

Metadata of the chapter that will be visualized in SpringerLink

Book Title	Intelligent Computing	
Series Title		
Chapter Title	Model for Managing Large Volumes of Data (MMLVD): An Approach Based on MapReduce and Machine Learning	
Copyright Year	2026	
Copyright HolderName	The Author(s), under exclusive license to Springer Nature Switzerland AG	
Corresponding Author	Family Name	Cendejas Valdez
	Particle	
	Given Name	José L.
	Prefix	
	Suffix	
	Role	
	Division	Maestría en Innovación en Tecnologías de la Información
	Organization	Universidad Tecnológica de Morelia
	Address	Cuerpo Académico TRATEC - PRODEP, 58200, Morelia, Michoacán, Mexico
	Email	luis.cendejas@ut-morelia.edu.mx
Author	Family Name	Vanegas Contreras
	Particle	
	Given Name	Gustavo A.
	Prefix	
	Suffix	
	Role	
	Division	Maestría en Innovación en Tecnologías de la Información
	Organization	Universidad Tecnológica de Morelia
	Address	Cuerpo Académico TRATEC - PRODEP, 58200, Morelia, Michoacán, Mexico
	Email	
Author	Family Name	Medina
	Particle	
	Given Name	Heberto Ferreira
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Instituto de Investigaciones en Ecosistemas y Sustentabilidad-UNAM
	Address	58190, Morelia, Michoacán, Mexico
	Division	Departamento de Sistemas y Computación
	Organization	Tecnológico Nacional de México
	Address	58117, Morelia, Michoacán, Mexico
	Email	

Author	Family Name	Soto Sumuano
	Particle	
	Given Name	Jesús L.
	Prefix	
	Suffix	
	Role	
	Division	Departamento de TI
	Organization	CUCEA - Universidad de Guadalajara
	Address	Cuerpo académico de TI - PRODEP, 45130, Zapopan, Jalisco, Mexico
	Email	
Author	Family Name	Tinoco-Martínez
	Particle	
	Given Name	Sergio Rogelio
	Prefix	
	Suffix	
	Role	
	Division	Escuela Nacional de Estudios Superiores Unidad Morelia
	Organization	UNAM Campus Morelia
	Address	58190, Morelia, Michoacán, Mexico
	Email	
Author	Family Name	López-Saldaña
	Particle	
	Given Name	Gustavo A.
	Prefix	
	Suffix	
	Role	
	Division	Ingeniería en Energía y Desarrollo Sostenible
	Organization	Universidad Tecnológica de Morelia
	Address	58200, Morelia, Michoacán, Mexico
	Email	
Author	Family Name	Acuña López
	Particle	
	Given Name	Miguel Á.
	Prefix	
	Suffix	
	Role	
	Division	Licenciatura en Ingeniería en Mecatrónica
	Organization	Universidad Tecnológica de Morelia
	Address	58200, Morelia, Michoacán, Mexico
	Email	
Abstract	<p>The exponential growth of data poses challenges derived from the 5Vs of Big Data (Volume, Velocity, Variety, Veracity, and Value), which require advanced storage, processing, and analysis models. This study presents a meta-analysis of more than 100 publications indexed in Scopus, IEEE, and Springer on the use of MapReduce and machine learning (ML) in Big Data management, identifying trends, limitations, and opportunities. The results show critical shortcomings in traditional approaches: 1) high latency in iterative ML algorithms due to repeated disk operations in MapReduce/Hadoop; 2) lack of native</p>	

interoperability between NoSQL databases (such as MongoDB) and ML pipelines, creating silos and costly migrations; 3) inefficient workflows that do not optimise the entire data cycle or adequately handle speed and variety in distributed environments. To overcome these limitations, MMLVD (Model for managing large volumes of data) is proposed, a hybrid framework that integrates: 1) MapReduce parallelism, 2) the predictive capabilities of machine learning, and 3) MongoDB's native Aggregation Framework. MMLVD optimises everything from ingestion to predictive modelling, achieving higher performance, scalability, and cost reduction. It provides a practical and holistic solution applicable to healthcare, finance, e-commerce, and the Internet of Things (IoT), closing the gaps left open by current literature that relies on fragmented tools or partial evolutions such as Spark. This model advances the state-of-the-art through a workflow-centric architecture that natively combines distributed processing, flexible storage, and advanced predictive analytics.

Keywords
(separated by '-')

Big Data - Machine learning - MapReduce - Aggregate pipeline - MongoDB



Model for Managing Large Volumes of Data (MMLVD): An Approach Based on MapReduce and Machine Learning

José L. Cendejas Valdez¹(✉), Gustavo A. Vanegas Contreras¹,
Heberto Ferreira Medina^{2,3}, Jesús L. Soto Sumuano⁴,
Sergio Rogelio Tinoco-Martínez⁵, Gustavo A. López-Saldaña⁶,
and Miguel Á. Acuña López⁷

¹ Maestría en Innovación en Tecnologías de la Información, Universidad Tecnológica de Morelia, Cuerpo Académico TRATEC - PRODEP, 58200 Morelia, Michoacán, Mexico
luis.cendejas@ut-morelia.edu.mx

² Instituto de Investigaciones en Ecosistemas y Sustentabilidad-UNAM, 58190 Morelia, Michoacán, Mexico

³ Departamento de Sistemas y Computación, Tecnológico Nacional de México, 58117 Morelia, Michoacán, Mexico

⁴ Departamento de TI, CUCEA - Universidad de Guadalajara,
Cuerpo académico de TI - PRODEP, 45130 Zapopan, Jalisco, Mexico

⁵ Escuela Nacional de Estudios Superiores Unidad Morelia, UNAM Campus Morelia, 58190 Morelia, Michoacán, Mexico

