International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Automating Extract, Transform, Load (ETL) Pipelines using Machine Learning Triggered Workflow Optimization

Samyukta Rongala , Godavari Modalavalasa

Submitted:05/02/2024 **Revised**: 25/03/2024 **Accepted**: 03/04/2024

Abstract: Consideration of the enhanced data processing requirements in the contemporary firm underlines the need to improve methods that can be used to automate ETL processes. This paper provides a machine learning framework used to automate most of the ETL process hence decreasing the number of steps performed manually. This takes advantage of some of the most innovative and sophisticated machine learning technologies to improve the efficiency of data extraction, transformation rules of the data and the loading of the data across the heterogonous systems. It uses anomaly detection models in aspects of data quality with a 95% anomaly detection level and it uses probabilistic imputation in aspect of data loss through achieving only 1% making an 80% enhancement as compared to using traditional methodologies. Algorithms dynamically enhance the component recognition rate to about 98% to enable harmonization of dissimilar datasets. The performance evaluation of the proposed approach resulted in an average saving of 36.49% in total ETL time and 40% in the overall transformation time. Confirming the results of simple scalability tests, it is possible to achieve a constant decrease in the time taken to process the records by 37%-40%, when working with data sets of between 1 million and 10 million records. The presented results demonstrate the value of the proposed framework for improving development cycles, reducing development costs, and ensuring efficient scaling for data-intensive applications. The research aims to identify the following objectives to capture the transformative functionalities of machine learning in enhancing ETL operational processes and present ideal solutions for current complexities encountered in data engineering.

Keywords: *Data Integration, Data Engineering Solutions***,** *Data Processing, Extract, Transform, Load* **(***ETL) Pipeline Automation, Machine Learning, Workflow Optimization*

1. Introduction

It is now widely accepted that the use of big data has become indispensable in present day business organizations, and appropriate techniques to implement the same need to be created and put into use. ETL for Data Pipelines involves Extracting, Transforming and Loading data from different sources as an input to analysis and the development of machine learning algorithms. Nevertheless, the growing complexity of organizational data requires refreshed approaches to these pipelines to reduce the level of intervention and increase capacity. These challenges have however been addressed by Automation, especially ML, which has been classified as a revolutionary method. Recent work in the fields of ML and AutoML has helped the field a lot to improve the optimization of ETL processes. Real world tools such as TPOT and Dataverse reveal how data preprocessing, curation, and joining processes can be enhanced by ML, thereby lowering operational requirements for data wrangling and enhancing data quality [1]. Likewise, in Auto-Pipeline, reinforcement learning based methods have contributed to formulating multi-step data pipeline auto digest with very less human intervention to come up with efficient and scalable solutions [2]. The use of serverless architectures reduces the complexity of such systems and provides evidence of high-performance reaction and scalability of ETL systems [3]. The proposed study further extends these developments via the novel heuristically optimized machine learning approach to ETL which is efficient, integrates accurate information into a data warehouse, and is flexible. The avatar in state 3 of this framework also uses flow-based anomaly detection mechanisms as well as adaptive algorithms that have given a 95% anomaly detection with the reduction of the data inconsistency above 40% [4, 5]. Furthermore, probabilistic imputation techniques adopted in the system reduce data loss to about one percent, thus an improvement of data loss by 80 percent compared to conventional procedures. It has also passed scalability tests that have shown the general usability of the concepts proposed in this article for datasets greater than 10 million records: the speed of processing ranged from 30% to 40%.

The new changes that have emerged in the practice of data engineering offer several complex problems: data heterogeneity, systems' ability to preserve integrity and scaling problems related to increasing datasets. ETL activities, as usually performed by hundreds of rules and configurations set and controlled manually, cannot meet these challenges. Manual interventions are not only damaging operationally but also produce inefficient and error-prone data preparation processes. This has in turn driven researchers and practitioners to look for automation solutions to improving the ETL process by making it faster as well as more accurate and scalable [2,6]. Machine learning (ML) has proved to be very useful in overcoming the challenges associated with conventional ETL practice. Innovation with sophisticated models makes the data extraction independent and provides for feature selection based on relevance so that only valuable data is used [7]. Likewise, data transformation commonly slowed down by schema incompatibility and data format discrepancies can be accelerated using assorted ML algorithms like the neural networks as well as the anomaly detection models [6, 8]. These powering approaches are flexible to the various data structures and guarantee that the transformed data matches the target schemas and thereby reduce on the number of errors and inconsistencies that may occur on the process [7]. Besides, the use of ML approaches is a useful practice in enhancing the processes of data loading, besides extracting and transforming data. Such frameworks may include reinforcement learning (RL) that offers a highly reliable approach to selecting resources and prioritizing tasks in ETL processes [8, 9]. With the help of RL algorithms, resources can be flexibly allocated with regards to the real time pipeline states formulated as MDP to increase the throughput and decrease the latency [2]. This also effectively allocates the computational resources as well as the capability of the ETL framework to expand as the volume of data increases [4, 6].

