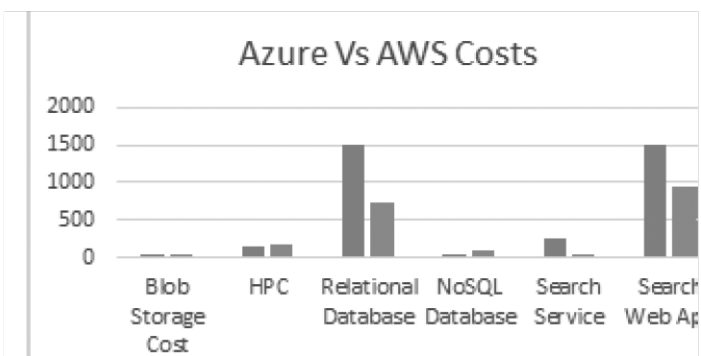



**Figure 11.** Azure Vs AWS Cost Comparison

## CONCLUSION

Digitization is believed to bring cost savings to a company. The real value is achieved if the knowledge out of digitized data is extracted and available. The paper presents a generic knowledge extraction framework to bring cost savings and faster turnaround to business. The framework proposed in the paper provides an innovative solution for a complex problem that arises out of digitization. The services used in both Azure and AWS are listed in the paper. This paper also proposed a novel way of classifying the data using the concepts of PSO. Experiments were carried out on the dataset provided by BSEE consisting of 2943 PDF documents consisting of images. It is observed that the performance is almost the same in both Azure and AWS environments. The framework presented in this paper can be extended to other domains and extended to other cloud environments based on the services' availability.

## REFERENCES

**Yim J., Ju J., Jung H., Kim J. 2015** Image Classification Using Convolutional Neural Networks With Multi-stage Feature. In: Kim JH., Yang W., Jo J., Sincak P., Myung H. (eds) Robot Intelligence Technology and Applications 3. Advances in Intelligent Systems and Computing, vol 345. Springer, Cham. https://doi.org/10.1007/978-3-319-16841-8_52

**Browne M., Ghidary S.S., Mayer N.M. 2008** Convolutional Neural Networks for Image Processing with Applications in Mobile Robotics. In: Prasad B., Prasanna SRM (eds) Speech, Audio, Image and Biomedical Signal Processing using Neural Networks. Studies in Computational Intelligence, vol 83. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-75398-8_15

**Xiaoyue Jiang,Abdenour Hadid,Yanwei Pang,Eric Granger, Xiaoyi Feng 2019** Deep Learning in Object Detection and Recognition: Springer Nature Singapore Pte Ltd.

**Han Z., Li S. 2020** Research on Machine Translation Model Based on Neural Network. In: Liang Q., Liu X., Na Z., Wang W., Mu J., Zhang B. (eds) Communications, Signal Processing, and Systems. CSPS 2018. Lecture Notes in Electrical Engineering, vol 517. Springer, Singapore. https://doi.org/10.1007/978-981-13-6508-9_31

**L. N. Smith, 2017,** Cyclical Learning Rates for Training Neural Networks, 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, 464-472, doi: 10.1109/WACV.2017.58.

**Ruder S. 2016** An overview of gradient descent optimization algorithms. arXiv preprint arXiv:160904747.

**Cloud Data Migration on AWS |** Amazon Web Services. https://aws.amazon.com/cloud-data-migration/

**Choose an Azure solution for data transfer |** Microsoft Docs. https://docs.microsoft.com/en-us/azure/storage/common/storage-choose-data-transfer-solution

**C. Kanchibhotla, P. Venkatesh, D. Somayajulu and P. R. krishna, 2019** An Efficient Cloud-Based Framework for Digital Media Knowledge Extraction, IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 1841-1850, doi: 10.1109/BigData47090.2019.9005480.

**Bureau of Safety and Environmental Enforcement.** https://www.data.bsee.gov/

**Directory of Azure Cloud Services |** Microsoft Azure. https://azure.microsoft.com/en-in/services/

**Wu, Jian & Killian, Jason & Yang, Huaiyu & Williams, Kyle & ray choudhury, Sagnik & Tuarob, Suppawong & Caragea, Cornelia & Giles, C.. (2015). PDFMEF:** A Multi-Entity Knowledge Extraction Framework for Scholarly Documents and Semantic Search. 1-8. 10.1145/2815833.2815834.

**Singh, Mayank & Barua, Barnopriyo & Palod, Priyank & Garg, Manvi & Satapathy, Sidhartha & Bushi, Samuel & Ayush, Kumar & Rohith, Krishna & Gamidi, Tulasi & Goyal, Pawan & Mukherjee, Animesh. (2016).** OCR++: A Robust Framework For Information Extraction from Scholarly Articles.