Report for the Degree of Master of Computer Science

**End to End Automation in ETL Pipeline**



**Pikesh Maharjan**

**(LC00015002358)**

**Masters in Computer Science (MCS) IIMS college**

**Computer Science Department**

**Lincoln University, Malaysia**

**January 2025**

Research project for the Degree of Master of Computer Science

**End to End Automation in ETL Pipeline**

**Supervised by Prof. Sudan Jha, Ph.D.**

A report submitted in partial fulfilment of the requirements for the

degree of Master of Computer Science

**Pikesh Maharjan**

**(LC00015002358)**

**Masters in Computer Science (MCS) IIMS college**

**Computer Science Department**

**Lincoln University, Malaysia**

**January 2025**

List of Figures

Fig. 1.1 Illustration of ETL ........................................................................................ 5

Fig. 1.2 Illustration of Advanced ETL ........................................................................ 6

Fig. 3.1 Proposed Methodology for ETL automation .................................................. 18

Table of contents

[CHAPTER 1: INTRODUCTION 4](#_Toc1160452674)

[1.1 Background 6](#_Toc821259582)

[1.2 Statement of the Problem 7](#_Toc1584716983)

[1.3 Research Questions 9](#_Toc490625909)

[1.4 Objectives 9](#_Toc1193310145)

[1.5 Significance of the Study 10](#_Toc1263493825)

[1.6 Scope and Limitations of the Study 11](#_Toc522203415)

[CHAPTER 2: LITERATURE REVIEW 12](#_Toc1743848441)

[CHAPTER 3: METHODOLOGY 15](#_Toc98011457)

[3.1 Research Design 16](#_Toc993607962)

[3.2 Tools and Frameworks 16](#_Toc1264667727)

[3.3 Data Collection and Preparation 16](#_Toc87680662)

[3.4 System Architecture 17](#_Toc652342709)

[3.4.1 Data Ingestion 17](#_Toc672024904)

[3.4.2 Meta Data Identification 17](#_Toc1615387531)

[3.4.3 Configuration File Generation 17](#_Toc885591872)

[3.4.4 Human Interaction for Validation 17](#_Toc1031462215)

[3.4.5 Incorporating Transformation Logic 18](#_Toc292613263)

[3.4.6 Data Transformation and Cleaning 18](#_Toc1932359795)

[3.4.7 Data Loading 18](#_Toc2071342777)

[3.4.8 Automation and Scalability 18](#_Toc388742924)

[CHAPTER 4: REFERENCE 19](#_Toc1387167161)

# **CHAPTER 1: INTRODUCTION**

In today's digital transformation era, data has become an invaluable asset for organizations, aiding in decision-making and offering insights into customer behavior and market trends. However, the vast amount of information generated from various sources presents challenges in effectively analyzing, transforming, and utilizing this data. This is where the extraction, transformation, and loading (ETL) process come into play. ETL involves extracting data from multiple sources and recording or loading all the data into a centralized repository, such as data warehouses or data lakes. This process ensures that businesses can make informed, data-driven decisions by relying on consistent, accurate, and up-to-date information for their analytics with the help of data analysis and visualization of the analyzed data.

Extract, Transform and Load (ETL) is a process in data management and integration that plays an important role in digital and modern businesses to take informed decision based on the data [9]. It is used to develop and improve applications for analysis and thus making it easier to take data driven decisions. It involves three primary steps:

Diagram of a diagram of a process

Description automatically generated

*Fig. 1.1 Illustration of ETL [6]*

Extracting data from various sources involves transforming it into a suitable format and loading it into a central repository, like a data warehouse or data lakes, for analysis and decision-making with help of visualizations. This phase includes gathering data from diverse sources such as databases, APIs, files, or cloud storage and extracting data from these sources. These sources can vary in structure, including relational databases, spreadsheets, or unstructured web data, and can differ significantly from one another and thus may be structured, semi-structured or un-structured data.

