NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOLL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

MASTER OF SCIENCE
COMPUTER, TELECOMMUNICATIONS AND NETWORK ENGINEERING

MASTER OF SCIENCE THESIS

Distributed PCA Techniques

Christos E. Magkaniaris

Supervisor: **Stathes P. Hadjiefthymiades,** Professor

ATHENS

MARCH 2022

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΣΤΗ ΜΗΧΑΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ, ΤΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κατανεμημένες Τεχνικές Ανάλυσης Κύριων Συνιστωσών

Χρήστος Ε. Μαγκανιάρης

Επιβλέπων: **Ευστάθιος Χατζηευθυμιάδης,** Καθηγητής ΕΚΠΑ

ΑΘΗΝΑ

ΜΑΡΤΙΟΣ 2022

MASTER OF SCIENCE THESIS

Distributed PCA Techniques

Christos E. Magkaniaris
S.N.: EN3180004

Supervisor: **Eustathios P. Hadjiefthymiades,** NKUA Professor

EXAMINATION COMITEE **Panagiotis Stamatopoulos,** NKUA Professor
Konstantinos Kolomvatsos, University of Thessaly

March 2022

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κατανεμημένες Τεχνικές Ανάλυσης Κύριων Συνιστωσών

ΧΡΗΣΤΟΣ Ε. ΜΑΓΚΑΝΙΑΡΗΣ
Α.Μ.: ΕΝ3180004

ΕΠΙΒΛΕΠΩΝ: **Ευστάθιος Π. Χατζηευθυμιάδης,** Καθηγητής ΕΚΠΑ

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ **Παναγιώτης Σταματόπουλος** Καθηγητής ΕΚΠΑ
Κωνσταντίνος Κολομβάτσος Πανεπιστήμιο Θεσσαλίας

Μάρτιος 2022

ABSTRACT

Data volume creation and consumption presents a continuous increasing tendency. Mass sensor deployment and continuous monitoring are some of the reasons why this phenomenon is observed. According to a survey, data created, consumed and stored will rise from 79 zetabytes in 2021, to 94 zetabytes in 2022 and is forecasted to reach 181 zetabytes for 2025 [1]. Consequently, the need for resources to store and deliver large amounts of data, as well as the time needed to process them, increases proportionally. Dimensionality Reduction is the method of reducing the number of random variables under consideration. Principal Component Analysis, or PCA, is a popular dimensionality-reduction method that is often used to reduce the dimensionality of large data sets. This is done by transforming a large set of variables into a smaller one that still contains most of the information in the large set. What is going to be conducted is a distributed PCA algorithm scheme in an effort to address high computational costs of linear (orthogonal) PCA analysis.

Main goal of this thesis is to reduce the total computing costs by reducing the communication costs and at the same time examining the effects of grouping costs.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Dimensionality Reduction

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Dimensionality Reduction, PCA analysis, Distributed PCA, Dataset Reduction

ΠΕΡΙΛΗΨΗ

Η δημιουργία δεδομένων και η ανάγκη για διακίνηση μεγάλου όγκου δεδομένων παρουσιάζει μια συνεχώς αυξανόμενη τάση. Η μαζική χρήση αισθητήρων και οι εφαρμογές συνεχόμενου ελέγχου είναι μερικοί από τους λόγους για τους οποίους παρατηρείται αυτό το φαινόμενο. Σύμφωνα με μια έρευνα, τα δεδομένα που δημιουργούνται, καταναλώνονται και αποθηκεύονται θα αυξηθούν από 79 zetabyte το 2021 σε 94 zetabytes το 2022 και προβλέπεται να φτάσουν τα 181 zetabyte για το 2025 [1]. Κατά συνέπεια, η ανάγκη για πόρους για την αποθήκευση και τη διακίνηση δεδομένων μεγάλου όγκου, καθώς και ο χρόνος που απαιτείται για την επεξεργασία τους, αυξάνεται αναλογικά. Η τεχνική Μείωσης Διαστάσεων είναι μια μέθοδος μείωσης του αριθμού των τυχαίων μεταβλητών που εξετάζονται σε ένα μεγάλο σύνολο δεδομένων. Η Ανάλυση Κύριων Συνιστωσών ή PCA, είναι μια δημοφιλής μέθοδος μείωσης διαστάσεων που χρησιμοποιείται συχνά για τη μείωση της διάστασης μεγάλων συνόλων δεδομένων. Αυτό γίνεται μετατρέποντας ένα μεγάλο σύνολο μεταβλητών σε ένα μικρότερο, το οποίο εξακολουθεί να περιέχει τις περισσότερες από τις πληροφορίες του αρχικού. Αυτό που θα εξεταστεί είναι ένα κατανεμημένο σχήμα αλγορίθμου PCA σε μια προσπάθεια να μειωθεί το υψηλό υπολογιστικό κόστος της γραμμικής (ορθογώνιας) ανάλυσης PCA.Κύριος στόχος αυτής της διπλωματικής εργασίας είναι η μείωση του συνολικού υπολογιστικού κόστους, μειώνοντας το κόστος επικοινωνίας και εξετάζοντας παράλληλα τις επιπτώσεις του κόστους ομαδοποίησης.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Μείωση Διαστάσεων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Μείωση διαστάσεων, Ανάλυση PCA, Κατανεμημένη PCA, Μείωση συνόλου δεδομένων

Αφιερώνεται στη σύζυγο μου και τα δύο μου παιδιά,
για το χρόνο που στερήθηκαν την παρουσία μου,
προκειμένου να συνταχθεί η παρούσα εργασία,
καθώς και στην αδερφή μου,
που είναι μόνιμη πηγή έμπνευσης.

AKNOWLEDGEMENTS

I would like to address my deepest gratitude to the supervisor and scientific manager of this thesis and the scientific manager of the research team of Pervasive Computing Research of NKUA (Pervasive Computing Research Group), Mr. Efstathios Hatziefthymiadis for all the help and close support in the preparation of this work.