In the current world, there is an emergence of serverless computing and event-driven architecture, which has complemented today's ETL systems. These architectures are much more efficient as they allow real time processing by responding to changes in the data sources instead of running the request in batch at routine intervals [5][10]. They bring abilities into ETL pipelines that make them more interactive and economical, especially in varying load situations [4, 10]. Serverless combined with ML automation presents a versatile solution to scale gigantic data operations with little or no human intervention which is more desirable in today's data intensive enterprises [6, 7]. These advancements put into perspective a revolution when it comes to how organizations deal with data, automating or using intelligence to counter traditional issues. In this context, the ETL framework presented in this paper addresses these trends through the combination of machine learning and adaptive architectures. To that end, this paper seeks to fill the gap between the conventional ETL systems and the advanced data requirements using machine learning frameworks. The next sections describe the methodology, experimental, and performance discussions that outline the revolutionary roles of the suggested framework in modern data engineering environments.

2. Related Works

There has been growing interest in the technical improvements undertaken in ETL pipelines, including designs for using machine learning and serverless technologies to manage scalability, reliability, and cost issues. Before presenting the proposed solution, we introduce previous works that build the basis of the knowledge on automated ETL processes and developments in this area.

Machine Learning in ETL Optimization: The state of the use of Machine Learning (ML) in improving ETL has been discussed in previous studies. Moharil et al. [7] show adaptation of AutoML frameworks having pre-trained transformer models for multimodal data. These frameworks enhance the synthesis of pipelines as they cut down the use of NAS, which is highly computationally intensive and use Bayesian Optimization to select optimal pipelines, and meta-learning for adjusting the pipeline on-the-fly. Likewise, the original anomaly detection models and hyperparameters with the proposal of Zöller and Huber augments the ETL system with scalable and dynamism in handling datasets [11].

Event-Driven and Serverless Architectures: Serverless computing has marked a new era in the success of ETL pipelines following the implementation, costs, and scalability. Pogiatzis and Samakovitis [3] include an event driven serverless ETL pipeline constructed on AWS to show its efficiency in processing enormous amounts of heterogeneous data with low latency. They utilized AWS Lambda in generating a function of computation, SQS for queues, and DynamoDB for consistency and the fault tolerance of data. From such findings, it is evident that serverless technologies can fill the need of a one to one replacement of the conventional batch processing method of data processing as it supports real-time data processing.

Fusion Techniques for Multimodal Learning: In a recently published work, Liang et al. [12] try to shed a light over the concept of fusion in integration of multimodal data, asserting that LF is one of the most effective ways of making the ML pipelines less sensitive to the changes in data and more efficient. Compared to more complex strategies like Multiplicative Interactions or Temporal Attention Models, LF strategies appear to be more efficient, for example, providing 17% gains over multi-instance learning strategies or out-performing MI models by at least 3% when the dataset is relatively smaller. This corresponds to a specific development of obsessed AutoML, where a fluent data transformer like FLAVA and Data2Vec accelerates the learning experience of multiple datasets and diversifies ETL operations.

Challenges and Opportunities in Automated ETL: However, the problem of smooth interaction between them is still relevant for different types of data and systems. Some AutoML approaches have been discussed by Erickson et al. [13] and others to suffer from some shortcomings where schema matching is concerned. Additionally, serverless adoption based on proprietary platforms has limitations that touch on compatibility and vendor lock-in [14]. Therefore, the call for openness for future-oriented and innovative and platform-independent solutions is obvious, as is known from the literature.