The next step involves cleaning, formatting, and standardizing raw data from the sources collected in extract phase to align with business requirements and the repositories where the data will be loaded. This transformation process may include eliminating duplicate entries, addressing missing values, and converting the data into a format that satisfies business needs. Business may need to replace null values with some other default values, or some may completely ignore rows with null values. Thus, transformation phase differs to organization to organization.

Loading is the final step in the ETL process, where transformed data is loaded into systems like a Data Warehouse or Data Lake, making it accessible for users who need it for advanced analysis and visualization. This process ensures that data from various sources is prepared for analysis and visualization. The following figure illustrates the use of ETL in a diagram format.

A screen shot of a diagram

Description automatically generated *Fig. 1.2 Illustration of Advanced ETL [6]*

# **1.1 Background**

In today's world, where vast amounts of data are generated every second, ETL pipelines are essential for capturing, transferring, and loading this information. These pipelines are critical in data management and analysis, facilitating the movement and transformation of raw data that is subsequently loaded into databases, data warehouses, or other storage systems. The transformed data can then be turned into actionable insights for any data-driven organization.

Traditional ETL workflows are typically manual, necessitating human input in several steps like schema detection and classification. This reliance on manual processes can be time-consuming and increases the risk of errors. Such manual ETL methods may result in inconsistencies, problems, and delays that hinder the accurate and timely extraction of valuable insights from the data.

With the rapid increase in the volume and variety of data, there is an urgent need to automate ETL pipelines to reduce human effort. Automated ETL pipelines can manage vast amounts of data across many fields with little human intervention. This automation helps to reduce delays and minimizes potential issues that may arise from human errors.

Automated ETL processes utilize advanced technologies such as machine learning and artificial intelligence to handle tasks that once needed human intervention. Machine learning algorithms can identify and automatically adjust to changes in data schemas. For instance, they can recognize the characteristics of the data schema and apply them effectively. They also categorize data types and accurately map data fields. This not only accelerates the ETL process but also improves the accuracy and consistency of the results being processed.

Automated ETL pipelines have proven to be more flexible and adaptable to evolving business needs. They can easily scale to handle growing data volumes and can be quickly modified to integrate new data sources or adjust to changes in existing ones. This flexibility is crucial in today's fast-paced business landscape, where organizations must be agile and responsive to maintain a competitive edge.

In summary, automation of ETL pipelines is essential for modern data management and analytics. It addresses the challenges posed by traditional manual processes such as time consumption errors and scalability issues.

## **1.2 Statement of the Problem**

The manual creation of ETL processes has long been a challenge in data-driven industries, particularly given the rapid increase in data volume. Traditional workflows often demand significant human involvement for various tasks. These tasks encompass file ingestion, where raw data files are brought into the system; metadata recognition, which involves identifying and comprehending the structure and characteristics of the data; and mapping fields from source to destination to ensure proper alignment of data across different systems.

The reliance on manual processes leads to significant inefficiencies in both time and resource management. Human involvement is not only time-consuming but also increases the likelihood of errors, especially when handling data in various formats and levels. This can create data inconsistencies that undermine the integrity of the data and the insights drawn from it. Another labor-intensive aspect of manual ETL processes is the identification of data types across different datasets. As discussed in Chapter 3D, regulatory methods for classifying data that depend on predefined rules often lack the flexibility needed to adapt to new and evolving data types. These methods require considerable maintenance to keep pace with changes in data structures and formats, making them less efficient over time.

Despite the potential of machine learning to address these challenges, its application in the ETL domain remains limited. The complexity associated with implementing machine learning models for tasks like schema detection, data type categorization, and defining transformation logic has slowed their adoption. However, as machine learning technologies evolve and become more user-friendly, there is an increasing opportunity to utilize these tools to automate and enhance ETL processes.