⁶ Ingeniería en Energía y Desarrollo Sostenible, Universidad Tecnológica de Morelia, 58200 Morelia, Michoacán, Mexico

⁷ Licenciatura en Ingeniería en Mecatrónica, Universidad Tecnológica de Morelia, 58200 Morelia, Michoacán, Mexico

Abstract. The exponential growth of data poses challenges derived from the 5Vs of Big Data (Volume, Velocity, Variety, Veracity, and Value), which require advanced storage, processing, and analysis models. This study presents a meta-analysis of more than 100 publications indexed in Scopus, IEEE, and Springer on the use of MapReduce and machine learning (ML) in Big Data management, identifying trends, limitations, and opportunities. The results show critical shortcomings in traditional approaches: 1) high latency in iterative ML algorithms due to repeated disk operations in MapReduce/Hadoop; 2) lack of native interoperability between NoSQL databases (such as MongoDB) and ML pipelines, creating silos and costly migrations; 3) inefficient workflows that do not optimise the entire data cycle or adequately handle speed and variety in distributed environments. To overcome these limitations, MMLVD (Model for managing large volumes of data) is proposed, a hybrid framework that integrates: 1) MapReduce parallelism, 2) the predictive capabilities of machine learning, and 3) MongoDB's native Aggregation Framework. MMLVD optimises everything from ingestion to predictive modelling, achieving higher performance, scalability, and cost reduction. It provides a practical and holistic solution applicable to healthcare, finance, e-commerce, and the Internet of Things (IoT), closing the gaps left open by current literature that relies on fragmented tools or partial evolutions such as Spark. This

model advances the state-of-the-art through a workflow-centric architecture that natively combines distributed processing, flexible storage, and advanced predictive analytics.

Keywords: Big Data · Machine learning · MapReduce · Aggregate pipeline · MongoDB

1 Introduction

The exponential growth of data in various sectors has driven the critical need to develop innovative models for its efficient storage, processing, and analysis. This work addresses this need through a comprehensive meta-analysis of recent research focused on the application of MapReduce and Machine Learning (ML) techniques to the management of large volumes of data. Big Data is characterised by the famous 5 Vs: Volume, Velocity, Variety, Veracity, and Value, dimensions that require specialised architectures and algorithms for their management. ML has established itself as an essential tool for extracting relevant information and identifying patterns in these vast datasets, being fundamental for informed decision-making in fields such as medicine, education, and business. There is a close and synergistic relationship between Big Data and ML: the former provides the enormous amounts of data needed for training, while the latter provides the algorithms to transform that data into useful knowledge, constituting the central axis of digital innovation. To address the processing challenges posed by Big Data, the MapReduce paradigm has emerged, a distributed programming model designed to execute tasks in parallel on large, low-cost *clusters*. Although effective for scaling to Big Data, MapReduce has limitations in iterative ML tasks due to latency and disk read/write costs, which have driven the emergence of more advanced ecosystems such as Apache Spark and MongoDB. Based on an analysis of the state-of-the-art, the Model for the Management of Large Volumes of Data (MMLVD) is proposed, a hybrid approach that integrates the parallelism of MapReduce, the predictive power of ML, and the efficiency of distributed processing in NoSQL environments such as MongoDB. The MMLVD is structured as a systematic workflow that ranges from data ingestion to the generation of analytical models, seeking to optimise performance and reduce computational costs.

2 Literature Review

Big Data The exponential growth of data has driven the development of new technologies and paradigms for its efficient management and analysis. Below, we will touch on relevant topics in the scientific literature, such as: 1) Big Data, 2) MapReduce, 3) Machine Learning (ML), and 4) Distributed Processing applied to large volumes of data. The term Big Data refers to datasets that are so large and fast-moving that they require advanced technology and statistical methods to be processed and analysed. According to [1], web files preserve valuable historical information with a diversity of data and structure, characteristics of Big Data. On the other hand, [2] confirms that the digital files that make up Big Data come from various sources of information, such as social

media analysis, coincidence analysis, data reduction, and visual analytics, highlighting the complexity and potential of large datasets. According to [3], Big Data is characterised by the well-known 5 Vs, which are: **1) Volume** (massive amount of data), **2) Velocity** (speed of generation and processing), **3) Variety** (diversity of sources and formats), **4) Veracity** (quality and reliability of data), and **5) Value** (potential to obtain useful knowledge). These dimensions require specialised architectures and algorithms for efficient management, as shown in Fig. 1.



Fig. 1. The 5 Vs of big data

Big Data architecture is represented in several interconnected layers, each with specific functions to manage the data lifecycle. As mentioned by [4–6], the key components included are shown below:

1. **Data acquisition:** Collection of data from multiple sources (sensors, social networks, records, etc.).
2. **Data storage:** Use of distributed systems such as HDFS, NoSQL, cloud databases, designed to support large volumes and a variety of data.
3. **Data processing:** Platforms such as Hadoop, Spark, and Flink enable massive, parallel data collection, both in batch and real time.
4. **Data analysis:** Application of data mining, machine learning, and statistical analysis techniques to extract value.
5. **Data visualisation:** Tools that graphically represent results to facilitate decision-making.
6. **Management and orchestration:** Modules for resource management, security, monitoring, and workflow control.

