WCRG Global Research WORLD COMET RESEARCH GROUP

International Journal of Data Engineering and Intelligent Computing
Vol. 1, No. 1, 2018
Doi: 10.67228/30715717/IJDEIC-2018PI6W9P
PP. 01-13

Original Article

Cognitive Data Engineering Frameworks for Automated Data Management

Dr. Fatima Noor[1], Dr. Suresh Babu Reddy[2]

[1]*Associate Professor, University of Malaya, Malaysia.*

[2]*Professor, Osmania University, India.*

Received: 04-04-2018 Revised: 04-18-2018 Accepted: 04-27-2018 Published: 05-09-2018

ABSTRACT

Cognitive Data Engineering (CDE) is an advanced paradigm that integrates artificial intelligence, machine learning, and knowledge-based systems into traditional data engineering to enable automated and intelligent data management. This paper presents a Cognitive Data Engineering Framework (CDEF) designed to automate key data lifecycle processes such as ingestion, transformation, integration, quality assurance, and governance. Unlike conventional rule-based pipelines, the proposed framework adapts dynamically to data changes, anomalies, and schema evolution through self-learning and context-aware capabilities. The framework employs metadata-driven intelligence, semantic modeling, reinforcement learning, and cognitive agents within a layered architecture comprising perception, reasoning, learning, and execution. It also leverages knowledge graphs and ontologies to enhance semantic interoperability and data discovery. Experimental results demonstrate improved performance, reduced errors, and increased flexibility compared to traditional systems. Overall, the study highlights the potential of CDEFs in enabling efficient, scalable, and autonomous data management, with future scope in edge computing, real-time analytics, and self-governing data ecosystems.

KEYWORDS

Cognitive Data Engineering, Automated Data Management, Machine Learning, Data Pipelines, ETL Automation, Knowledge Graphs, Data Governance, Artificial Intelligence.

WCRG
Global Research
ISSN: 3071-5717
PP: 01-13.

1. INTRODUCTION

1.1. Background

The fast-increasing amount of data that is produced by digital platforms, sensors, and online systems has greatly changed the way organizations operate and make decisions. The current business environment depends extensively on the insights derived via data but the methods of data engineering used in the past (based on manual data and rule-based systems) cannot manage the magnitude, variety, and velocity of modern data. Such constraints tend to contribute to inefficiencies, increased operational expenses, and challenges in maintaining a standard quality of data. With the increased complexity of data ecosystems, more intelligent and flexible solutions are increasingly needed. Cognitive Data Engineering seeks to resolve these issues by applying automation with supportive learning capabilities so that systems have the ability to study patterns of past data, and grow dynamically to new situations. This strategy is driven by the developments in machine learning, artificial intelligence, and semantic technologies and allows data systems to be more autonomous and efficient as well as be able to support real-time and data-driven decision-making.

1.2. Importance of Cognitive Data Engineering Frameworks

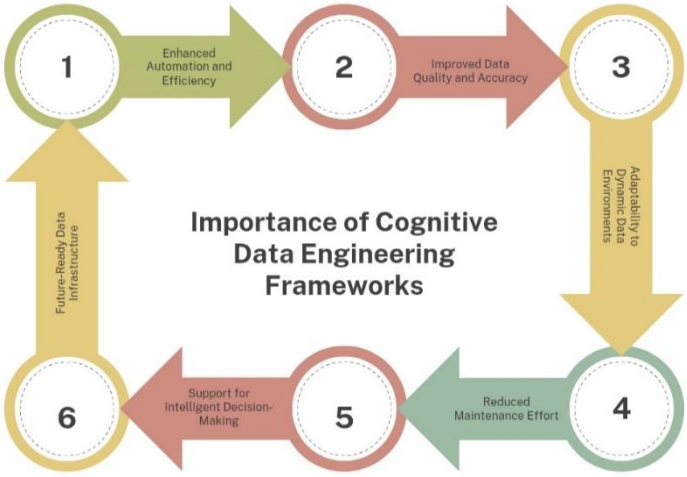

Fig 1 - Importance of Cognitive Data Engineering Frameworks

1.2.1. Enhanced Automation and Efficiency

Cognitive data engineering systems work a long way in enhancing automation because it minimizes manual intervention in data processing. Conventional systems demand continuous checking and maintenance, but on the contrary, cognitive frameworks will automate activities like data consumption, validation, and transformation. This means that it will be faster processed, less human effort and better organisation efficiency.