Another important challenge is the absence of dynamic configuration systems that facilitate smooth integration with user-defined transformation logic. Whenever a new dataset is added, the pipeline code often requires modifications. This not only consumes a lot of time but also greatly restricts the flexibility of the ETL system. The necessity to manually tweak the code for each new dataset leads to delays and adds to the workload for data engineers, making the system less agile and less responsive to evolving data needs.

There is a notable gap in automating the final loading stage to target systems like SQL Server. This stage frequently requires repeated manual validations and corrections to ensure that the data is loaded correctly and meets the necessary standards. These manual interventions can result in inconsistencies and errors, which hinder the availability of data for analysis and decision-making. The absence of automation not only slows down the overall ETL process but also raises the risk of data inconsistencies, ultimately impacting the quality and reliability of the insights derived from the data.

The current state of data management heavily relies on specific ETL (Extract, Transform, Load) tools to carry out ETL processes. Many companies depend on commercial ETL software, which often comes with drawbacks like high costs, limited customization options, and vendor lock-in. Additionally, these tools frequently necessitate specialized knowledge and training, posing a challenge for smaller organizations or those with constrained resources.

Relying on these tools can complicate the adaptation to new data sources or evolving business needs. This, in turn, impacts the flexibility and scalability of ETL processes. For instance, when a new data source is added or an existing one undergoes significant changes, substantial modifications may be necessary for the ETL pipeline. This often requires reconfiguring the tool, updating scripts, and ensuring compatibility with new data formats, which can be both time-consuming and resource intensive [8].

Furthermore, the high costs associated with commercial ETL tools can be prohibitive for smaller organizations or startups. These costs include not only the initial price or subscription fees but also ongoing maintenance support and training expenses [8]. The financial burden can limit the ability of these organizations to invest in other critical areas like data analytics and business intelligence thereby hindering their overall growth and competitiveness.

Vendor lock-in is a significant concern when relying on specific tools and methods like GAP. Once an organization has made a substantial investment in a particular tool or software, transitioning to a different solution can be challenging and expensive. Additionally, it restricts the ability to tailor and enhance ETL processes to align with unique business requirements, as organizations often find themselves limited by the features and constraints of the selected tool.

This research seeks to address these issues by automating the ETL processes without the need of ETL Tools, implementing Machine learning for data type identification, dynamic configuration for user-driven transformation and loading data into the SQL Server.

## **1.3 Research Questions**

The study seeks to address the following questions:

1. How can machine learning methods be integrated to identify data types in ETL pipelines?
2. Is ETL possible without using ETL tools?
3. How can ETL be carried out with minimum human interaction?

## **1.4 Objectives**

This study has the objective of automating ETL pipeline and reducing as much as human intervention in ETL processing. It also aims to remove dependency on ETL tools for ETL processing paying high costs and a techie without knowledge of ETL tools can also process data and data load up to the SQL server. This study has been presented with general and specific objectives.

General Objective:

1. To design and implement an automated ETL pipeline capable of handling ASCII delimited datasets while ensuring accuracy.

Specific Objectives:

1. Develop mechanisms for data type recognition using ML methods.
2. Create a configuration file to allow human intervention when necessary.
3. Automate the data loading process into SQL Server.
4. Remove the dependency on ETL tools.
5. Reduce human intervention.
6. Ensure one can process data up to SQL server without ETL tool specific knowledge.
7. Increase ETL processing Speed.

## **1.5 Significance of the Study**

The automation of the complete ETL pipeline is set to revolutionize traditional data engineering methods. This research aims to reduce the reliance on human intervention in data processing, which in turn minimizes human errors and enhances data integration for better decision-making. By automating these processes, the study intends to create a framework for managing delimited file formats, focusing on accurately identifying data types and applying configurable transformation logic. These features can be utilized in real-world applications and align with current market trends, demonstrating that the research is both relevant and impactful.