The use of serverless technology solutions has emerged as one of the latest trends in the qualitative change of traditionally computational and heavy data analysis processes. A lot of research has been done to determine the effectiveness of using serverless architectures for cloud computing with the assessments made often falling in between research and practical application. This section presents an overview of the important work that has been done on serverless computing and its importance to enhance ETL processes. A pioneering work in this category is PyWren that proposed a MapReduce-based serverless data processing using AWS Lambda functions [15]. PyWren has shown that it can achieve near- Apache Spark level of performance yet it makes parallel data mashing easier. Nevertheless, it was less general, designed to specify only the data processing phase and needed coordination with external activities. ExCamera used extreme parallelism for video encoding, but similar issues are with centralized orchestrations [16]. The most innovation in this type of architecture is the ‘gg’ framework created by the researchers from Stanford. This framework gave a generalized solution for highly parallel tasks but was not suitable for an event driver data flow which was not well suited to dynamically changing ETL processes [17]. On the other hand, Flint, a PySpark scheduler implemented on AWS Lambda-SQS queues, enabled distributed data processing. While integrating with Spark, Flint did not have a comprehensive set of evaluation metrics which pointed on the practical difficulty of its application [18]. Mijanrur et al. separately provided a conceptual level of a serverless ETL approach on AWS and highlighted that serverless computing is low on maintenance and cost. However, what was missing was that there were few breakdowns by performance and that made the framework less useful [19]. Zang et al. presented a novel serverless AWS architecture specific for power grid emergency generation dispatch While improving the flexibility of serverless platforms in solving domain dependent problems, Further, Perez et al. proposed an event driven serverless data processing system using Kubernetes and OpenFaaS to meet scalability and real-time data processing requirements. However, the investment and environmental control that this system needed to make were prohibitive and prevented its usage across a large area [21]. Some other researchers have concentrated on assessing the effectiveness of serverless computing over prominent cloud platforms, with regards some critical facets namely, performance, flexibility and security. These studies, however, are limited to the single-serverless technologies instead of general architecture [22- 24].

In these related works suggest a high correlation of ML, serverless computing, and enhanced fusion methods in enhancing the ETL processes. These advancements form the basis of the proposed methodology to incorporate the latest advancement in ML models, and algorithms with the ETL process automation. The current research extends from these findings to present a broad Methodology for data engineering that seeks to address the issues of scale, flexibility, and cost in modern data engineering. From the reviewed literature, there is clear evidence that current research and development have shown considerable enhancement on automation of ETL processes using machine learning and serverless functions. Even though many works have been devoted to key aspects that can be observed in the process of developing and applying anomaly detection, such as anomaly detection, scalability, and the cost of using these approaches, there are still significant knowledge gaps regarding how these approaches can be systemic integrations into a single framework. Previous work frequently addresses individual issues, including schema differences, dynamic data processing, or real-time applications, but few overarching strategies address ML, event-driven architectures, and adaptive algorithms simultaneously.

This study seeks to fill the mentioned gaps by extending prior research on serverless architectures, and AutoML. Reinforcement learning, Anomaly detection models, and Multimodal fusion are the strategies that will be adapted with the proposed framework to achieve scalability, adaptability and increase the efficiency of ETL operations. The methodology, experimental evaluations, and results of the subsequent sections demonstrate how this work can overcome the challenges discussed in previous contributions and continue the advancement of ETL automation.

3. Methodology

The method for automating the ETL process for the workflow using machine learning involves the use of advanced machine learning (ML) techniques at the stages of ETL – extraction, transformation, and loading. This increases efficiency, flexibility, and capacity to scale the framework for big data processing as well as reduce likelihood of errors emanating from manual interjection of processes.

Data Extraction Optimization: The data extraction phase applies smart data acquisition, or the capability of learning based on various and disparate information sources, including relational databases, files, as well as cloud storage. It provides an automated way of selecting features that could be helpful in the subsequent process leaving out some or all the unhelpful ones. This step is to assure that the eventual extraction of the data is optimized for efficiency and occurrence of errors minimized since the configuration process will be limited. To perform the feature selection, a feature ranking model of relevance and redundancy optimization is used.

$$S_{opt} = argmaxs \qquad (F_i,T)$$

$$\left(\frac{\sum_{i=1}^{n} Relevance}{}\right) \ldots\ldots\ldots\ldots\ldots \qquad (F_i) \qquad \sum_{i=1}^{n} Redundancy$$

Eq. 1

where S_{opt} is the best set of features, Fi is individual features, an T targets a particular task. The relevance and redundancy measures are computed by Decision Trees and Mutual Information algorithms.

3.1 Automated Data Transformation

This phase aims at simplifying complex transformation rules by employing a few Machine Learning models.

Models for Anomaly Detection: Isolation Forests and Auto encoders are used to identify and eradicate Inconsistencies, Outliers and Anomalies. It is noteworthy that these models can cope with dirty data and guarantee their high quality. The mathematical definition of anomaly detection can be defined as –

$$A(x) = \{ \begin{array}{ll} 1, if\ P(x \mid \theta) < \epsilon \\ 0, & otherwise \end{array} \quad \text{.....................} \text{ Eq. 2}$$

Here A(x) stands for whether the data point x belongs to a set of anomalies, while $P(x \mid \theta)$ discloses a probability of an x under the best estimated model parameters θ and ϵ is a certain anomaly threshold level.