CONTENTS

LIST OF IMAGES

PROLOGUE

In our days, with the ever-evolving internet and its continuously increasing coverage, data needs are only getting bigger, maxing out network's infrastructures capacity. Other than that, there is an emerging need to handle this enormous amount of data, in an efficient way.

In this thesis, we examine a possible solution to this problem by implementing a dimensionality reduction technique.

1. INTRODUCTION

High-dimensional spaces are hard to work with and can easily lead to misconceptions. For many reasons, raw data are often sparse and analyzing them is usually computationally intractable. It is also a common tendency that datasets used nowadays are becoming larger in terms of features and more diverse in terms of values, in an effort to spherically examine a phenomenon and obtain more accurate results. Consequently, datasets tend to increase storage needs and process time and burden system and network resources. In addition to the former, as the volume of datasets increase, data sparsity becomes more intense, a phenomenon known as “the curse of dimensionality”.

Dimensionality reduction is the transformation of data from a high-dimensional space to a low-dimensional one, keeping at the same time, the most important properties of the original dataset that are close to the initial dimension. It is a fundamental procedure that reduces time and storage needed.

Applying Dimensionality Reduction benefits in many ways since it reduces the dimensions of the features and thus the space required to store the dataset. In addition, Machine Learning applications require less computation training time, since dimensions are reduced. Visualizing data becomes an easier and quicker task and in cases where redundant features are present, they are removed avoiding multicollinearity issues.

However, there are some disadvantages that must be taken into consideration. Data loss occurs and a lot of caution is needed, to ensure that the original hypothesis is not affected. Secondly, in the PCA dimensionality reduction technique, which we are going to implement, sometimes the principal components required to consider are unknown.

2. DIMENSIONALITY REDUCTION

2.1 Introduction –Techniques

Dimensionality Reduction is the general term used to describe the procedure of reducing the size of a dataset. There are a lot of DR Techniques and depending on the restrictions of the phenomena we are addressing there are a lot of ways to approach. The two main categories are Feature Selection and Feature Reduction.

https://medium.com/free-code-camp/an-overview-of-principal-component-analysis-6340e3bc4073

2.2 Feature Selection

Feature Selection is the process of selecting the subset of the relevant features and leaving out the irrelevant features present in a dataset to build a model of high accuracy [2]. Simply put, it is a way of selecting the optimal features from the input dataset and leaving the rest out of observation. There are three main methods of feature selection methods. The first method is by applying a filter. Some common filters used for this case are Correlation filters, Chi-Square Test, ANOVA, Information Gain, etc. The second method is by using Wrappers. The difference of this method compared to the filters is that it takes a machine learning model for its evaluation. In the wrapper's method, some features are fed to the Machine Learning model, and then performance is evaluated. According to the performance of the results the examined features are kept or removed to increase the overall accuracy of the model. The advantage of this method is that it is more accurate than the filtering one, but at the expense of a little added complexity. Some common techniques of wrapper methods are the Forward and the Backward Selection and the Bi-directional Elimination. The third category of Feature Selection methods are the Embedded Methods. The evaluation of the importance of each feature is done depending on the different training iterations of the machine learning model. Some common Embedded Methods are LASSO, Elastic Net and Ridge Regression.

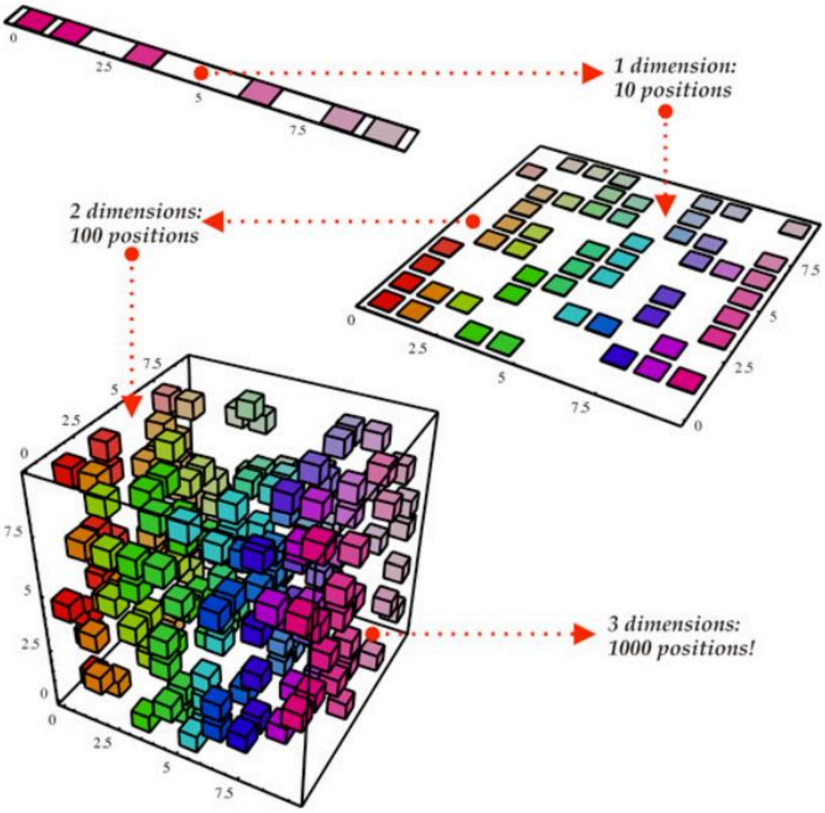

Figure 1: Dimensionality Reduction Visualization

2.3 Feature Extraction

Feature extraction is the process of transforming a space containing many dimensions into one with fewer dimensions. This approach is useful when the whole information has to be kept using the minimum resources to process it. The extraction methods can be either projection based, or components/factors based. Most common projection-based methods are ISOMAP, t-SNE and UMAP. Most factor-based methods are Factor Analysis, Independent Component Analysis and lastly Principal Components Analysis.