This study aims to significantly enhance the ease and efficiency of ETL operations. As organizations produce and store vast amounts of data, the ability to manage this data with greater reliability and speed is essential. Automated ETL pipelines provide quicker and more accurate data processing than manual methods, as they reduce human errors through less human involvement. This ensures that data is readily available for analysis, reporting, and visualization. Improved efficiency will enable organizations to make faster, data-driven decisions, giving them a competitive edge in their industries.

This research aims to boost efficiency, offering significant benefits to data engineers and businesses by streamlining workflows and enhancing the overall quality and accessibility of data. Data engineers will have more time to focus on strategic initiatives that add value to the organization, as they will spend less time on repetitive manual tasks. Analysts will gain access to cleaner and more reliable data, allowing them to generate more accurate insights. Businesses will see improved data quality and faster decision-making processes, ultimately leading to better business outcomes. Additionally, the need to learn various ETL tools and their dependencies for the ETL process will be reduced, enabling engineers to concentrate on other important tasks.

The study aims to explore the current dependence on certain ETL tools, which frequently come with high expenses, restricted customization options, and vendor lock-in issues. By developing an automated ETL framework that is flexible and adaptable, this research provides organizations with a more affordable and customizable solution. This approach will alleviate the financial burden linked to commercial ETL technologies and enable organizations to customize their ETL processes to fit their specific needs.

## **1.6 Scope and Limitations of the Study**

This research focuses on creating and implementing an automated end-to-end ETL pipeline that includes automatic output transformation and loading. The study covers several key components designed to establish an efficient ETL process. It begins with the development of a script that allows users to input various details, such as the file path and connection information for the SQL server.

The project will also incorporate machine learning techniques to identify data types, enhancing the accuracy and reliability of the data type recognition process while reducing the need for human intervention. Many files contain numerous fields, and recognizing data types for all these fields typically demands significant human effort and time, which this study aims to eliminate

A key part of the research involves creating a configuration file that outlines metadata such as header delimiters and transformation logic, which can be adjusted manually if needed. This configuration file provides a flexible and customizable framework for defining the necessary transformations and data mapping. To maintain the accuracy and validity of the configuration, the system includes features that allow for human intervention, enabling users to validate or modify the configuration as needed. This capability ensures that users can review and adjust the configuration, guaranteeing the correct processing of data.

The study provides a comprehensive approach to automating ETL processes specifically for structured file formats, particularly delimited files. It primarily focuses on structured data formats, with minimal exploration of semi-structured or unstructured data formats, as the main goal is to develop a solution for managing structured data. The ongoing research will not address data ingestion and processing since the pipeline is tailored for batch processing scenarios. To ensure the validity and reliability of the research, artificial data will be utilized throughout the study for result analysis. This approach is designed to avoid the use of sensitive or proprietary data that could compromise the generalizability of the findings in specific industry contexts. Additionally, there will be no user interface; inputs will be provided directly through the script due to time constraints.

# **CHAPTER 2: LITERATURE REVIEW**

A literature review investigates the existing knowledge related to the study and projects carried out for automation of ETL (Extract, Transform, Load) pipelines. This examination is useful for studying the theoretical frameworks, methodologies, and practical applications presented in previous literature. Multiple studies have been carried out for the eradicating obstacles and progress in automating data integration and transformation processes, focusing on the increasing significance of scalable and efficient ETL systems within data-centric industries.

This paper concentrates on automating ETL process, where the study seeks to reduce manual intervention throughout the entire process i.e., extracting, transformation and loading. Additionally, this proposal addresses the demands for identifying data types of the multiple sources present in the source file by applying machine learning algorithms. The discourse encompasses various research efforts related to ETL automation, emphasizing the importance of creating efficient ETL pipelines capable of handling data processing in short intervals of time without or minimum human intervention.