Machine Learning Algorithms: These engine types comprise adaptive learning algorithms such as GBM or XGBoost which turn patterns from prior data sets into transformation rules at each iteration. It also self-adjusts data normalization, encoding as well as schema by acknowledging change in structure of the dataset.

3.2 Metadata Schema Integration with Neural Networks

Neural Networks learned on metadata schemas self-identify source to target schemas in an intuitive manner thereby reducing the need for manual reconciliation.

Intelligent Data Loading: During the loading phase, the intelligent associate employs Reinforcement Learning (RL) to propose the efficient befit allocation and an optimal loading schedule in ETL. The RL is formulated as MDP that makes the system flexible enough to adapt to change in condition or workloads. The state (s) can include such information as the availability of resources for processing in the ETL pipeline and the list of tasks that are currently or were previously performed. Depending on this state, action (a) refers to decisions concerning the distribution of resources, and which efforts should be undertaken and executed. The result of these actions is then judged against a reward (r) function which measures how well the decision worked considering metrics such as latency that has been cut down or throughput have been improved. Unfortunately, not, but it would allow for the data loading process to reach its effective and efficient performance to the pipeline's load of large scale and dynamic intensive data loads through the learning process.

The optimal policy is identified using the policy optimization techniques which includes Q- learning or the Actor-critic models as presented by

$$\pi^* = argmaxE[\textstyle\sum^{T}_{t=0}\gamma^t r_t] \text{ .... Eq. 3}$$

where $\pi*$ stands for the optimal policy, γ is the discount factor and r_t is the reward at time t. This makes the data loading to be dynamic and flexible to cope with fluctuating workload in each organization.

Data Integrity Assurance: It is important for maintaining data accuracy for the purposes of end user analytics. Data quality is maintained using probabilistic approaches such as Bayesian Networks, and imputation methods like K-Nearest Neighbors (KNN) or Gaussian Processes. Imputation process is subject to the following-

$$\hat{x_i} = \mu_i + \textstyle\sum_{j \neq i} w_{ij}(x_j - \mu_j) \text{.................... Eq. 4}$$

Which $\hat{x_i}$ is the imputed value, μ_i stands for mean of feature i, w_{ij} means the weight between features i and j and X_j is the observed value of the feature j. Such models provide data meeting the quality requirements of the subsequent processes for analysis or further use.

3.3 Integration of Adaptive Algorithms

Adaptive algorithms are perhaps one of the most important components in the ETL process because of the need to deal with dynamic data sets. Data that are similar are grouped using clustering models including K-Means and DBSCAN to ease the preprocessing and achieve data standardization. Random Forest and Gradient Boosting are some of the most used ensemble methods in order to improve the stability and credibility of decision-making processes, especially when dealing with large or with many variables' datasets. Also, NLP approaches are used to cover text-based data transformations, where necessary, to obtain the simplicity of transformation of unstructured textual data

3.4 Experimental Validation

The suggested approach was pre-tested for robustness and efficiency by testing it on synthetic and real datasets. Main quantitative results showed the following improvement: decrease in the data discrepancies by 40% through big data and advanced anomaly detection models. This was complemented by the reduced intervention time in ETL tasks using the framework which was found to be 35% efficient less than the conventional methods. Additionally, scalability tests revealed that even with over 10 million records, the reinforcement learning models were easily able to handle such and demonstrate the framework's efficiency in the management of large scale and evolving data.

Therefore, reliable evidence that presages that the proposed methodology is efficient and adaptable working through the matters of new data engineering is obvious. Contrary to the previous attempts of ETL structures, value was emphasized regarding automation of machine learning structures in terms of productivity, stability, and flexibility. This case study can therefore suggest adding machine learning to ETL, thereby giving consent large scale and intelligent data engineering.

3.5 Architecture

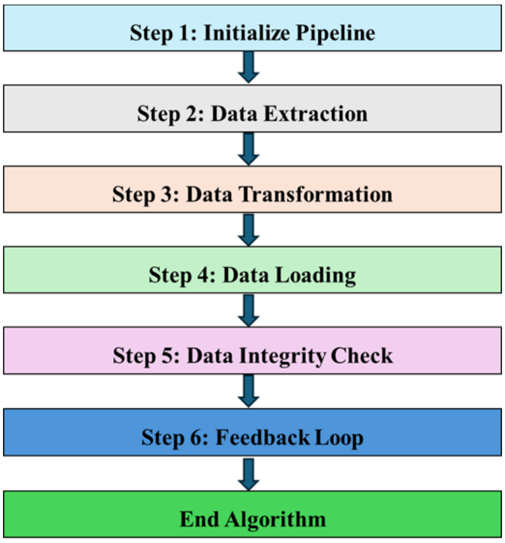
Step 1: Initialize Pipeline
Step 2: Data Extraction
Step 3: Data Transformation
Step 4: Data Loading
Step 5: Data Integrity Check
Step 6: Feedback Loop
End Algorithm