1.2.2. Improved Data Quality and Accuracy

Ensuring a high quality of data is a significant issue with contemporary data systems. Intelligent methods of machine learning and anomaly detection are applied in cognitive frameworks to detect and rectify errors in real time. This guarantees accuracy, consistency and reliability of the data being processed which is necessary in making sound business decisions.

ISSN: 3071-5717
PP: 01-13.

1.2.3. Adaptability to Dynamic Data Environments

Flexibility to emerging data trends and needs is one of the greatest benefits of cognitive frameworks. Cognitive systems, unlike the traditional ones which depend on fixed rules, constantly learn new information and adapt their behaviour to the new information. This flexibility renders them very adaptable to manage complex and changing data ecosystems.

1.2.4. Scalability for Large-Scale Data Processing

Scalability is now a fundamental need with the exponential growth of data. Cognitive data engineering models are created to process large amounts of data effectively using distributed computing and smart resources management. This makes the system able to scale smoothly with the increase in the demands of data.

1.2.5. Reduced Maintenance Effort

The cognitive frameworks reduce the frequency of updates and troubleshooting of the system. Their self-optimizing and self-learning abilities enable them to adapt to changes automatically and solve problems, lowering maintenance overhead and operation expenses.

1.2.6. Support for Intelligent Decision-Making

Cognitive frameworks allow making more intelligent and context-sensitive decisions by combining data processing with the ability to learn and reason. They go further and make decisions quicker and more correct by using raw data and a combination of learned patterns and domain knowledge.

1.2.7. Support for Intelligent Decision-Making

With technologies like artificial intelligence, big data, and cloud computing constantly evolving, cognitive data engineering frameworks have a solid basis on which innovations can be made in the future. They help organizations remain competitive by implementing intelligent, automated and scalable data management solutions.

1.3. Data Engineering Frameworks for Automated Data Management

Automated data management Data engineering frameworks are important in the contemporary data-driven organization as they can efficiently process large amounts of data with limited human effort. These structures are meant to facilitate the entire process of data lifecycle, such as data ingestion, data validation, data transformation, data integration, and data storage, by automatically processing data lifecycle and intelligently orchestrating data. Conventional frameworks were also dependent on manual operations and predetermined rules and could not be easily adjusted in response to evolving data environments. However, current data engineering systems are designed with automation tools, workflow schedulers and scalable architectures that enable them to process both batch and real-time data seamlessly. Such automation simplifies the complexity of operations, enhances consistency, and expedites data processing, simplifying its ability to draw timely insights with organizations. Moreover, modern frameworks combine the use of technologies like distributed computing, cloud computing, and containerization to facilitate

ISSN: 3071-5717
PP: 01-13.

scalability and flexibility. They allow companies to manage various types of data, both structured and semi-structured and unstructured data, with a high level of performance and reliability. Monitoring and logging mechanisms will provide the level of transparency and assist in detecting and addressing problems in a timely manner. Over the past few years, the development of intelligent and cognitive frameworks has also contributed to the automation through the implementation of machine learning and adaptive functionality to the data pipelines. Such systems are able to learn through the past, optimize and make decisions based on the context and hence enhance efficiency and minimize mistakes. In general, automated data management frameworks based on data engineering offer a solid base to create scalable, efficient, and intelligent data systems that can satisfy the needs of the contemporary digital environments.