The work by Embley et al. [10] highlights the potential of ontological conceptual modeling in addressing the challenges of unstructured data on the Web. By focusing on data-rich documents, their approach provides a framework for automating the extraction and structuring of information, which is particularly valuable in domains where data is abundant but difficult to query using traditional methods. This research contributes to the broader field of data extraction by offering a systematic and efficient method for transforming unstructured Web data into structured formats, enabling more effective data utilization and analysis [10]. Another Paper [1] suggests the role of machine learning in automating the ETL pipeline. Machine learning based data pre-processor is used to pre-process the data more rigorously. It reduces the processing time significantly and produces a good quality of data from the data warehouse [1]. An architecture is designed to address the challenge faced in the practical application of near real-time ETL processing [1]. Though it has suggested an architecture for automating ETL processes through the help of machine learning techniques, it hasn’t suggested exact methods and machine learning algorithms that can be used for ETL automation processes.

The paper [11] highlights the significance of improving ETL process system data flows to enhance business investment returns. It emphasizes the role of enterprise-level scheduling solutions that are user-friendly and capable of managing heterogeneous environments, paving the way for automation in ETL processes [11]. Moreover, the study envisions the development of future ETL tools with enhanced support for command-based or script-based automation, offering end-to-end process handling. By leveraging these advanced technologies, enterprises can achieve more efficient data integration, improve operational efficiency, and reduce manual errors in ETL execution [11].

Another paper [3] by Chiara Van der Putten focuses on automating the creation of workflows in SQL Server Integration Services (SSIS) to reduce manual efforts in data loading and integration processes. The research outlines a multi-stage approach, including the development of custom SSIS components, the creation of databases to store user automation needs and workflow modules, and the integration of advanced artificial intelligence models. A question-answering system leverages embedding-based methodologies and generative AI to interpret natural language user requests and generate tailored ETL workflows. The proof of concept demonstrates how AI-driven automation can streamline ETL processes, enabling organizations to handle complex data integration tasks more efficiently while allowing employees to focus on strategic initiatives [3].

The study by Pham [4] provides a valuable reference for understanding the challenges and approaches in automating ETL pipelines, particularly in addressing manual processes and enhancing data extraction capabilities. This thesis builds upon such insights by focusing on automating the entire ETL pipeline with minimal user intervention [4].

The work by Akisetti et al. [12] explores the integration of CI/CD pipelines into ETL workflows to address the inefficiencies of traditional ETL processes in machine learning applications. The study underscores the transformative potential of automating data preparation and processing to ensure a reliable and repeatable pipeline, facilitating seamless collaboration between data engineers and scientists. Similarly, this thesis builds on these principles by automating end-to-end ETL processes while addressing specific challenges such as error handling, real-time processing, and adaptability to user requirements [12].

Previous research has proposed the integration of SQL-based database systems for loading transformed data, with many emphasizing the use of SQL Server for its robust support for batch data processing and reporting capabilities. Additionally, best practices in ETL pipeline design, including modularity, scalability, and error handling, have been extensively documented.

In summary, the literature underscores the importance of designing ETL pipelines that are not only automated but also adaptive, efficient, and user-friendly. This review identifies gaps in practical implementations, such as handling dynamic configurations, using machine learning for data type identifications, automating without need of using ETL tools or any other CI CD pipeline which this research aims to address.

# **CHAPTER 3: METHODOLOGY**

# **3.1 Research Design**

This research follows a quantitative approach to evaluate the effectiveness of automating ETL pipelines using Python, Pandas, and machine learning for data type identification. The study is structured into three phases: system development, data processing, and performance evaluation.

1. System Development – This phase involves designing and implementing an ETL automation framework using Python and Pandas. A machine learning model is integrated to classify and identify data types from various sources, enabling dynamic schema generation and transformation.

2. Data Processing – The implemented system extracts raw data, applies transformations based on the detected data types, and loads structured data into the target system. The effectiveness of the ML model is assessed using a labeled dataset to validate its predictions.