2.3.1 Projection-based Dimensionality Reduction

Projection-based methods aim to find a set of coefficients which make some norm of the residual function as close to zero as possible or solve some projection using test functions [3]. they are known for their power, simplicity, and relatively low error rates. According to experimental results, random projection methods preserve distances well, but empirical results are sparse [4]. The main idea behind random projection is that if points in a vector space are of adequtely high dimension, then, they may be projected into a suitable lower-dimensional space in a way that approximately preserves the distances between the points given. This is known as the Johnson-Lindenstrauss lemma [5]

2.3.2 Projection-based Dimensionality Reduction

Components or Factors-based Dimensionality Reduction are statistical methods used to describe variability among observed, correlated variables in terms of a potentially lower number of unobserved variables called factors. Factor Analysis tries to understand how the different underlying factors influence the variance among our variables. Some factors explain more variance than others, meaning that the factor more accurately represents the variables it's comprised of. Although the rest of the Components-based DR methods may appear to have similarities, they are applied on different occasions. Independant Components Analysis aims to separate information by transforming the input space into a maximally independent basis. On the other hand, Principal Component Analysis aims to compress information. The following figure briefs out the aforementioned techniques.

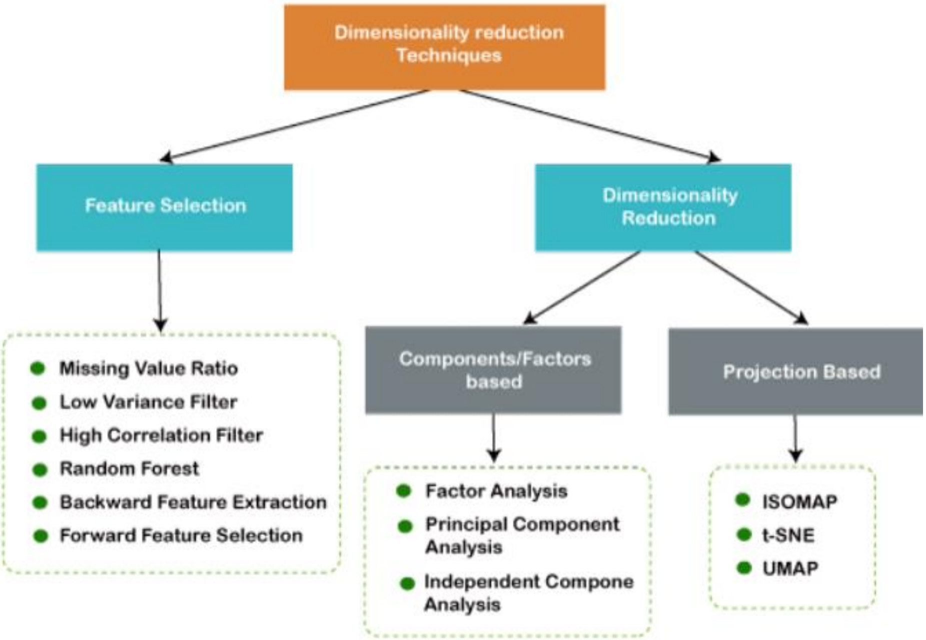

Figure 2: Dimensionality Reduction Techniques

3. PCA ANALYSIS

3.1 PCA Definition

Principal component analysis (PCA) is a technique for reducing the dimensionality of high-dimensional datasets while trying to increase their interpretability and at the same time minimize information loss. What we seek is to trade a little accuracy for simplicity. This is achieved by creating new uncorrelated variables that successively maximize variance [6]. The following steps describe the PCA algorithm. First, the range of the continuous initial variables must be standardized to ensure that each one of them contributes equally to the analysis. Following, we compute the covariance matrix to identify correlations. That is, to discover how the variables of the input data set vary from the mean, with respect to each other. The deduced conclusion is that by examining the sign of the covariance we understand the relations between them. If it is positive the two variables are correlated and increase or decrease together and if it is negative there is an inverse correlation (one increases, while the other decreases). The third step is to compute the eigenvectors and the eigenvalues of the covariance matrix to determine the principal components. Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables [7].

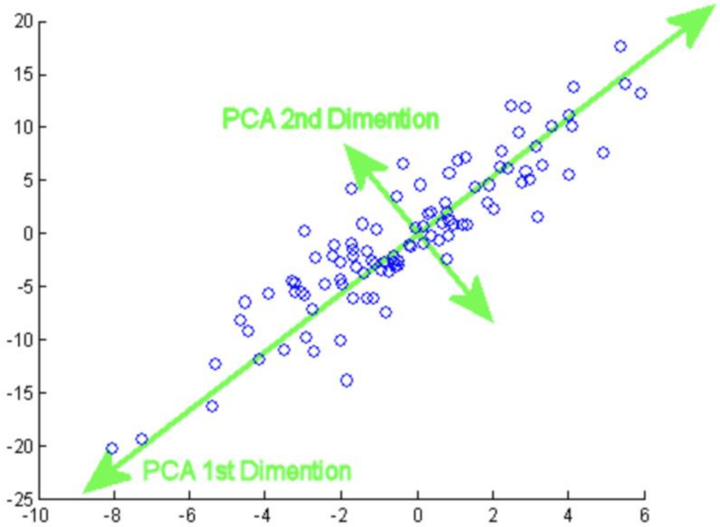

Figure 3: PCA projected dimensions

They are produced in such a manner that the first principal component accounts for the largest possible variance and to compute the percentage of variance accounted for by each component, we divide the eigenvalue of each component by the sum of eigenvalues. The next step is to create the Feature Vector, that is to form a matrix which consists of the most significant components (those with the biggest eigenvalues) and discarding those with the lowest ones. By choosing the most significant components we take the first step in dimensionality reduction since we leave out of consideration the least significant ones. Finally, we have to reposition data from the original axes to those represented by the principal components and this is done by multiplying the transpose of the original dataset by the transpose of the feature vector.