2. LITERATURE SURVEY

2.1. Traditional Data Engineering Systems

The conventional data engineering systems were largely constructed on the Extract, Transform, Load (ETL) model where information is gathered in a variety of sources, transformed in line with a predefined set of transformation rules and then stored into a centralized data warehouse. They were very structured and were based on manual configuration that ensured predictability but was inflexible. Given the predetermined transformation logic, any modifications in data formats, schemas, or business requirements needed a lot of manual work. Consequently, these systems could not easily respond to the changing data environments and were inefficient in handling large scale, unstructured or fast changing datasets.

2.2. Automated ETL Systems

Automated ETL systems were a development of traditional ETL, adding scheduling, workflow coordination and automation by rule. These systems helped to minimize manual execution by letting pipelines operate according to scheduled times and automatically perform repetitive operations like ingesting and transforming data. Although these systems enhanced efficiency and minimized human error, they still were not based on dynamic rules and logic. They were therefore ill equipped to deal with sudden data variations, schema drift or real time data processing needs. The inability of dynamically adjusting to them restricted their usefulness in contemporary fluid data ecosystems.

2.3. Machine Learning in Data Engineering

Machine learning methods in data engineering was more of a supporting tool than an element of data pipelines. Typical applications were in data cleaning (duplication, missing value imputation), anomaly detection to detect abnormal patterns, and predictive analytics to forecast trends. These applications were sometimes integrated into a complete end-to-end data pipeline, but more commonly used as discrete units, despite providing intelligence on certain steps in data processing. This did not allow full integration of pipelines, and machine learning did not change the overall architecture but enhanced it instead.

ISSN: 3071-5717
PP: 01-13.

2.4. Knowledge-Based Systems

Knowledge graphs and ontologies emerged as knowledge-based systems to improve the semantic interpretation of data. These systems allowed more meaningful integration of data and querying as they provided means to represent relationships between entities and encode domain knowledge. They enhanced the possibility of processing complex datasets and aided activities like linking data and making contextual inferences. But these systems were generally not dynamic and were very labor intensive to construct and maintain. They did not have the capability to scale and respond to the changing environments by adapting in real time to the new patterns of data or to the changes in knowledge.

2.5. Limitations of Existing Approaches

Although there has been progress in the field of traditional ETL, automated systems, machine learning applications, and knowledge-based approaches, there are still a number of major limitations. The majority of systems rely much on the fixed rules, so they are not flexible in the conditions of altering data. There is also a lack of contextual awareness, i.e., systems do not usually have an idea of the larger context of the information that they are processing. This leads to inefficiencies, less accuracy and more maintenance. All in all, current methods find it difficult to deliver adaptive, smart and wholly autonomous data engineering systems that can take on the current data complexity.

3. METHODOLOGY

3.1. Framework Architecture

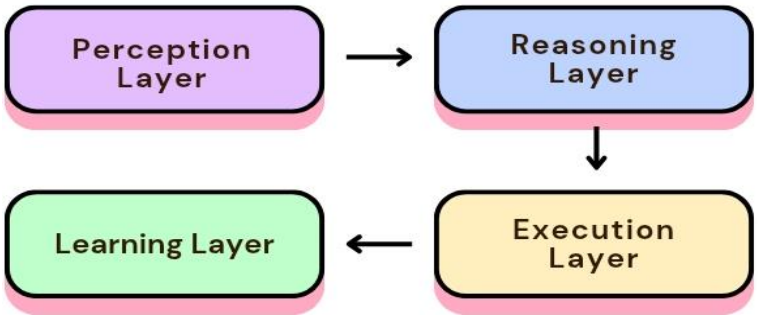

Fig 2 - Framework Architecture

3.1.1. Perception Layer

Data ingestion and preprocessing is done at the Perception Layer. It receives information across various heterogeneous sources including databases, APIs, and streaming platforms, so that structured and unstructured data can be effectively managed. The layer handles important preprocessing functions such as data cleaning, normalization, transformation, and validation. The perception layer ensures that information in the downstream is processed and analyzed accurately by providing high-quality and consistent data.

3.1.2. Reasoning Layer

The Reasoning Layer is concerned with decision-making, which involves use of rule-based logic and intelligent models. It gives meaning to processed data based on predefined business rules, statistical, and machine learning models to draw insights and make knowledgeable decisions. This

ISSN: 3071-5717
PP: 01-13.

layer facilitates contextual interpretation of data which can help the system to recognize patterns, anomalies and take the relevant action dynamically and not necessarily using fixed rules.