3. Performance Evaluation – The system’s accuracy and efficiency are measured using standard metrics such as precision, recall, and execution time. An iterative approach is followed to refine the workflow, ensuring optimal automation and minimal manual intervention.

## **3.2 Tools and Frameworks**

The primary tools used in this research are Python and Pandas, with machine learning techniques to automate ETL pipeline development. The selected frameworks and libraries enable efficient data extraction, transformation, and schema identification. The key tools include:

Python – The core programming language for scripting ETL automation and integrating machine learning models.

Pandas – Used for data manipulation, transformation, and validation of structured and unstructured data.

Scikit-learn – Employed for training and implementing machine learning models for data type identification.

NumPy – Supports numerical operations, especially in handling large datasets efficiently.

## **3.3 Data Collection and Preparation**

The dataset used in this study consists of structured and semi-structured data sourced from various databases and files. The steps involved in data preparation are as follows:

Data Acquisition: Collect raw data from different sources such as CSV files, relational databases, and external APIs. Ensure diverse datasets to test the robustness of the ETL automation system.

Preprocessing: Handle missing values, inconsistencies, and outliers in the dataset using Pandas. Normalize and standardize data formats for consistency across different sources. Apply machine learning techniques to automatically identify data types and detect schema anomalies.

## **3.4 System Architecture**￼

The system architecture consists of the following components:

### **3.4.1 Data Ingestion￼**

The process starts with a data ingestion where users can input the location of the file path that needs to be processed. The user also inputs the destination server, database and table details to ensure the file is loading to the designated place where the users intend to load the file. The backend validates the file to ensure before proceeding to the next stage.

### **3.4.2 Meta Data Identification￼**

To identify Meta data such as column delimiter, row separators, and column data types, a combination of rule-based logic and machine learning techniques is used. The rule-based system examines predefined patterns to identify Meta data. Simultaneously, machine learning models are trained on sample datasets to predict Meta data with higher accuracy, particularly for those with ambiguous cases. This dual approach enhances reliability and adaptability in detecting various Meta data available in the source files.

### **3.4.3 Configuration File Generation￼**

Once the data types and other Meta data are identified, a configuration file is generated. This file provides detailed metadata about the dataset, including column names, data types, delimiters, and record separators. The configuration file acts as a blueprint for subsequent transformations and loading operations.

### **3.4.4 Human Interaction for Validation￼**

A provision is made for human users to review and modify the generated configuration file. This optional step is controlled via a check button on the frontend interface. If enabled, users can view the configuration file and adjust, such as adding specific transformation logic or correcting any discrepancies in data type detection. This step ensures greater flexibility and allows customization for unique use cases.

### **3.4.5 Incorporating Transformation Logic￼**

If additional transformation rules are required, users can specify them directly in the configuration file. For example, a user may need to standardize date formats, remove invalid rows, or aggregate data based on specific conditions. These transformations are dynamically parsed and applied to the dataset, ensuring that the data meets the desired quality and format standards before loading.

### **3.4.6 Data Transformation and Cleaning**

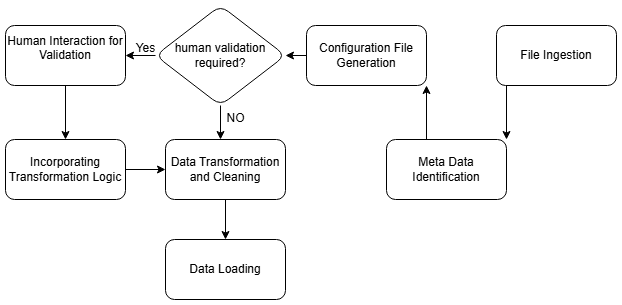
Using the instructions from the configuration file, the backend applies all specified transformations. These include renaming columns, handling missing values, normalizing data, and applying advanced transformations as defined by the user. This phase ensures that the dataset is prepared for seamless integration into the target database.