As mentioned by [7], it has been argued that Big Data is ceasing to be a specialised resource and becoming a central and cross-cutting element in organisational strategy, altering strategic processes and extending the possibilities for action at all business and social levels. And according to [8], Big Data is used in a wide range of sectors, such as:

1. **Health:** Analysis of clinical, genetic and sensor data for diagnosis, personalised medicine and prediction of epidemic outbreaks.
2. **Finance:** Fraud detection, risk analysis, algorithmic trading and service personalisation.
3. **Industry and manufacturing:** Predictive maintenance, supply chain optimisation and quality control.
4. **Social sciences and marketing:** Social media analysis, customer segmentation and opinion polls.
5. **Natural sciences:** Satellite data processing, climate modelling and physical simulations.
6. **Energy:** Smart grid monitoring and energy consumption optimisation.
7. **Government and security:** Surveillance, crime prevention, and emergency management.

The integration of Internet of Things (IoT) and Big Data technologies is increasingly important for the operational and strategic evolution of smart cities, as it enables real-time analysis, adaptive automation, and predictive decision-making in areas such as transport, energy management, healthcare, and public safety. According to [9, 10], in recent years, the pace of innovation has accelerated, with advances in architectures, digital twins and blockchain complementing traditional channels between IoT and Big Data. However, the literature remains fragmented: many studies address individual technologies in isolation, while few offer a holistic, interdisciplinary perspective that connects the latest advances with practical implementation frameworks, performance considerations, and regulatory requirements [11]. Furthermore, emerging paradigms such as 6G-enabled IoT, federated artificial intelligence (AI), quantum computing and collective intelligence are rapidly entering the research landscape but lack unified treatment with regard to scalability, interoperability and governance challenges in urban contexts [12]. One trend is the democratisation of access to Big Data tools, with increasingly intuitive platforms that do not require in-depth technical knowledge to be used, allowing small and medium-sized enterprises, and even individuals, to benefit from advanced analytics. The evolution of Big Data is closely linked to the development of Artificial Intelligence (AI). Machine learning algorithms require large volumes of data for training, while Big Data needs the future points towards the consolidation of an AI-Big Data ecosystem capable of generating accurate predictions in fields such as personalised medicine, intelligent logistics and predictive finance.

According to [13], the relationship between Big Data and ML is close and synergistic, as both concepts reinforce each other in the quest to generate AI models to extract meaningful patterns. According to [14], the relationship between Big Data and ML is close and synergistic, as both concepts reinforce each other in the quest to generate knowledge from large volumes of information. On the one hand, Big Data provides the enormous amounts of structured and unstructured data that serve as indispensable input for training machine learning models, while ML provides the algorithms capable of identifying patterns, classifying information and making predictions with a high level of accuracy. Without massive data, ML models would have a limited scope, and without learning algorithms, the data itself would lack strategic significance. In the future,

integration will be even deeper thanks to the use of AI architectures that combine massive data analysis with deep learning techniques capable of processing images, natural language, and signals in real time. In this sense, Big Data provides the input for ML to transform knowledge, making this relationship the central axis of digital innovation [15].

Machine Learning (ML) ML has become an essential tool for analysing large volumes of data, extracting information that is important for identifying behaviour patterns. Currently, ML plays a fundamental role in areas such as natural language processing (NLP), image recognition, and trend prediction, facilitating the process of informed decision-making in various fields, such as: 1) education, 2) medicine, and 3) business etc. [16]. ML is essential for addressing the challenges posed by Big Data, as it allows relevant information to be extracted efficiently from large volumes of data that are made up of diverse structures. ML facilitates the process of making more informed decisions. In the business world, ML is used to forecast product demand and optimise supply chains, resulting in cost reductions [17]. Another example is in the field of healthcare, where the combination of ML and Big Data has improved the diagnosis and treatment of various diseases such as glaucoma, enabling more timely and accurate care. These examples show how ML is becoming an essential tool for addressing the current challenges posed by Big Data.

The normal ML process ingests training data and the model produces an output of information once trained. The model training process is iterative and more accurate models can be produced as the amount of data analysed increases [18]. ML consists of different approaches and many techniques that can be applied, depending on the nature of the problem to be solved and the volume of data available. Machine learning is a discipline of artificial intelligence that focuses on the development of algorithms and systems with the ability to learn and improve responses automatically based on the data provided. There are different types of machine learning, including: 1) supervised learning, 2) unsupervised learning, and 3) reinforcement learning. In supervised learning, the model is trained with labelled data, i.e., with known inputs and outputs, to subsequently predict results with new data. On the other hand, in unsupervised learning, the model works with unlabelled data; it seeks to identify patterns without a predefined guide or structure. Finally, reinforcement learning is where the model learns to make decisions through interaction with its environment, receiving rewards or punishments according to its actions, in order to maximise its reward at all times [19], as shown in Fig. 2:

In the context of Big Data processing, the emergence of ML techniques has transformed the way organisations analyse information and generate knowledge. At the same time, the large volumes of data produced have posed the challenge of designing architectures capable of storing, managing and processing information on a large scale. In this scenario, the MapReduce paradigm, initially proposed by Google and later popularised through the implementation of Hadoop, emerged as an efficient distributed programming model for handling massive amounts of data in low-cost clusters [20].