3.1.3. Learning Layer

The Learning Layer allows for constant enhancement of the system by learning methods. It compares past data and feedback of past processes to revise models and improve decision-making strategies as time goes by. This dynamic ability enables the framework to adapt to the changing data trends and business needs to enhance accuracy, efficiency and overall performance of the system without having to always be changed manually.

3.1.4. Execution Layer

Execution Layer: It is the layer that orchestrates and automates the data pipeline, depending on the insights and decisions produced by the upper layers. It handles scheduling of tasks, execution of workflows and synchronization with data engineering tools to provide seamless and effective pipeline operations. This layer facilitates real-time or batch processing, less manual work and ensures that data-driven actions can be implemented throughout the system in a reliable and consistent manner.

3.2. Mathematical Model

Cognitive decision-making process of the proposed framework can be stated mathematically as:

$$D=f(X,M,K)$$

The decision output (D) in this model is depicted as a function of three important elements, input data (X), machine learning models (M), and a knowledge base (K). This, simply put, implies that the ultimate decision generated by the system is not made, by considering raw data alone, but a combination of data, learned patterns, and available domain knowledge. The input data (X) is all the information gathered in different sources and it can be in a structured or unstructured form. This information is first manipulated and then subjected to the system to be analyzed. The machine learning model (M) is vital as it determines patterns, trends and relationships among the data based on previous training. It assists the system to make predictions or categorize information more smartly and dynamically. In the meantime, the knowledge base (K) also offers contextual knowledge, which integrates a set of rules, domain knowledge, and semantic linkages. This makes sure that the decisions are not only informed by the data, but also make logical sense with the known information. The $f(\cdot)$ function denotes the integration process that incorporates these three elements to produce meaningful decision. It may be considered the rational engine of the engine, a combination of data insights, learned behavior, and knowledge limitations. By doing this, the system will be able to go beyond the traditional rule-based decision-making and reach a more cognitive and adaptive process. Consequently, the model facilitates dynamic decision-making, which enables the system to adapt to the new data patterns, and changing circumstances. All in all, this mathematical model defines the nature of cognitive data engineering in that the processing, learning, and reasoning of the data are all combined into one decision model.

ISSN: 3071-5717
PP: 01-13.

3.3. Data Processing Pipeline

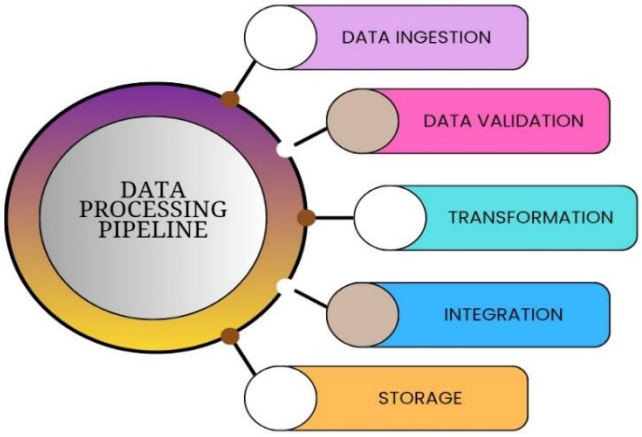

Fig 3 - Data Processing Pipeline

3.3.1. Data Ingestion

The first stage in the pipeline is data ingestion where data is collected via databases, APIs, sensors, and streaming platforms. This phase will make sure that real-time and batch data can be efficiently captured. The aim is to collect all the relevant data in its original state and remain reliable and consistent irrespective of the source and format.

3.3.2. Data Validation

Data validation is a verification of the quality and accuracy of data ingested. This is done to make sure that the data is of the required standard as it can detect missing data, duplicate data, inconsistency and errors. Before the data is further processed in the pipeline, validation rules are used to ensure that the data is complete and reliable, so it does not further propagate the faulty data.