3.2 PCA Linearity

PCA is considered as a linear procedure and its linearity emerges from the mapping procedure. When moving from a high-dimensional space to a lower-dimensional space this mapping is given by a multiplication of the original matrix by the matrix of PCA eigenvectors. Since multiplication between matrices is linear, so is the procedure categorized as linear. Its linearity leads us to the conclusion that process time is linear as well. The dataset size is proportional to the process time and consequently to the network resources.

On the contrary, algorithms such as LLE (Locally Linear Embedding) ISOMAP/UMAP and t-SNE are nonlinear and their implication in distributed schemes increases complexity. Given this disadvantage, the approach in this thesis will be limited in linear algorithms to keep it as simple as it can get.

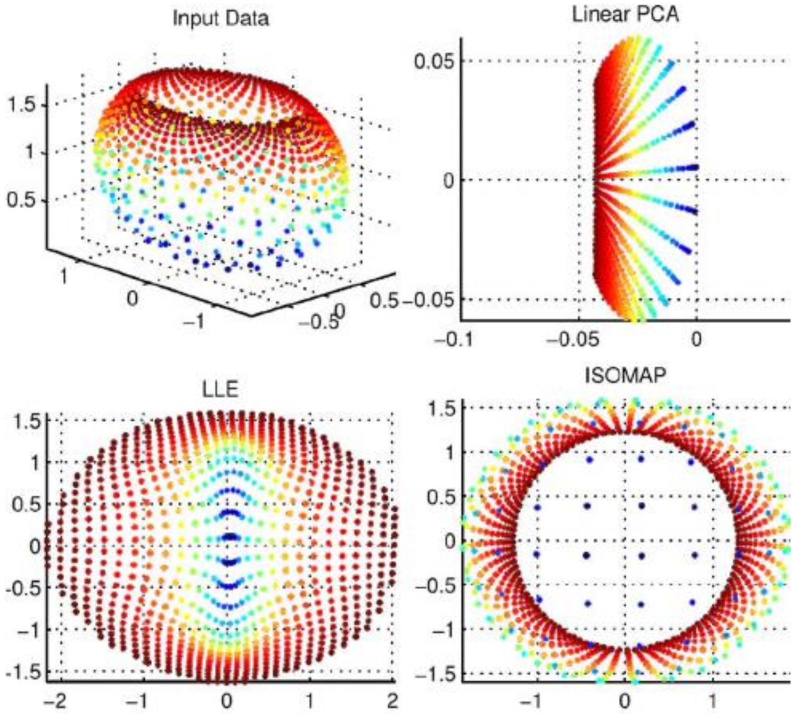

Figure 4: Comparison of linear PCA, LLE and ISOMAP on Punctured Sphere dataset.

3.3 Distributed PCA

Together with the increasing employment of distributed data acquisition systems, came the development of distributed data processing systems, on the grounds of better performance. The hypothesis of the concept is that local communication costs. can overcome the total cost of communicating and processing the whole dataset. In general, distribution approaches are summed up into two main categories. The first one refers to the way the original data are divided into sub-totals. The second one refers to the way communication costs and network architecture are computed. A key feature of up-to-date distributed PCA algorithms is that they defy the conventional notion that, the first step toward computing the principal vectors, is to form a sample covariance [8].

3.4 Other methods

Significant research has been done on how to perform PCA analysis efficiently. One line of research scientifically investigated the time and communication complexity of the process. They performed an analysis on two methods, the eigen decomposition of covariance matrix and the SVD of bi-diagonalized matrix. They proved that both are computationally intensive as their time complexities are either cubic, in terms of the dimensions of the input matrix which is quite high for many datasets. They concluded that, Stochastic SVD (SSVD) and Probabilistic PCA are two potential candidates for conducting PCA on large datasets, since they have the best computational complexity performance, but the most promising PCA approach for large datasets is the probabilistic PCA. [9]

A second line of research examined recent advancements in distributed PCA methods. These advancements were motivated by signal processing strategies that have been applied depending on how the data are acquired in the network. After comparing them to centralized PCA, they came to the conclusion that the examined methods can efficiently harness the computation and storage resources at the distributed agents. Their conclusions were confirmed by theoretical and empirical analysis as well [10].

3.5 Examined Distributed PCA scheme

We present a distributed PCA model that initially equally divides the original dataset into four sub-datasets. Distribution of datasets is done sequentially and each subset is sent to an identical server in order to be processed and get the coefficients matrix. Afterwards, the coefficients matrix is sent back to the main server and we examine the synthesis of the global coefficients’ matrix of the initial dataset. MATLAB by default, uses the Singular

Value Decompostition (SVD) algorithm to perform the principal component analysis [11]. The SVD algorithm uses three matrices, Left singular vectors, Singular values and Right singular vectors, let them be U_i, D_i and E_i respectively. After each node has computed the separate pca's, results are sent back to the main server. Subsequently, the main server subtracts the coefficients matrix of the Right singular vectors' matrix, from the Singular values one, to compute the differences, named S_i. The above-described procedure is repeated for every different node. Following, the distinct differences matrices of the subsets, are added together to create the total difference matrix. The addition is done by the main server, after the results of the separate pca's have been communicated. The steps of the procedure are summed up using the following mathematical equations.

1. $Si = Ui - Ei$
2. $\sum_{\iota=1}^{4} S = S1 + S2 + S3 + S4$

In order to compare the findings, we compare the eigenvectors of the total differences' matrix of the subsets, to the primary coefficients' matrix.