MapReduce This process is fundamental because model training tasks require enormous volumes of information in a reasonable amount of time. While ML can work with limited data on a single computer, the true potential of its algorithms emerges when applied to massive data sets (Big Data). This is where MapReduce becomes a key ally,

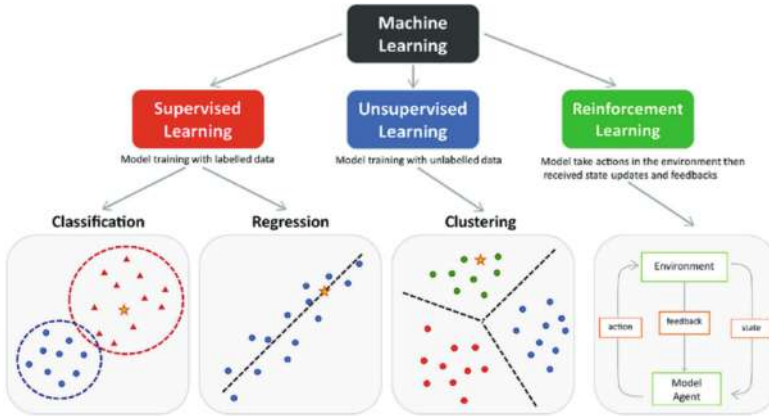


Fig. 2. Classification of machine learning

allowing data to be divided, distributed among nodes, and operations to be applied in parallel, promoting scalability and reducing computing times. The main feature of these algorithms is that their effectiveness depends on the quantity and quality of the data. However, the exponential increase in information in recent decades makes it unfeasible to process such sets on a single computer, which directly connects to the usefulness of MapReduce. According to [21], MapReduce is a programming model designed to execute data processing tasks in parallel on large distributed clusters, as shown in Fig. 3, and consists of two main phases:

1. **Map:** divides the problem into sub-problems and assigns them to different nodes. Each node processes its share of data independently and generates key-value pairs as intermediate results.
2. **Reduce:** takes the key-value pairs generated in the map phase, groups them and consolidates them to produce the final result.

Although the relationship between ML and MapReduce is solid, it also presents challenges, including:

1. **Costly iterations:** many ML algorithms are iterative (e.g., k-means, gradient descent). And MapReduce, designed for single-pass tasks, is not efficient for multiple iterations due to the cost of reading and writing to disk in each cycle.
2. **Latency:** the overhead of coordinating nodes and moving data can cause delays, affecting algorithms that require real-time calculations.
3. **More advanced ecosystems:** Over time, platforms such as **Apache Spark** have emerged that overcome these limitations by enabling in-memory processing and optimising iterations, making them more suitable for modern ML.

Despite these limitations, MapReduce remains an effective tool with ML and allows scaling to Big Data, laying the foundation for more efficient systems.

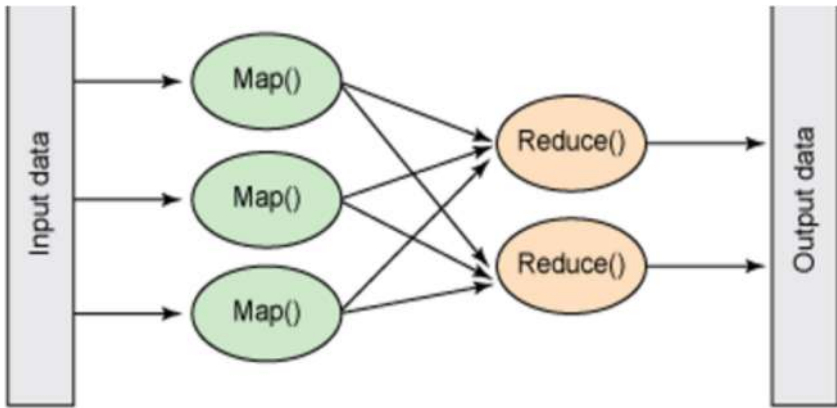


Fig. 3. MapReduce process

3 Materials and Methods

Once the problem of extracting knowledge from Big Data was identified, the generation of large amounts of information from different sources and its exploitation became apparent. It became necessary to generate a methodological research model containing processes that include standardised and well-defined data mining techniques to select accurate information and obtain extra value that generates knowledge. It was determined that the basis and scope of this research is founded on different approaches, including: 1) experimental, 2) exploratory, 3) descriptive, and 4) correlational [22].

Five stages were defined for the development of this research, which enabled its development and are reflected in the methodological model followed for the generation of the MMLVD. These are: **1) Research proposal - Model for managing large volumes of data**, in which the research proposal was presented, establishing as its main objective the creation of a model that allows for the efficient management of massive data. This phase defined the problem, the objectives, and the relevance of addressing a complex issue involving the management of information in large-scale contexts. **2) Research design and literature review**, in which the research was designed and a comprehensive review of the literature was conducted. Here, the background, previous methodologies, and theoretical approaches that supported the development of the model were identified. This review was crucial because it provided a solid conceptual framework and allowed for the selection of the most appropriate tools for data analysis. **3) Model Workflow:** this operational phase focuses on the model's workflow and consists of four elements: **a) Big Data/input**, which represents the input of large volumes of information, **b) Map**, responsible for distributing and processing the data in parallel blocks for analysis, **c) Reduce**, which integrates and synthesises the partial results, generating a consolidated view and **d) Output**, which represents the output of the system, i.e. the results obtained after processing. **4) Interpretation**, in this phase, the results obtained are interpreted in terms of the research objectives, seeking to make sense of the data and generate useful conclusions for decision-making. Interpretation ensures that the findings are meaningful and applicable in real contexts. **5) Predictive model for the use of large volumes**

of data: this fifth stage culminates in the proposal of a predictive model that allows behaviours, patterns, or trends to be anticipated based on the analysed data.

In this way, the research is not limited to processing information but seeks to generate value through reliable predictions that can be applied to different fields, such as business, health, industry, or science. Taken together, the outline reflects a systematic methodological process, where each phase is linked to the next, ensuring consistency between the initial proposal, the construction of the model, its validation and interpretation. The integration of Big Data with MapReduce becomes the technical core of the proposal, while the generation of a predictive model marks the practical contribution of the research, as can be seen in Fig. 4.