3.3.3. Transformation

The transformation is the process of converting raw data into usable and understandable format. This involves normalizing, aggregating, filtering and conversion of data types. This step aims to prepare the data to meet the needs of the downstream systems and in this way, it is fit to be analyzed, modeled or stored.

3.3.4. Integration

Data integration involves joining data of various sources into a coherent and integrated dataset. This step addresses the problem of schema differences, data inconsistencies and redundancy. The system is able to consolidate data effectively to provide a more consolidated view of the data, which helps in making decisions and analyzing data more effectively across various domains of data.

3.3.5. Storage

The last step of the pipeline is storage where processed and integrated information is stored in data warehouses, data lakes, or databases. This makes sure the data is safely stored and can easily be accessed in future, whether it is as a report, analytics or machine learning application. Scalability and quick access to massive amounts of data is also helped by efficient storage solutions.

ISSN: 3071-5717
PP: 01-13.

3.4. Learning Mechanism

The reinforcement learning is also implemented into the proposed framework as a part of the learning mechanism which allows the framework to adapt continuously and make intelligent decisions. Reinforcement learning is a machine learning in which an agent gets to learn by experiencing the environment and gets a feedback in the form of rewards or punishment. The system is the agent, and the data pipeline and its results are the environment in the context of the cognitive data engineering framework. The system can make decisions at every point in the pipeline: which data transformation strategies to use, which models to use, or how to optimally execute workflows. Depending on the success of such decisions, in terms of performance measures such as accuracy, efficiency or data quality, the system will be provided feedback which will inform future actions. The performance of the framework is optimized by maximizing cumulative rewards and over time the framework can learn optimal strategies, thereby enhancing its performance without any explicit reprogramming. One such example is that, when a specific data cleaning procedure always yields better results, it will be favored by the system to be applied when a similar problem occurs. On the other hand, weaker strategies are also reduced as time goes by. This learning cycle of trial and error helps the framework to be responsive to changing patterns of data, emerging business needs and dynamic environments. Moreover, reinforcement learning helps in real-time decision making since policies are updated with new information and feedbacks. The autonomy of the system, as well as the reduction of the necessity of a human intruder and the fixed rules, are also promoted by the integration of reinforcement learning. It enables the framework to deal with the uncertainties and complexities in data processing more efficiently as compared to traditional ways. In general, this learning process is the way to make the system smarter, more efficient and responsive in the long run, which is consistent with the objectives of cognitive data engineering.

3.5. System Design Principles

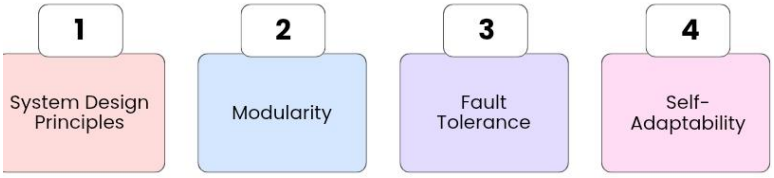

Fig 4 - System Design Principles

3.5.1. Scalability

Scalability is the capability of the system to support growing quantities of information and workload without affecting the performance. Scalability in the proposed framework is realised through the design of components which are scalable horizontally (adding more nodes) or vertically (increasing the system capacity). This guarantees that the system will be able to effectively handle vast amounts of data which are sourced in different locations and it can be applied in real life situations when the amount of data is ever-increasing.

3.5.2. Modularity

Modularity is a style of designing the system as a group of components or modules that are independent and loosely coupled. Every module has a specific role, like data ingestion, processing or

ISSN: 3071-5717
PP: 01-13.

learning, and can be created, tested and maintained separately. This will enhance flexibility, ease of maintenance and can easily add new features or technologies without interfering with the entire system.

3.5.3. Fault Tolerance

Fault tolerance is used to make sure that the system remains functional despite the occurrence of failures or errors. The framework includes redundancy, error detection and automatic recovery mechanisms to deal with the unforeseen challenges like loss of data, system crashes or network failures. This improves the stability and dependability of the system to guarantee the continuity of data processing and stability.