4. PREREQUISITIES – ASSUMPTIONS

4.1 Network prerequisites

4.1.1 Architecture Overview

The architecture of the server's network greatly affects the overall algorithm's performance. Architecture employed in distributed schemes can fall down into two main categories. The first one is in the context of a star network topology, based on a master-slave relation. One of the nodes is the master and is located at the center. He is charged with executing global computational tasks. Agent servers are responsible for performing the local computational tasks and communicating their results to the master in order to complete the algorithm. This is a typical architecture for parallel computation when using multicore processors. The end goal is to accelerate PCA computation by utilizing local servers processing, storage and memory resources. The second category is meshed networks and is implied when parallel processing is performed in GPUs, or in distributed storage systems. A major drawback in this implication is that in many cases, due to architecture, multihop transmissions are required resulting in undesirable communication delays.

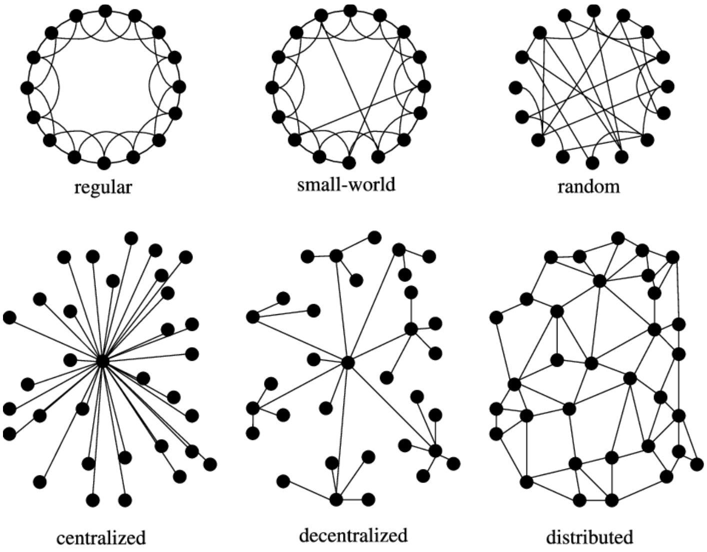

Figure 5: Most used network topologies

4.1.2 Network Description

The structure of the whole network is presented to sufficiently describe the context in which the analysis takes place. Examining the big picture makes it is easier to point out all the results that emerge. The network consists of a set of 29 smart wireless sensors strategically placed in the ship, monitoring on a 24-7 basis, a lot of basic parameters regarding the ship's status. Parameters relate to the status of the ship's engine (power, torque, etc.), its course (speed knots, latitude, longitude, water speed, tracked degrees, etc.) and its hull parameters (XYZ inclinometers, draught). In addition, external parameters are also monitored, like weather conditions (wind angle, wind speed). All sensors connect to two gateways with LoRaWAN protocol. These two gateways connect to the main network server using 802.3-2018 [12]. In this essay we consider a centralized master-slave architecture and that servers connect to each other wirelessly, using the IEEE 802.11ax-2021 protocol [13]. Network speeds according to the protocols used are considered to be of their mean value, in order to have more realistic results and take out of consideration any network underperformance. The following figure shows the architecture used.

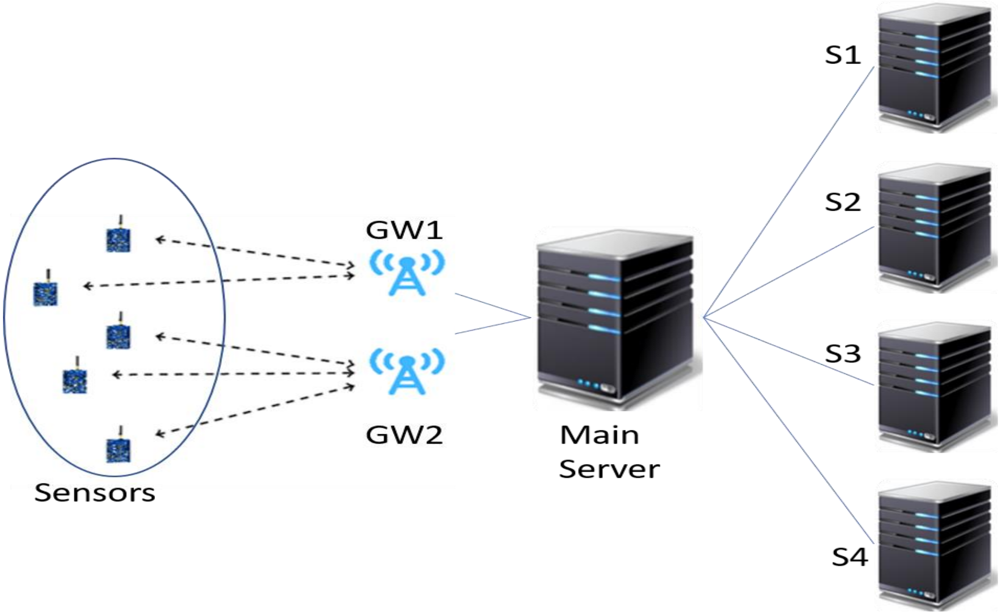

Figure 6:Networks' architecture

4.2 Servers’ specifications

Hardware plays very important role in total process time and since simulation time is going to be recorded a description of the hardware we used, is essential. We assume that all servers have the exact same specifications to avoid any miscalculations concerning process time. The specifications of the server used to run the code are the following:

- CPU: 6Cores/12 Threads
    - Base Clock 3.6GHz (Up to 4.2GHz)
    - Total L1 Cache: 384KB
    - Total L2 Cache: 3MB
- RAM: 16GB (2x8GB) Dual Channel Kit
    - Tested Speed:3200 MT/s
- Storage: 500 GB SSD
    - Read Speed 550 MB/s
    - Write Speed 520 MB/s

4.3 Dataset Description

In order to prove the hypothesis in practice, we used a relatively large dataset. It is clear that theory applies irrespective of the contents of the dataset. Same rules apply to a set of observations of weather stations, to a set of measurements coming from an airplane or even a live data streaming coming from a UAV swarm. However in every different case, distribution costs play a very important role and have to be carefully investigated. The used dataset is a 5 minutes recording of the measurements of 29 sensors placed on a commercial cargo ship. Every observation of each of the sensors is timestamped and has a steady interval of 3.472222 milliseconds. All values are listed in a spreadsheet that consists of 21,144 rows by 30 columns.