### **3.4.7 Data Loading**

The final step involves loading the transformed data into a SQL Server database. The loading process uses optimized techniques to handle large datasets efficiently. Indexing and batch operations are employed to minimize load time and enhance performance. The system also logs the loading process, capturing details such as the number of rows loaded, errors encountered, and time taken.

### **3.4.8 Automation and Scalability**

Although workflow orchestration tools like Apache Airflow are not implemented in the initial version, provisions are made to include automation for scheduling and monitoring tasks in future iterations. The current process can be run manually or triggered programmatically, ensuring flexibility in deployment.



*Fig. 3.1 Proposed Methodology for ETL*

# **REFERENCE**

[1] K. C. Mondal, N. Biswas, and S. Saha, "Role of Machine Learning in ETL Automation," in \*Proceedings of the 21st International Conference on Distributed Computing and Networking (ICDCN 2020)\*, Kolkata, India, Jan. 2020, pp. 1-6. Available: [(PDF) Role of Machine Learning in ETL Automation](https://www.researchgate.net/publication/339368691_Role_of_Machine_Learning_in_ETL_Automation)

[2] W. Yaddow, "Considerations for Automating Data Warehousing and ETL Tests," presented at the Datagaps, Feb. 2020. Available: [(PDF) Considerations for Automating Data Warehousing and ETL Tests](https://www.researchgate.net/publication/339384332_Considerations_for_Automating_Data_Warehousing_and_ETL_Tests)

[3] C. Van der Putten, "Transforming Data Flow: Generative AI in ETL Pipeline Automatization," M.S. thesis, Dept. Data Sci. and Eng., Politecnico di Torino, Turin, Italy, Apr. 2024. Available: [tesi.pdf](https://webthesis.biblio.polito.it/30855/1/tesi.pdf)

[4] P. Pham, "A Case Study in Developing an Automated ETL Solution – Concept and Implementation," Bachelor's thesis, Dept. Inf. and Commun. Technol., Turku Univ. of Appl. Sci., Turku, Finland, 2020. Available: <https://www.theseus.fi/bitstream/handle/10024/340208/Pham_Phuong.pdf?sequence=2>

[5] M. T. Maulik, "Automated ML ETL Pipeline of Electric Motor Temperature Sensor Data for Commercial Vehicles," Master's thesis, Dept. Data Sci. and Eng., Politecnico di Torino, Turin, Italy, Apr. 2024. Available:[https://github.com/maulikt04/Automated-ML-ETL-pipeline-of-electric-motor-temperature-sensor-data-for-commercial-vehicles-.](https://github.com/maulikt04/Automated-ML-ETL-pipeline-of-electric-motor-temperature-sensor-data-for-commercial-vehicles-.%20)

[6] Shopdev, "ETL Pipeline Using Snowflake," Shopdev, Dec. 16, 2022. [Online]. Available: [https://www.shopdev.co/blog/etl-pipeline-using-snowflake.](https://www.shopdev.co/blog/etl-pipeline-using-snowflake.%20)

[7] “Real-Time Data Warehousing.” Auckland University of Technology. [Online]. Available:<https://dsrc.aut.ac.nz/our-research/research-projects/real-time-data-warehousing.>

[8] L. Lucius, "What is ETL," presented at SlideServe, [Online]. Available[: https://www.slideserve.com/lucius/what-is-etl-powerpoint-ppt-presentation.](file:///C:\Users\pikesh.maharjan\AppData\Roaming\Microsoft\Word\:%20https:\www.slideserve.com\lucius\what-is-etl-powerpoint-ppt-presentation)

[9] "ETL," IBM. [Online].

Available: <https://www.ibm.com/think/topics/etl>.