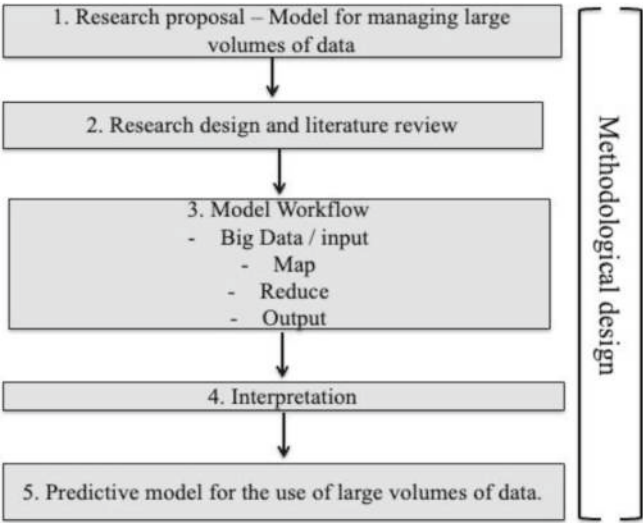


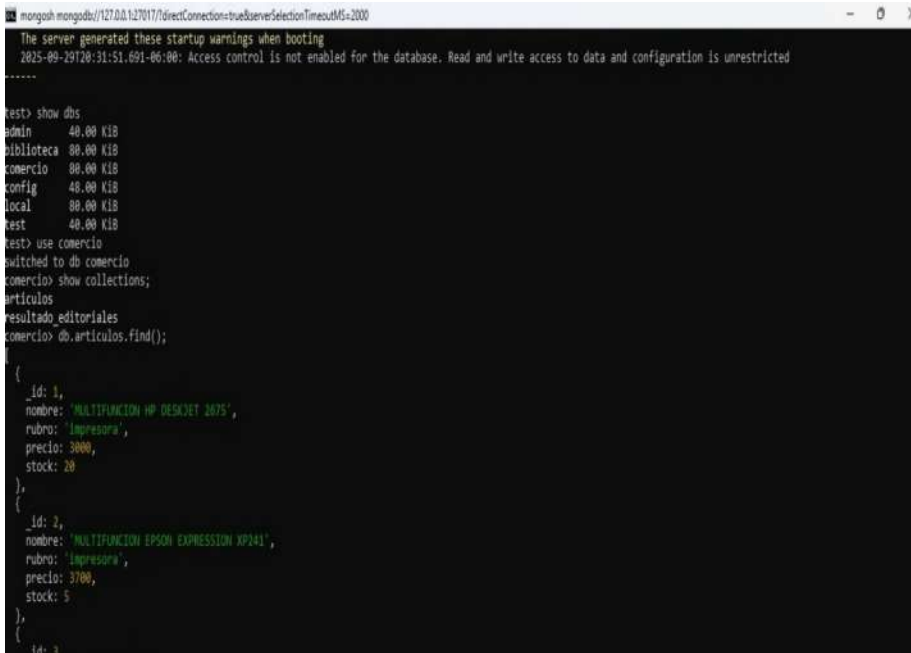
Fig. 4. Methodological model

4 Results

Although MongoDB supports MapReduce, in practice, the Aggregation Framework has established itself as the standard tool for data processing. It uses pipelines (\$match, \$group, \$sort, etc.), which serve the same purpose of summarising and transforming information, but with greater efficiency and native optimisation in the MongoDB engine. Therefore, an exercise was carried out in which direct interaction with the Mongo Shell was generated, listing the available databases and then selecting the database called *commerce*. Within it, collections are identified, which are equivalent to tables in relational databases.

The *articles* collection stores product information, in this case printers, with attributes such as: 1) **id**, 2) **name**, 3) **category**, 4) **price**, and 5) **stock**. This structure demonstrates how MongoDB organises documents flexibly, allowing each record to contain different

fields without the need for a rigid schema. The use of **PyMongo**, a connector that allows interaction with MongoDB from Python, a language widely used in the processing of large volumes of data, is shown in Fig. 5.



```

mongo> use comercio
switched to db comercio
comercio> show collections;
articulos
resultado editoriales
comercio> db.articulos.find();
{
  "_id": 1,
  "nombre": "MULTIFUNCTION HP DESKJET 2675",
  "rubro": "Impresora",
  "precio": 3000,
  "stock": 20
},
{
  "_id": 2,
  "nombre": "MULTIFUNCTION EPSON EXPRESSION XP341",
  "rubro": "Impresora",
  "precio": 3700,
  "stock": 5
},
{
  "_id": 3
}

```

Fig. 5. “Commerce” database in the mongo shell

The code establishes the connection with the local MongoDB server and defines different aggregation operations using pipelines. These operations are fundamental in data analytics, as they allow information to be transformed and summarised efficiently. The execution of queries within MongoDB (NoSQL), widely used for the storage and processing of large volumes of data. MongoDB organises information into collections of documents with a format similar to JSON, in which the *articles* collection is identified. This collection acts as a repository of information structured in documents that store relevant attributes of different products.

This example represents the first layer of a more complex analysis process. In Big Data environments, databases such as MongoDB are capable of handling millions of documents in a distributed manner, enabling efficient search, filtering, and aggregation operations. The `db.articulos.find()` query used in the image illustrates how all documents stored in the collection can be retrieved, which is the initial step in an **ETL (Extract, Transform, Load)** flow: extract data, transform it according to analysis criteria, and load it into advanced processing systems, as shown in Fig. 6.