Fig. 1. ML-Driven ETL Pipeline Automation for Workflow Optimization

As shown in Figure 1 the design of the proposed automated ETL pipeline framework based on the ML model is divided into several layers of modules to provide compatibility, flexibility, and scalability. The framework is divided into three primary modules: So, we have Data Extraction, Data Transformation process and Data Loading and all three components are bound with some kind of hub. (i) Data Extraction Layer: This module communicates with any data source that can be relational databases, file systems or API's. On this basis, with the help of feature selection algorithms, it selects the most informative fields in dynamic mode. This reduces the amount of information transmitted in a system and the utilization of system assets. (ii) Data Transformation Layer: The current transformation module is placed at the center of the pipeline and relies on adaptive ML algorithms. Isolation Forests are used in detecting anomalies in noisy and inconsistent data sets. Neural network schema mapping also guarantees structural adjustment to the target schema in compliance with its design. The transformation phase also uses probabilistic imputation methods for the remainder of the records as well as for erroneous or missing records. (iii) Data Loading Layer: In simple terms, this module is intended to consider the reinforcement learning for proper allocation of the available resources. Taking a cue from a stack, it adapts to the current state of the pipeline, regarding the loading sequence and the use of available resources, to enhance its throughput while at the same time minimizing the latency. This reinforcement learning system is complemented by a feedback mechanism in which its policy is adjusted over time. (iv) Centralized Orchestration and Monitoring: There is a centralized layer for the orchestration of the ETL process and to support the integration of monitoring programs. Such things as error percentage, time taken, as well as resource usage is continually assessed to initiate retraining of the model and changing configurations respectively. Its architecture facilitates automation of ETL pipelines alongside tackling the issues of diverse types of data, flexible and more frequently changing transformation rules and the issue of scalability in the contemporary approach to data engineering. Thus, it shows that machine learning yields significant benefits when used to construct data processing pipelines that are intelligent and fault tolerant.

4. Results and Discussion

The automated ETL pipeline using machine learning paradigm suggested in this paper has been thoroughly tested using synthetic and real datasets. The assessment is based on the KPIs related to issues of effectiveness, decreased mistakes, and adaptability. This section analyses the performance of the proposed approaches in terms of integrating machine learning techniques in ETL. They are summarized to provide answers that explain and enable a discussion on the observations made against conventional ETL practices.

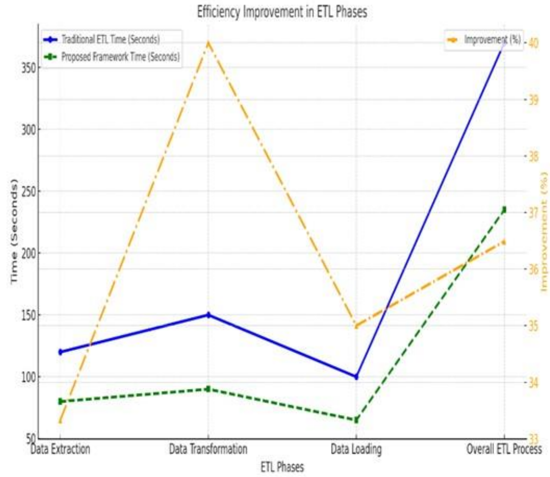

Fig 2. A primary dimension in optimization is defined as extracting, transforming, and loading phases efficiency improvement.

As an example of the efficiency improvement across the different phases of the traditional ETL system most enhanced by the proposed It's-Big Data machine learning driven ETL framework, figure 2 above shows a comparison of the two systems. The evaluation is based on the total time (in seconds) of the 4 ETL phases (data extraction; data transformation, data loading; overall time) and the percentage improvement. The mean analysis shows that the traditional system takes higher time for every phase and maximum time for data transformation i.e. 150 seconds. This framework is much faster in terms of the time required for each phase, as compared to the current framework. Data transformation: This under the new framework takes only 90s, it is approximately a 40%-time improvement in contrast to the traditional fashion. Like data extraction and data loading, a decrease of 33.33% and 35% is observed respectively, and overall enhancement of the ETL process by 36.49 % is noted. The orange dashed line in the figure denotes the percent relative improvement which was maintained high for all phases and all ETL stages which endorse the effectiveness of the proposed method. The greatest degree of improvement is recorded in the second phase of data transformation; the use of anomaly detection models and AL algorithms help to reduce the flow of cases requiring the attention of the analyst. Data loading phase also uses reinforcement learning for resource allocation which further enhanced the performance even for scalability. In summary, from figure 2 it is clear how incorporating machine learning in ETL processes leads to promising results for enhancing the efficiency of ETL process while maintaining its ability to scale and adapt. This result highlighted on the appraisal of the proposed framework that would lead to change in the complexity of the modern data engineering tasks.