4.4 Tools used

In order to import data and run the PCA algorithm and get the simulation results we used MathWorks MATLAB® 2020a, a powerful computing platform used by engineers analyze data. It was chosen because it expresses matrices and arrays directly and because of its ability to scale the analysis on clusters with only minor code changes. For the rest of the computations and results comparison we used Microsoft Excel 2010®, since no complex computations were needed. The communications costs were calculated using the Omnicalculator® online data transfer calculator tool. Lastly, we used the cloud-based Git repository GitHub®, to release the code and make it available to everyone interested in the topic. All links are provided in Appendix I.

5. TEST OVERVIEW - RESULTS

5.1 Data Partitioning

Before proceeding to the execution of the algorithm, it is necessary to choose a data partitioning strategy based on the architecture model. We have to take into consideration all restrictions since the design of the algorithm and its performance heavily rely on the way data are partitioned, stored and communicated between servers. Data may me scattered, stored in different parts of the network and in such cases, it is preferable to perform local PCA's and communicate the most important features. When data are stored in different servers, a star architecture performs better since communication costs are lower. A second choice is to distribute data in a block-by-block, or even row-by-row basis but it is mostly applied in mesh networking architectures. In this essay we consider that all data are provided from the LoRaWAN gateways to the master-server as a matrix and then stored there.

5.2 Scenarios' description and why we chose them

At first, we are going to perform PCA to the whole dataset in the main server. Afterwards, we are going to divide the main dataset into four equal subsets and distribute them to the agent servers in order to perform the PCA algorithm and get the results. After, we return the aliquot PCA matrices to the coordinator server to examine the composition of the principal matrix that corresponds the whole data sets' main components Finally, we are going to examine the results in terms of time and resources needed and in terms of overall performance. While examining the scenarios we are going through two types of trials. One is by repeating 20 successive times the algorithm and the other by repeating it 1000 times. The reason why we chose to do so, is because we wanted to examine the algorithm's performance both in a congested and an uncongested environment.

5.3 Results

In this section the results of the mentioned tests are presented in detail. All values are measured in seconds and averaged. As depicted in the following tables, there is a significant time decrease of 19.5% approximately in the case of the non-congested environment. This decrease is more significant in the second case, reaching a decrease of 63.43%.

Table 1: Process Time for 20 iterations

	AVG PROCESS TIME (sec)	AVG DISTRIBUTION TIME (sec)	AVG TOTAL TIME (sec)	FULL DATASET PROCESS TIME (sec)	
Q1	0.49335	0.00001813	0.49335	0.61085	**-19.24%**
Q2	0.49725	0.00001813	0.49725	0.61085	**-18.60%**
Q3	0.4737	0.00001813	0.4737	0.61085	**-22.45%**
Q4	0.5019	0.00001813	0.5019	0.61085	**-17.84%**
					-19.53%

Table 2: Process Time for 1000 iterations

	AVG PROCESS TIME (sec)	AVG DISTRIBUTION TIME (sec)	AVG TOTAL TIME (sec)	FULL DATASET PROCESS TIME (sec)	
Q1	0.057640086	0.00001813	0.057640086	0.17936631	**-67.86%**
Q2	0.061745301	0.00001813	0.061745301	0.17936631	**-65.58%**
Q3	0.068945357	0.00001813	0.068945357	0.17936631	**-61.56%**
Q4	0.07406021	0.00001813	0.07406021	0.17936631	**-58.71%**
					-63.43%

As the following picture reveals the fluctuation of the average process time stays constant when in the non-congested environment while in the other case presents a rather declining variance.

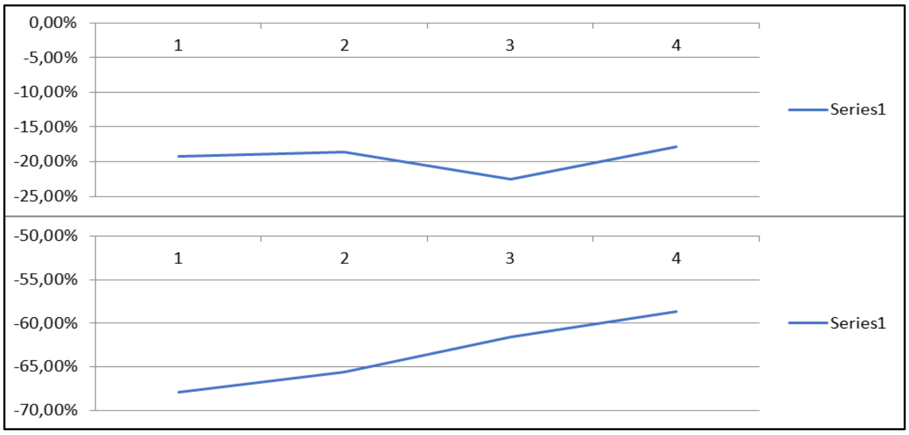

Figure 7: Process Time Fluctuation

As for each sub-dataset process time it is shown that in the first case almost every subset shows an approximate decline of 19.5% apart from the Q3 dataset that begins with a spike of -45% that progressively aligns with the rest. In the second case examined all subsets demonstrate an almost stable behavior. Every subset has a slightly different variation with Q1 being the faster with -67.86% process time, Q2 the second with -65.58% and Q3 and last Q4 to follow with a decline of -61.56% and -58.71% respectively.