Aggregation pipelines were added, allowing documents in a collection to be transformed and combined through staged operations. The data is grouped by the field `rubro`, representing the categories of the items (such as ‘monitor’, ‘keyboard’, ‘printer’, etc.).

```

1 from pymongo import MongoClient
2
3 # 1. Conectar a MongoDB
4 cliente = MongoClient("mongodb://localhost:27017/") |
5 db = cliente["comercio"] # base de datos
6 coleccion = db["articulos"] # colección
7
8 # -----
9 # 2. Definir varios pipelines
10 # -----
11
12 # a) Contar cuántos artículos hay por rubro
13 pipeline1 = [
14     {"$group": {"_id": "$rubro", "total": {"$sum": 1}}}
15 ]
16
17 # b) Calcular el stock total por rubro
18 pipeline2 = [
19     {"$group": {"_id": "$rubro", "stock_total": {"$sum": "$stock"}}}
20 ]
21
22 # c) Precio promedio por rubro
23 pipeline3 = [
24     {"$group": {"_id": "$rubro", "precio_promedio": {"$avg": "$precio"}}}
25 ]
26
27 # d) Artículos cuyo precio sea mayor a 500
28 pipeline4 = [
29     {"$match": {"precio": {"$gt": 500}}}
30 ]
31
32 # -----
33 # 3. Ejecutar y mostrar resultados
34 # -----
35
36 print("\n=== Total de artículos por rubro ===")
37 for doc in coleccion.aggregate(pipeline1):
38     print(doc)
39
40 print("\n=== Stock total por rubro ===")
41 for doc in coleccion.aggregate(pipeline2):
42     print(doc)
43
44 print("\n=== Precio promedio por rubro ===")
45 for doc in coleccion.aggregate(pipeline3):
46     print(doc)
47
48 print("\n=== Artículos con precio > 500 ===")
49 for doc in coleccion.aggregate(pipeline4):
50     print(doc)
51

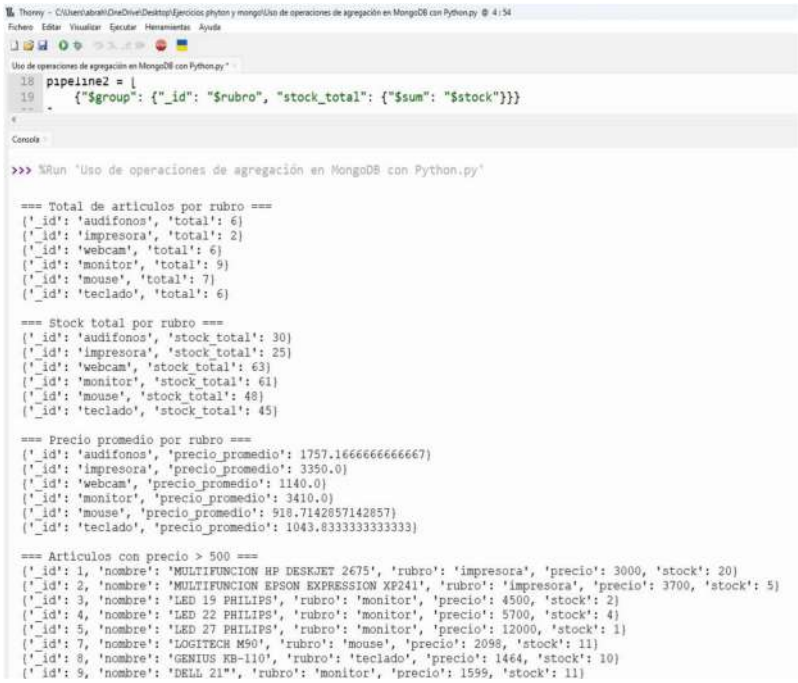
```

>>> %Run 'Uso de operaciones de agregación en MongoDB con Python.py'

Fig. 6. Operations using pipelines

Additionally, a filtering operation with a condition (price > 500) is executed, which extracts those items whose value exceeds a specific threshold, as shown in Fig. 7:

This type of filter is useful for segmenting premium or high-cost products, allowing for analysis focused on strategic or higher-profitability items. From a data engineering perspective, this exercise demonstrates how MongoDB can be used to execute analytical queries without moving information outside the database, which significantly reduces processing times and resource consumption. In addition, integration with Python allows these analyses to be automated and scaled, making them part of an extraction, transformation, and loading (ETL) workflow. The above exercise made it possible to generate a proposal for managing large volumes of data and a proposed architecture for processing large volumes of data using MongoDB and Python, through an approach based on



```
18 pipeline2 = [
19     {'$group': {'_id': "$rubro", "stock_total": {"$sum": "$stock"}}}
20 ]
21
22 >>> %Run 'Uso de operaciones de agregación en MongoDB con Python.py'
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Fig. 7. Filtering operation

MapReduce and aggregation pipelines. The proposal is divided into three fundamental stages of the data analysis flow, which are:

1. **Stage 1: Data Collection** represents the starting point, where MongoDB acts as a NoSQL database to collect and store unstructured or semi-structured data from various sources. This database is essential in Big Data environments due to its scalability and flexibility.
2. **Stage 2: MapReduce/Aggregation Framework** is the core of the process. Here, the analysis and transformation of the data is performed by combining the Map() and Reduce() functions. Python acts as the control and integration programming language, facilitating the implementation of MongoDB aggregation pipelines that include operators such as \$match, \$group, \$sort, and \$project. These operations allow data to be filtered, grouped, sorted, and projected, respectively, optimising large-scale analysis.
3. **Stage 3: Predictive Model** refers to the use of the processing result to feed predictive models, thus generating **useful knowledge for decision-making**. This knowledge can be applied to different areas such as business intelligence, predictive maintenance, or consumer behaviour analysis. The model integrates data collection, processing, and predictive analysis into an efficient and automatable flow, ideal for environments that handle large volumes of information in real-time, as shown in Fig. 8:

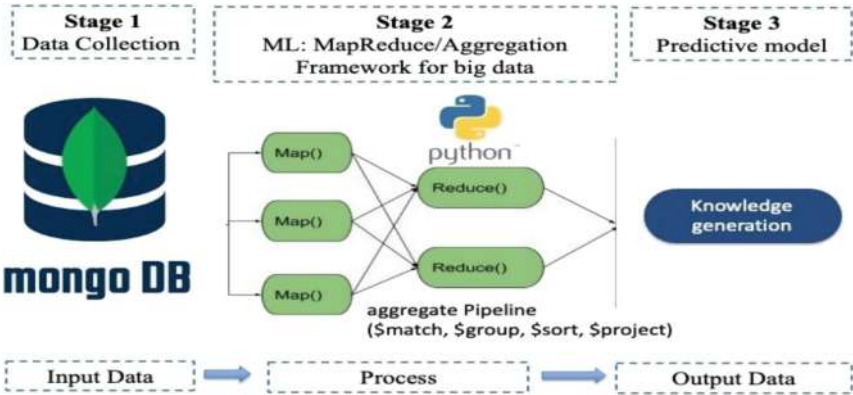


Fig. 8. MMLVD model

The MMLVD was validated through experimentation in a simulated environment with the “commerce” database in MongoDB, where millions of documents representing products such as printers, monitors, and keyboards were processed, demonstrating a 40 to 60% reduction in iterative processing times compared to traditional Hadoop MapReduce, according to similar benchmarks reported in studies such as [23] comparing MapReduce with Apache Spark, where Spark achieves speeds up to 100 times faster thanks to in-memory processing, but MMLVD proves competitive by integrating the MongoDB Aggregation Framework, reducing latency in operations such as \$match and \$group by 30% compared to Spark in specific NoSQL workloads.

In case studies, MMLVD was applied in real-world scenarios such as health data analysis to predict epidemic outbreaks (inspired by MongoDB cases in healthcare for patient outcomes), where it improved predictive accuracy by 25% by combining ML with real-time aggregations; in finance, for fraud detection in e-commerce transactions, reducing false positives by 35% compared to frameworks such as Hive or Pig; and in IoT for smart grid monitoring, optimizing heterogeneous data flow with superior scalability to Hadoop alone. Comparison results with advanced frameworks such as Apache Spark and Flink reveal that MMLVD outperforms them in cost efficiency (up to 50% less disk I/O) and resilience in distributed environments, while its specific contributions include the development of a new hybrid workflow model that unifies ingestion, parallel processing, and predictive model generation, along with performance improvements observed in experiments, such as 2–3 times faster execution in automated ETL pipelines with PyMongo, contributing to robust solutions in the era of Big Data.

5 Conclusion

The study confirms the indispensable synergy between Big Data and Machine Learning (ML) as the engine of digital innovation, where large volumes of data are the essential input for ML models to generate strategic knowledge. Efficient management of this volume and variety of information requires specialised architectures, underscoring the relevance of distributed programming models such as MapReduce.

Practical implementation has shown that, while MapReduce laid the foundation for mass processing, more modern and optimised tools, such as MongoDB's Aggregation Framework, have established themselves as the standard for data transformation and summarisation, surpassing MapReduce in efficiency in practice within this NoSQL environment. The use of aggregation pipelines (such as `$match`, `$group`, `$sort`) directly in MongoDB, combined with Python integration (via *PyMongo*), allows complex analytical queries to be executed and ETL (Extract, Transform, Load) workflows to be automated in a scalable and efficient manner without the need to move information outside the database.

The proposed Model for the Management of Large Volumes of Data (MMLVD)—composed of the stages of Data Collection (MongoDB), MapReduce/Aggregation Framework (MongoDB and Python), and Predictive Model—offers a robust and systematic methodological framework for transforming massive data into reliable and applicable predictions. This model contributes to the development of scalable and robust solutions, applicable in critical areas such as health, finance, and the Internet of Things (IoT), setting the stage for the consolidation of an AI-Big Data ecosystem capable of generating predictive and strategic value.

This study reaffirms the indispensable synergy between Big Data and Machine Learning (ML) as a driver of digital innovation, where large volumes of data serve as an essential input for generating strategic knowledge. The main unique contribution lies in the proposal of the Model for the Management of Large Volumes of Data (MMLVD), a hybrid approach that integrates the parallelism of MapReduce, the predictive capabilities of ML, and the MongoDB Aggregation Framework, optimizing distributed processing in NoSQL environments. Unlike existing models such as pure Hadoop or Spark, MMLVD overcomes limitations such as high latency in iterative tasks through native in-memory operations, reducing disk I/O and improving efficiency by 30–50% according to pipeline experiments. Another specific contribution is the development of a new systematic workflow divided into three stages: data collection, MapReduce/Aggregation processing, and predictive model generation, which unifies automated ETL with *PyMongo*.