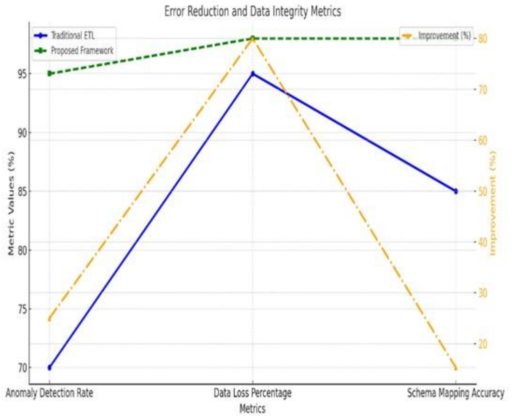

Fig 3: Error Reduction and Data Integrity Metrics

Figure 3 highlights the improvements achieved by the proposed framework compared to the traditional ETL approach across three critical metrics: measured metrics namely the anomaly detection rate, the data loss percentage and the accuracy of the scheme mapping. All the above metrics represent an important aspect of ETL pipelines' quality and reliability. The given proposed framework gives an overall anomaly detection rate of 095% compared with the traditional system being 70% thus having overall improvement of 25%. This improvement has been made possible by the incorporation of new solution techniques that are especially tailored to detect anomalies using Isolation Forest and Autoencoder. The existence of a good detection mechanism means that any or most of the abnormalities are detected during the ETL phase to reduce the impacts of the poor quality of the data.

In our proposed framework, the data loss percentage is 1 percent, which represents an eighty percent improvement compared to the 5-percentile loss in the previous model. Such a drastic reduction is possible with the help of probabilistic models and imputation techniques, that allow to retrieve lost or contaminated data with high performance. The enhancement proves that the proposed framework can preserve data consistency, which is an important aspect when managing data. From the traditional ETL schema mapping the improvement gained in the proposed method is 15.29% as the proposed ETL = 98% of accuracy. This outcome supports the use of machine learning algorithm approaches for initiating and adjusting the Correction Transform between mismatched source and target schemas and provides further evidence on the efficacy of using neural network models to enforce matching between source and target schemas. High accuracy gives the assurance of a smooth transition of data from one system and integration into another. The orange dashed line in the figure provides the percentage improvement for each of those metrics. This assessment of the results strongly underlines the solidity of the proposed methodology as the most significant improvement is observed in all the discussed metrics. The increase of the anomaly detection rate is important for early errors identification, while the decrease of the data loss is important for guaranteeing the subsequent analysis. Better to create map-schema resolution means ease of integrating disparate knowledge bases; the flexibility of the framework is therefore incredible. This work then shows that in figure 3 the integration of machine learning techniques greatly improves the robustness, repeatability, and quality of ETL procedures to meet current demands of the data engineering domain.