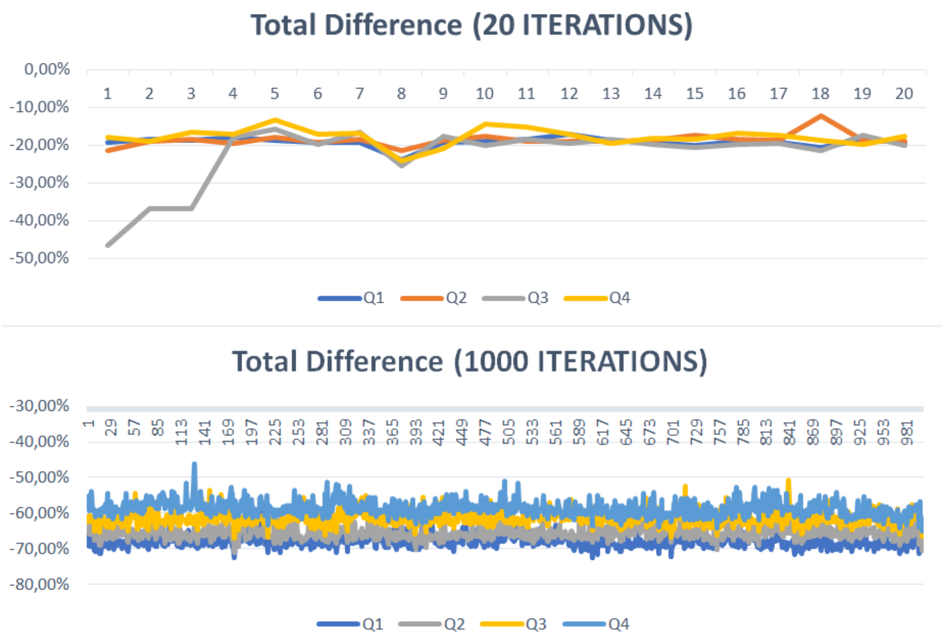

Figure 8: Process Time for each sub-dataset

In terms of performance, the metrics examined where the overall CPU and memory utilization. We examined the second scenario, as it is more resource greedy, and results showed an average increase by 17% in CPU utilization and by 19% in memory usage. These results let us assume that the overall process is not resource demanding

In terms of correlation, the comparison between the eigenvectors of the total differences' matrix of the subsets, to the primary coefficients' matrix, resulted in an acceptable variation value (2,85), in the concept of trading a little accuracy for total execution time.

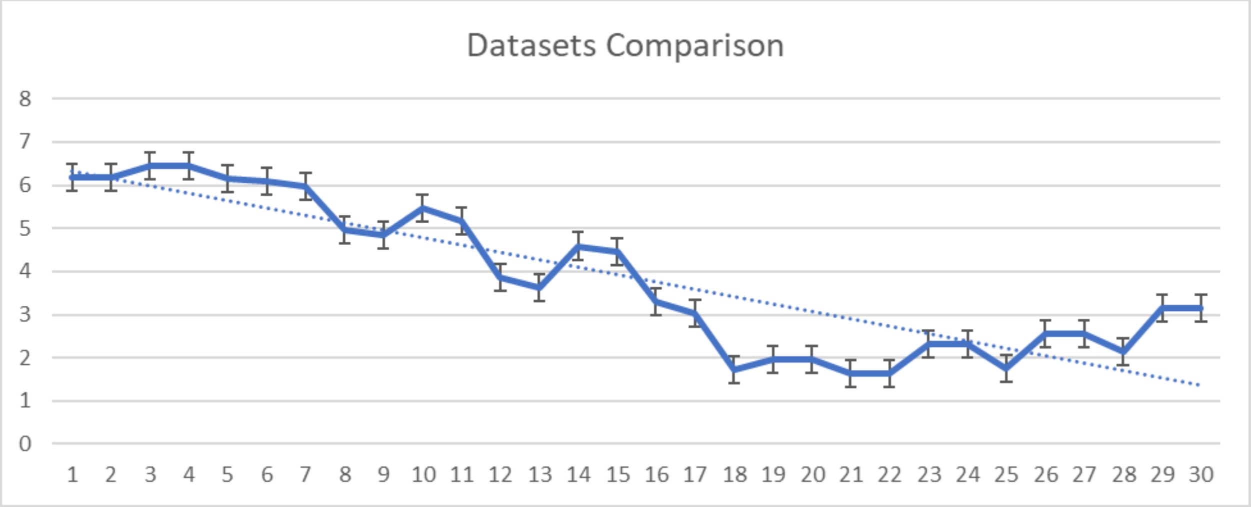

Figure 9: Variance between Initial Dataset and subsets eigenvectors

All results are available in the GitHub repository. Link is provided in Appendix I.

6. RESULTS DISCUSION

6.1 Time- Performance Comparison

The following results emerge after comparing total time needed in each process. In general, the distributed scheme seems to outperform the centralized approach by a significant margin. First, as anticipated, the total time needed was less with respect to the non-distributed scheme. In addition, it turns out that in congested environments, the difference is even bigger.

As far as the average time deviation of the processes is examined, it seems to remain constant when in the non-congested environment while on the other hand, that variance seems to decline.

As for the resource allocation, results showed that the overall procedure neither is CPU nor memory stressing. It seems also that it is not causing any network congestion. Hardware used, showed no stress during tests and so did the network. However, this is not the case for every scenario as it heavily depends on the datasets' size. In such cases there are alternative solutions such as to decrease the partitions' size or determine standard time intervals to broadcast the results. What is more, important role in resource needs play the datasets content variation and the hardware's characteristics.

As for the dataset's linear correlation, the examined algorithm showed a relatively small covariation value between the separate sub-sets and the initial coefficients matrix. That value has to be taken into consideration in cases where accuracy is of great importance. In the examined dataset accuracy can be traded for significantly lowered total execution time, and in that perspective is acceptable.