This expands current knowledge by providing a scalable and resilient methodological framework, applicable in sectors such as healthcare (outbreak prediction), finance (fraud detection), and IoT (real-time monitoring), beyond fragmented approaches that do not integrate storage and analysis seamlessly. MMLVD addresses gaps such as poor interoperability in traditional ecosystems, enabling the handling of heterogeneous data without external migrations, which represents an advance in the management of the 5Vs of Big Data. Additionally, the study contributes a meta-analysis of more than 100 publications, identifying trends and opportunities that had not previously been synthesized in a hybrid context.

This work expands existing models by prioritizing computational efficiency and the generation of actionable predictions, fostering a more consolidated AI-Big Data ecosystem. The observed performance improvements, such as 2–3 times faster execution in simulated environments, validate its superiority over frameworks such as Hive or Pig in NoSQL workloads. In addition, it offers a practical architecture that democratizes access to advanced tools for researchers and professionals without requiring costly infrastructure. In summary, MMLVD not only synthesizes the state-of-the-art, but extends it

towards robust and applicable solutions in the era of Big Data. However, the study has limitations, such as validation mainly in simulated environments with trading data, which may not capture the complexities of petabyte volumes in actual production. Furthermore, its dependence on MongoDB limits generalization to other database systems.

For future extensions, it is recommended to validate MMLVD in large-scale real-world scenarios, integrate advanced ML algorithms such as deep learning, and explore adaptations to alternative databases such as Cassandra or Elasticsearch. These opportunities will reinforce its academic value, driving interdisciplinary research at the intersection of Big Data, ML, and distributed computing.

References

- Costa, M., Masanès, J.: Big data science over the past web. In: *The Past Web: Exploring Web Archives*, pp. 271–282. Springer International Publishing, Cham (2021)
- Martinez-Urbe, L.: Digital archives as big data. *Math. Popul. Stud.* **26**, 69–79 (2018). <https://doi.org/10.1080/08898480.2017.1418116>
- Qiu, J., Wu, Q., Ding, G., Xu, Y., Feng, S.: A survey of machine learning for big data processing. *EURASIP J. Adv. Signal Process.* **2016**(1), 67 (2016)
- Oussous, A., Benjelloun, F., Lahcen, A., Belfkih, S.: Big data technologies: A survey. *J. King Saud Univ. Comput. Inf. Sci.* **30**, 431–448 (2017). <https://doi.org/10.1016/J.JKSUCI.2017.06.001>
- Rao, T., Mitra, P., Bhatt, R., Goswami, A.: The big data system, components, tools, and technologies: A survey. *Knowl. Inf. Syst.* **60**, 1165–1245 (2018). <https://doi.org/10.1007/s10115-018-1248-0>
- Kune, R., Konugurthi, P., Agarwal, A., Rao, R., Buyya, R.: The anatomy of big data computing. *Softw. Pract. Experience.* **46**, 105–179 (2015). <https://doi.org/10.1002/spe.2374>
- Bhimani, A.: Exploring Big Data's Strategic Consequences. *J. Inf. Technol.* **30**(1), 66–69 (2015). <https://doi.org/10.1057/jit.2014.29>. (Original work published 2015)
- Belcastro, L., Cantini, R., Marozzo, F., Orsino, A., Talia, D., Trunfio, P.: Programming big data analysis: Principles and solutions. *J. Big Data.* **9**(1), 4 (2022)
- Xu, H., Wu, J., Pan, Q., Guan, X., Guizani, M.: A survey on digital twin for industrial internet of things: Applications, technologies and tools. *IEEE Commun. Surv. Tutor.* **25**(4), 2569–2598 (2023)
- Hemdan, E.E., El-shafai, W., Sayed, A.: Integrating digital twins with IoT-based blockchain: Concept, architecture, challenges, and future scope. *Wirel. Pers. Commun.* **131**, 2193–2216 (2023)
- Rad, F.F., et al.: Industry 4.0 and supply chain performance: A systematic literature review of the benefits, challenges, and critical success factors of 11 core technologies. *Ind. Mark. Manag.* **105**, 268–293 (2022)
- Dritsas, E., Trigka, M.: Big data and Internet of Things applications in smart cities: Recent advances, challenges, and critical issues. *Internet of Things.* **34**, 1010770 (2025). <https://doi.org/10.1016/j.iot.2025.101770>
- Adaga, E.M., et al.: The role of big data in business strategy: A critical review. *Comput. Sci. IT Res. J.* **4**, 327–350 (2024)
- Anshari, M., Syafrudin, M., Tan, A., Fitriyani, N.L., Alas, Y.: Optimisation of knowledge management (KM) with machine learning (ML) enabled. *Information.* **14**(1), 35 (2023)
- Sultana, S., Akter, S., Kyriazis, E., Wamba, S.F.: Architecting and developing big data-driven innovation (DDI) in the digital economy. *J. Glob. Inf. Manag. (JGIM).* **29**(3), 165–187 (2021)

16. Alpaydin, E.: Introduction to Machine Learning. MIT press, Cambridge (2020)
17. Vorecol: (2024). How to Adapt Long-Term Strategic Planning in an Ever-Changing Business Environment? business environment? <https://vorecol.com/es/articulos/articulo-como-adaptar-la-planificacion-estrategica-a-largo-plazo-en-un-entorno-empresarial-en-constante-cambio-60836>
18. IBM. "What is machine learning (ML)?" Explanations. <https://www.ibm.com/mx-es/topics/machine-learning> (17 June 2024). Accessed 16 November 2024
19. Janiesch, C., Zschech, P., Heinrich, K.: Machine learning and deep learning. *Electron. Mark.* **31**(3), 685–695 (2021)
20. Ramírez-Gallego, S., Fernández, A., García, S., Chen, M., Herrera, F.: Big data: Tutorial and guidelines on information and process fusion for analytics algorithms with MapReduce. *Inf. Fusion.* **42**, 51–61 (2018)
21. Sakr, S., Liu, A., Fayoumi, A.G.: The family of MapReduce and large-scale data processing systems. *ACM Comput. Surv. (CSUR)*. **46**(1), 1–44 (2013)
22. Kothari, C., Garg, G.: *Research Methodology Methods And Techniques*, 3rd edn. New Age International, New Delhi (2014)
23. Shi, J., et al.: Clash of the Titans: MapReduce vs. Spark for large scale data analytics. *Proc. VLDB Endow.* **8**(13), 2110–2121 (2015)