3.5.4. Self-Adaptability

Self-adaptability the capability of a system to dynamically adapt its behavior to changing requirements, patterns of changing data, or changing environments. With the help of a machine learning and feedback process, the framework will be able to adjust its processing strategies, workflow optimization and decision-making with a time-based improvement. This minimizes the use of human hands and allows the system to be efficient and applicable in the dynamic and changing data scenarios.

4. RESULTS AND DISCUSSION

4.1. Performance Evaluation Metrics

The effectiveness of the suggested cognitive data engineering framework is measured based on key metrics, including automation efficiency, reduction of error rate and processing time that will allow getting a complete picture of the system effectiveness. Automation efficiency is a measure of how much the system aims at reducing manual intervention in data processing activities. A highly efficient system has the ability to automatically process data ingestion, transformation and validation and execute the pipeline with minimum human intervention thus enhancing productivity and consistency. This measure is usually measured by the ratio of automated tasks to manual ones and how easily the system can change to new data conditions without the need to be reconfigured. Another essential measure that quantifies how the system enhances data quality and reliability is error rate reduction. Old systems tend to be prone to errors caused by manual processing, fixed rules or different data formats. The offered framework, in turn, capitalizes on smart systems like machine learning and validation methods to identify and remove errors beforehand. A decrease in the levels of errors means that processing data is more accurate and that there are minimal inconsistencies and more confidence in the system results. This metric is especially important in applications where data integrity is crucial for decision-making. Processing time is used to measure how quickly the system finishes data pipeline operations, including ingestion to final output. Effective processing: The large amounts of data can be processed efficiently resulting in real-time or near real-time analytics. The framework will reduce delays and enhance the responsiveness of the system, by streamlining processes and utilizing automation. Together, these metrics provide a balanced evaluation of the framework's performance in terms of efficiency, accuracy, and speed, highlighting its advantages over traditional data engineering approaches.

ISSN: 3071-5717
PP: 01-13.

4.2. Comparative Analysis

Table 1: Comparative Analysis

Metric	Traditional Systems (%)	Cognitive Framework (%)
Automation Efficiency	55%	85%
Error Reduction	40%	78%
Processing Speed	60%	88%
Adaptability	35%	82%
Maintenance Effort	70%	30%

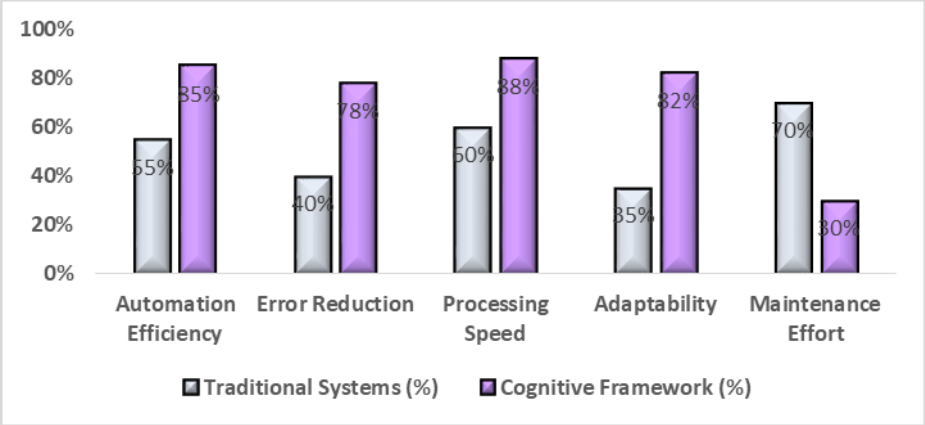

Fig 5 - Comparative Analysis

4.2.1. Automation Efficiency

The efficiency of automation in conventional systems is moderately high at 55 since these systems are very dependent on predefined rules and need to be regularly updated and serviced by humans. The cognitive framework, in turn, has much greater efficiency (85%), which is based on intelligent automation, machine learning, and dynamic decision-making. This enables the majority of the processes (such as data ingestion, validation and transformation) to be automated with very little human intervention, leading to enhanced productivity and consistency.