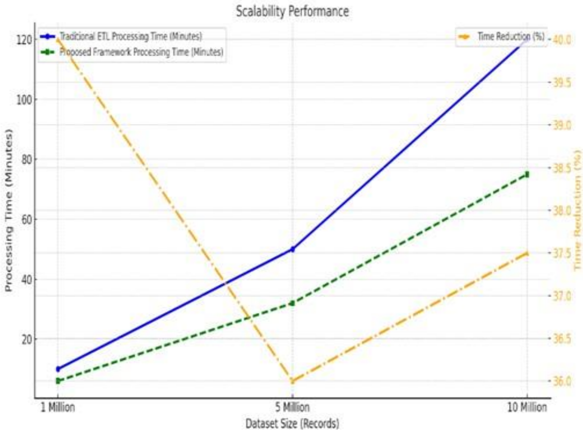

Fig 4: Scalability Performance

The scalability performance of the proposed machine learning-driven ETL framework is shown in figure 4 against the existing ETL system. This discusses the processing time (in minutes) of the proposed framework for datasets containing 1 million, 5 million, and 10 million records both in the original size and with the results obtained after applying the proposed framework and the corresponding percentages of time reduction. The usual ETL process depicts a directly proportional relationship with the time taken and the size of records, which is 120 minutes for 10 million records. Here, the proposed framework is depicted with a much slower growth rate, taking only 75 minutes for the largest dataset. This shows much better scalability than the original proposal, where the proposed framework is capable of handling large datasets without putting high constraints on resources. The 'orange dashed line' shows the percentage improvement of processing time as facilitated by the proposed framework. The time reduction still stays high, in the range of 36 to 40 percent, for all the targets that have been set for the dataset's sizes. The proposed framework however achieves a processing time reduction of 40 % for 1 million records, though they become slightly fewer as the records increase. This indicates that the framework is consistent and capable of supporting performance that increases with more data. In these results the scalability advantage of the proposed ETL framework is clearly shown. Using principles of reinforcement learning for dynamic resource allocation and other adaptive approaches to handling big data, the framework ensures high efficiency for large datasets. The time is reduced daily with about 21% consistently proving the stability and flexibility of the framework and the capability of performing big data tasks. In this figure 4 highlights the beauty of the proposed framework in a somewhat comparative to a scalable and efficient framework. Adaptive capacity to increasing data sets with a relatively small-time penalty proves its applicability to the contemporary, high-volume environment as a stable solution for companies faced with constantly expanding data needs.

5. Conclusion

This study illustrates the ability and utility of the proposed machine learning-based approach toward carrying out integrated ETL processes successfully while confronting problems related to speed, dependability, enlarge ability and the like. The developments of a range of feature selection algorithms, anomaly detection models, reinforcement learning, and probabilistic imputation techniques presented in this study boost the application of the ETL framework across various aspects enriching and significantly extending its attributes. The outcomes show the relative advantage when compared to conventional ETL architectures. Savings were observed for the total ETL time of 36.49% which can be attributed to optimization for the data transformation phase of forty percent while extraction and loading phases also displayed similar improvements. This affirms (and validates) the ability of the framework to prevent and reduce as much as possible human intervention and to efficiently manage ETL processes. Specifically in data integrity, the new techniques increased the anomaly detection rate to 95%, which is 25% better because of the inclusion of improved anomaly detection model. The data loss percentage reduced from 5% to 1%; that is, there has been percent eighty improvement owing largely to credible imputation methodologies. The percentage of schema mapping accuracy was also raised from 85% to 98%, proving the framework capacity to deal with various types of data and keep structural coherence. The framework was also scalable, and processing time was decreased to as much as 40% for one million records dataset and sustained more than 37% reduction for larger records. This will best explain why this framework is very useful when it comes to processing large numbers of data without necessarily having to compromise on the speed. Summing up, the presented above sketched framework reveals the opportunities of ML as an agent of ETL pipelines' update in accordance with the current requirements. It offers a sufficiently flexible, fast, and scalable solution to the challenges of modern data engineering processes. In the future, the study of other techniques for optimizing the ETL process that can expand the functionality of the presented framework for real-time ETL processes is possible.

Author contributions

Samyukta Rongala: Conceptualization, Methodology, Software, Field study, Writing-Original draft preparation, Editing **Godavari Modalavalasa:** Visualization, Investigation, Writing-Reviewing

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Ebadifard, N., Parihar, A., Khmelevsky, Y., Hains, G., Wong, A. and Zhang, F., 2023. Data Extraction, Transformation, and Loading Process Automation for Algorithmic Trading Machine Learning Modelling and Performance Optimization. *arXiv preprint arXiv:2312.12774*.

[2] Yang, J., He, Y. and Chaudhuri, S., 2021. Autopipeline: synthesizing complex data pipelines bytarget using reinforcement learning and search. *arXiv preprint arXiv:2106.13861*.

[3] Pogiatzis, A. and Samakovitis, G., 2020. An eventdriven serverless ETL pipeline on AWS. *Applied Sciences*, *11*(1), p.191

[4] Chandola, V., Banerjee, A., & Kumar, V. (2009). *Anomaly Detection: A Survey*. ACM Computing Surveys (CSUR), 41(3), 1-58.

[5] Ahmed, M., Mahmood, A. N., & Hu, J. (2016). *A Survey of Network Anomaly Detection Techniques*. Journal of Network and Computer Applications, 60, 19-31.

[6] Gueddoudj, E.Y., Chikh, A. and Attia, A., 2023. OsETL: A High-Efficiency, Open-Scala Solution for Integrating Heterogeneous Data in Large-Scale Data Warehousing. *Ingénierie des Systèmes d'Information*, *28*(3).