7. FUTURE WORK

7.1 Examine Smart Splitting datasets

As mentioned before the iterations performed considering as a given the splitting of the dataset to four sub-datasets. The criterion was to regard the measurements as of unknown importance so we divided the number of rows by 4. Although it might be an uncomplicated way to do so, we are not certain that this is the most efficient way. Datasets could be divided and thus processed in order of measurements importance. That hypothesis remains to be further investigated and put into test, to be able to explore the outcome with tangible results.

7.2 Examine performance on non-timestamped datasets

Data used in the test were timestamped when collected and thus values, follow a “rough” pattern. This is because measurements come from the sensor readings. While we can manually remove any anomalies or pre-select the columns we want to include in the analysis, it is interesting to investigate the algorithms behavior to non- timestamped datasets and compare the results.

7.3 Examine Performance on diverse types of networks

As described, we assumed that all clusters were connected to 802.11g wireless network. Since network protocols continue to advance, this might not be the case for contemporary implementations and it would be of great interest to examine the algorithm in distinct types of wireless or wired networks.

7.4 Apply supervised PCA

PCA is an unsupervised technique in the sense that, while computing data, it does not take into consideration any labels of the dataset. Supervised PCA is a generalization of PCA which shows satisfactory performance mainly in regression and classification problems with high-dimensional input data. It works by estimating a sequence of principal components that have maximal dependence on the response variable. There are supervised PCA algorithms that are solvable in closed-form, and have a dual formulation that significantly reduces the computational complexity [14]. Furthermore, the algorithm can be kernelized, which makes it applicable to non-linear dimensionality reduction tasks. A comparison of this approach and our method would lead to interesting conclusions.

8. CONCLUSIONS

To conclude, this essay presented a distributed PCA algorithm and tested its performance in a proposed network. The code and the results of the tests are released to illustrate the details and to prove the original hypothesis. We hope that it will provide future researchers a source to advance the state of the art in this scientific field.

ABBREVIATIONS – ACRONYMS

DR	Dimensionality Reduction
PCA	Principal Components Analysis
UAV	Unmanned Aerial Vehicle
TCP/IP	Transmission Control Protocol/ Internet Protocol
ML	Machine Learning
GPU	Graphics Processing Unit
CPU	Central Processing Unit
SVD	Singular Value Decomposition
NKUA	National and Kapodistrian University of Athens

APPENDIX I

The source code of this thesis, as well as the results of analysis, can be found in the following link:
https://github.com/x2mag/PCA-Analysis

MathWorks MATLAB can be found here:

https://www.mathworks.com/products/matlab.html

Online Data Transfer calculator can be found here:

https://www.omnicalculator.com/other/data-transfer

Microsoft Office can be found here:

https://www.microsoft.com/el-gr/microsoft-365/excel

REFERENCES

[1] Arne von See, “Amount of data created, consumed, and stored 2010-2025” https://www.statista.com/statistics/871513/worldwide-data-created/ [Accessed 10/3/22]

[2] https://www.javatpoint.com/dimensionality-reduction-technique/ [Accessed 12/3/22]

[3] Timothy P. Hubbard, Harry J. Paarsch, Chapter 2 - On the Numerical Solution of Equilibria in Auction Models with Asymmetries within the Private-Values Paradigm, Editor(s): Karl Schmedders, Kenneth L. Judd, Handbook of Computational Economics, Elsevier, Volume 3, 2014, Pages 37-115

[4] Ella, Bingham; Heikki, Mannila (2001). "Random projection in dimensionality reduction: Applications to image and text data". KDD-2001: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: Association for Computing Machinery. pp. 245–250.

[5] Johnson, William B.; Lindenstrauss, Joram (1984). "Extensions of Lipschitz mappings into a Hilbert space". Conference in Modern Analysis and Probability (New Haven, Conn., 1982). Contemporary Mathematics. Vol. 26. Providence, RI: American Mathematical Society. pp. 189–206.

[6] Jolliffe Ian T. and Cadima Jorge 2016Principal component analysis: a review and recent developments. Phil. Trans. R. Soc. A.374: 2015/02/02

[7] https://builtin.com/data-science/step-step-explanation-principal-component-analysis [Accessed 10/3/22]

[8] S. X. Wu, H. -T. Wai, L. Li and A. Scaglione, "A Review of Distributed Algorithms for Principal Component Analysis," in Proceedings of the IEEE, vol. 106, no. 8, pp. 1321-1340, Aug. 2018, doi: 10.1109/JPROC.2018.2846568.

[9] Tarek Elgamal and Mohamed Hefeeda” Analysis of PCA Algorithms in Distributed Environments “, Technical Report, 20 April 2015 p.17

[10] S. X. Wu, H. -T. Wai, L. Li and A. Scaglione, "A Review of Distributed Algorithms for Principal Component Analysis," in Proceedings of the IEEE, vol. 106, no. 8, pp. 1321-1340, Aug. 2018, doi: 10.1109/JPROC.2018.2846568.

[11] https://www.mathworks.com/help/stats/pca.html?s_tid=srchtitle_pca_1#bth9ibe-2/
[Accessed 10/3/22]

[12] "IEEE Standard for Ethernet," in IEEE Std 802.3-2018 (Revision of IEEE Std 802.3-2015) , vol., no., pp.1-5600, 31 Aug. 2018, doi: 10.1109/IEEESTD.2018.8457469.

[13] "IEEE/ISO/IEC International Standard - International Standard-Telecommunications and exchange between information technology systems — Requirements for local and metropolitan area networks —Part 1AX: Link aggregation," in ISO/IEC/IEEE 8802-1AX:2021 , vol., no., pp.1-336, 21 Sept. 2021, doi: 10.1109/IEEESTD.2021.9546726.

[14] Elnaz Barshan, Ali Ghodsi, Zohreh Azimifar, Mansoor Zolghadri Jahromi, Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds, Pattern Recognition, Volume 44, Issue 7, 2011, Pages 1357-1371, ISSN 0031-3203