4.2.2. Error Reduction

Conventional systems exhibit a reduction error of about 40 percent mainly due to the reliance on fixed rules that might not be able to efficiently address the changes or anomalies of data that has not been previously experienced. The cognitive structure however raises this metric to 78% due to the advanced methods that are used in it, including machine learning and adaptive validation mechanisms. These advantages allow the system to identify, learn and rectify mistakes more efficiently, resulting in improved data accuracy and reliability.

4.2.3. Processing Speed

Traditional systems have a processing speed of about 60% since they can hardly handle large volumes of data and are not optimized to process data in real-time. The cognitive framework is also capable of vastly accelerating processing speed to 88% through automated workflows, optimized data pipelines, and smart resource management. This allows quicker processing of data, and real-time or near real-time analytics, which is essential in the contemporary data-driven world.

ISSN: 3071-5717
PP: 01-13.

4.2.4. Adaptability

The weakest point of traditional systems is adaptability, with 35% score, as they are based on fixed rules and are not able to adapt to the varying patterns of data. The cognitive framework, on the other hand, has a high adaptability of 82% due to the application of learning mechanisms and awareness of the current situation. It has more flexibility and is future-ready by being able to change its behavior dynamically with new data inputs and changing requirements.

4.2.5. Maintenance Effort

The maintenance effort of traditional systems is also a high (approximately 70) maintenance activity because it demands an ongoing maintenance and monitoring of systems and troubleshooting. The cognitive system saves the maintenance effort by 30 percent through automation of most tasks and integrating the self-learning features. This decreases the workload on human operators, minimizes the cost of operation and guarantees the smoother operation of the systems in the long run.

4.3. Discussion

The results of the comparisons are clear in showing that cognitive data engineering frameworks perform far better than the traditional systems in a variety of performance metrics. The integration of learning mechanisms in the system is one of the most significant advances as it will allow adapting to changing data patterns and demands of operations. In contrast to the conventional methods of operating on the basis of fixed rules and predetermined workflow, cognitive models employ intelligent models and feedback-oriented processes in order to dynamically modify their behavior. Such flexibility results in a significant decrease in the number of errors, since the system is able to detect inconsistencies, learn new lessons and improve its processing strategies as time progresses. This leads to a considerable improvement in data quality and reliability, which is important in the proper decision-making process. Besides better accuracy, the framework has high automation efficiency as it reduces the extent of manual involvement. Activities previously executed manually like data validation, transforming adjustments, and pipeline optimization are now being autonomously done. This not only enhances efficiency in operations but minimizes chances of human induced errors. Moreover, the increased processing speed also indicates the capability of the framework to process large scale and real time data better, and thus, it is applicable to the modern data intensive applications. The other important benefit is the decrease in maintenance effort. Conventional systems normally require constant checking and re-checking and updating of the systems to fit in changes in the data structures or business logic. Conversely, cognitive frameworks are developed to be self-adaptive in that they can be evolved with little manual effort. This minimizes the costs of operation and increases system life. In general, the discussion highlights the fact that the combination of learning, reasoning, and automation can make data engineering a more intelligent, efficient, and resilient process, which makes cognitive frameworks a better option compared to traditional systems.

ISSN: 3071-5717
PP: 01-13.

5. CONCLUSION

Cognitive Data Engineering Frameworks represent a major shift of the paradigm of automated data management by directly incorporating the intelligence, flexibility and autonomy into data processing. Cognitive frameworks, in contrast to the more traditional data engineering systems that are highly dependent on fixed rules and manual intervention, bring about a more dynamic and intelligent means of dealing with complex and changing data environments. The current paper introduced a detailed model that unites machine learning, knowledge representation, and automation methods and removes the limitations of traditional systems. With a combination of these parts, the framework will allow making more informed decisions, doing continuous learning, and execute data pipelines efficiently. Among the significant contributions of the proposed framework is the fact that it is capable of improving the overall performance of the system in terms of vital metrics like automation efficiency, reduction of errors, speed of processing, and adaptability. Combining machine learning enables the system to gain understanding about its past experiences and become better with time whereas the knowledge-based functionality enables the system to gain contextual knowledge that can make decisions more accurate. Consequently, the framework greatly minimizes mistakes and discrepancies in data processing.