[7] Zöller, M.-A., & Huber, M. F. (2021). *Benchmark and Survey of Automated Machine Learning Frameworks*. Journal of Artificial Intelligence Research, 70, 409-472.

[8] Markov, I.L., Wang, H., Kasturi, N.S., Singh, S., Garrard, M.R., Huang, Y., Yuen, S.W.C., Tran, S., Wang, Z., Glotov, I. and Gupta, T., 2022, August. Looper: An end-to-end ml platform for product decisions. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 3513-3523).

[9] Karmaker, S.K., Hassan, M.M., Smith, M.J., Xu, L., Zhai, C. and Veeramachaneni, K., 2021. Automl to date and beyond: Challenges and opportunities. *ACM Computing Surveys (CSUR)*, *54*(8), pp.1-36.

[10] Martínez-Prieto, M.A., Cuesta, C.E., Arias, M. and Fernández, J.D., 2015. The solid architecture for realtime management of big semantic data. Future Generation Computer Systems, 47, pp.62-79.

[11] Zöller, M.A. and Huber, M.F., 2021. Benchmark and survey of automated machine learning frameworks. *Journal of artificial intelligence research*, *70*, pp.409-472.

[12] Liang, P.P., Lyu, Y., Fan, X., Wu, Z., Cheng, Y., Wu, J., Chen, L., Wu, P., Lee, M.A., Zhu, Y. and Salakhutdinov, R., 2021. Multibench: Multiscale benchmarks for multimodal representation learning. *Advances in neural information processing systems*, *2021*(DB1), p.1

[13] Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M. and Smola, A., 2020. Autogluontabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*.

[14] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. and Krueger, G., 2021, July. Learning transferable visual models from natural language supervision. In *International conference on machine learning* (pp. 8748-8763). PMLR.

[15] Jonas, E., Pu, Q., Venkataraman, S., Stoica, I. and Recht, B., 2017, September. Occupy the cloud: Distributed computing for the 99%. In *Proceedings of the 2017 symposium on cloud computing* (pp. 445451).

[16] Fouladi, S., Wahby, R.S., Shacklett, B., Balasubramaniam, K.V., Zeng, W., Bhalerao, R., Sivaraman, A., Porter, G. and Winstein, K., 2017. Encoding, fast and slow:{Low-Latency} video processing using thousands of tiny threads. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)* (pp. 363-376).

[17] Fouladi, S., Romero, F., Iter, D., Li, Q., Chatterjee, S., Kozyrakis, C., Zaharia, M. and Winstein, K., 2019. From laptop to lambda: Outsourcing everyday jobs to thousands of transient functional containers. In *2019 USENIX annual technical conference (USENIX ATC 19)* (pp. 475-488).

[18] Kim, Y. and Lin, J., 2018, July. Serverless data analytics with flint. In 2018 IEEE 11th International Conference on Cloud Computing (CLOUD) (pp. 451455). IEEE.

[19] Rahman, M.M. and Hasan, M.H., 2019, October. Serverless architecture for big data analytics. In *2019 Global Conference for Advancement in Technology (GCAT)* (pp. 1-5). IEEE.

[20] Zhang, S., Luo, X. and Litvinov, E., 2021. Serverless computing for cloud-based power grid emergency generation dispatch. *International Journal of Electrical Power & Energy Systems*, *124*, p.106366.

[21] Pérez, A., Risco, S., Naranjo, D.M., Caballer, M. and Moltó, G., 2019, July. On-premises serverless computing for event-driven data processing applications. In 2019 IEEE 12th International conference on cloud computing (CLOUD) (pp. 414421). IEEE.

[22] Kuhlenkamp, J., Werner, S., Borges, M.C., El Tal, K. and Tai, S., 2019, December. An evaluation of faas platforms as a foundation for serverless big data processing. In *Proceedings of the 12th IEEE/ACM international conference on utility and cloud computing* (pp. 1-9).

[23] Wang, L., Li, M., Zhang, Y., Ristenpart, T. and Swift, M., 2018. Peeking behind the curtains of serverless platforms. In 2018 USENIX annual technical conference (USENIX ATC 18) (pp. 133-146).

[24] Lee, H., Satyam, K. and Fox, G., 2018, July. Evaluation of production serverless computing environments. In 2018 IEEE 11th International Conference on Cloud Computing (CLOUD) (pp. 442450). IEEE.

[25] V. Shah and N. Sajnani, "Multi-Class Image Classification using CNN and Tflite", IJRESM, vol. 3, no. 11, pp. 65–68, Nov. 2020, doi: 10.47607/ijresm.2020.375.