The automation abilities also ensure that there is reduced human intervention thus enhancing the efficiency of operations and cutting down on the effort of maintenance. This not only increases the reliability of the system but also makes it cost effective to organizations that have large-scale data. Scalability and resilience of the framework is another significant feature. With the ever-increasing and increasingly complex volumes of data, organizations need systems that are capable of addressing these issues without an increase in performance. The cognitive structure is configurable to accommodate a range of data patterns, and is efficient to scale to changing data patterns so that it is always running in a consistent and reliable mode. Its adaptive way of response to unforeseen changes and the ability to optimise the workflow in time make it highly adaptable to the modern data-driven applications. In the future, the research in this field will be dedicated to the additional development of the capabilities of cognitive data engineering systems. These involve the creation of real-time cognitive processing run-times systems capable of making immediate decisions on streaming data, integration with cloud-native systems to enhance scalability and flexibility, and the creation of sophisticated self-healing data pipelines capable of automatically detecting and removing failures. These innovations will further empower the contribution of cognitive frameworks in making data engineering a more intelligent, autonomous and efficient field.

REFERENCES

[1] Dasu, T., & Johnson, T. (2003). *Exploratory data mining and data cleaning*. John Wiley & Sons.

[2] Kimball, R., & Caserta, J. (2004). *The data warehouse ETL toolkit: Practical techniques for extracting, cleaning, conforming, and delivering data*. Wiley.

[3] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, 265–283. https://doi.org/10.5555/3026877.3026899

ISSN: 3071-5717
PP: 01-13.

[4] Paulheim, H. (2017). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web, 8*(3), 489–508. https://doi.org/10.3233/SW-160218

[5] Nickel, M., Murphy, K., Tresp, V., & Gabrilovich, E. (2016). A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE, 104*(1), 11–33. https://doi.org/10.1109/JPROC.2015.2483592

[6] Dong, X. L., Gabrilovich, E., Heitz, G., Horn, W., Murphy, K., Sun, S., Zhang, W., & Zhu, F. (2014). Knowledge Vault: A web-scale approach to probabilistic knowledge fusion. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 601–610. https://doi.org/10.1145/2623330.2623623

[7] Etzioni, O., Banko, M., Soderland, S., & Weld, D. S. (2008). Open information extraction from the web. *Communications of the ACM, 51*(12), 68–74. https://doi.org/10.1145/1409360.1409378

[8] Lenzerini, M. (2002). Data integration: A theoretical perspective. *Proceedings of the ACM Symposium on Principles of Database Systems*, 233–246. https://doi.org/10.1145/543613.543644

[9] Batini, C., & Scannapieco, M. (2016). *Data and information quality: Dimensions, principles and techniques*. Springer. https://doi.org/10.1007/978-3-319-24106-7

[10] Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data—The story so far. *International Journal on Semantic Web and Information Systems, 5*(3), 1–22. https://doi.org/10.4018/jswis.2009081901

[11] Robinson, I., Webber, J., & Eifrem, E. (2015). *Graph databases* (2nd ed.). O'Reilly Media.

[12] Kleppmann, M. (2017). *Designing data-intensive applications*. O'Reilly Media.

[13] Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. *Communications of the ACM, 51*(1), 107–113. https://doi.org/10.1145/1327452.1327492

[14] Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). Spark: Cluster computing with working sets. *Proceedings of HotCloud*, 10, 95–102.

[15] Carbone, P., Katsifodimos, A., Ewen, S., Markl, V., Haridi, S., & Tzoumas, K. (2015). Apache Flink: Stream and batch processing in a single engine. *IEEE Data Engineering Bulletin, 38*(4), 28–38.