[10] D. W. Embley, D. M. Campbell, Y. S. Jiang, S. W. Liddle, D. W. Lonsdale, Y. K. Ng, and R. D. Smith, "Conceptual-Model-Based Data Extraction from Multiple-Record Web Pages," Data and Knowledge Engineering, vol. 31, no. 3, pp. 227-251, 1999. Available: <https://pages.cs.wisc.edu/~smithr/pubs/dke99.pdf.>

[11] V. Radhakrishna, K. SravanKiran, and V. Ravikiran, "Automating ETL Process with Scripting Technology," in Nirma University International Conference on Engineering (NUiCONE), 2012. Available: <https://ieeexplore.ieee.org/document/6493198.>

[12] A. S. V. V. Akisetty, A. Kumar, M. M. K. Dandu, P. Goel, A. Jain, and A. Shrivastav, "Automating ETL Workflows with CI/CD Pipelines for Machine Learning Applications," in Nirma University International Conference on Engineering (NUiCONE), 2012, pp. 1-4. [Online]. Available: <https://www.irejournals.com/formatedpaper/1705069.pdf.>

[13] W. Qu, V. Basavaraj, S. Shankar, and S. Dessloch, “Real-Time Snapshot Maintenance with Incremental ETL Pipelines in Data Warehouses,” in Big Data Analytics and Knowledge Discovery, Springer, 2015 Available: <https://link.springer.com/chapter/10.1007/978-3-319-22729-0_18>.

[14] V. Radhakrishna, K. SravanKiran, and V. Ravikiran, “Automating ETL process with scripting technology,” in Nirma University International Conference on Engineering (NUiCONE), IEEE, 2012, pp. 1–4. Available: <https://ieeexplore.ieee.org/document/6493192>. Last accessed: Nov. 26, 2024.

[15] F. Sebastiani, “Machine learning in automated text categorization,” ACM Computing Surveys (CSUR), vol. 34, no. 1, pp. 1–47, 2002. Available: <https://dl.acm.org/doi/10.1145/505282.505283>.

[16] D. Skoutas and A. Simitsis, “Designing ETL processes using semantic web technologies,” in Proceedings ACM 9th International Workshop on Data Warehousing and OLAP (DOLAP 2006), Arlington, Virginia, USA, 2006 Available: <https://dl.acm.org/doi/10.1145/1183346.1183358>.

[17] S. Suresh, J. P. Gautam, G. Pancha, F. J. DeRose, and M. Sankaran, “Method and architecture for automated optimization of ETL throughput in data warehousing applications,” US Patent 6,208,990, 2001. Available: <https://patents.google.com/patent/US6208990B1/en>.

[18] M. N. Tho and A. M. Tjoa, “Zero-latency data warehousing for heterogeneous data sources and continuous data streams,” in 5th International Conference on Information Integration and Web-based Applications Services, 2003, pp. 55–64. Available: <https://ieeexplore.ieee.org/document/1241161>

[19] V. Tziovara, P. Vassiliadis, and A. Simitsis, “Deciding the physical implementation of ETL workflows,” in Proceedings of the ACM Tenth International Workshop on Data Warehousing and OLAP, ACM, 2007, pp. 49–56. Available: <https://dl.acm.org/doi/10.1145/1317331.1317341>.

[20] P. Vassiliadis, “A Survey of Extract - Transform - Load Technology,” International Journal of Data Warehousing and Mining, vol. 5, no. 3, pp. 1–27, 2009.Available: <https://www.igi-global.com/article/survey-extract-transform-load-technology/37403>

[21] P. Vassiliadis and A. Simitsis, “Near Real Time ETL,” Springer Annals of Information Systems, vol. 3, no. 978-0-387-87430-2, 2008. Available: <https://link.springer.com/chapter/10.1007/978-0-387-87431-9_3>.

[22] P. Vassiliadis and A. Simitsis, “Extraction, transformation, and loading,” in Encyclopedia of Database Systems, Springer, 2009, pp. 1095–1101. Available: <https://link.springer.com/referenceworkentry/10.1007/978-0-387-39940-9_110>.

[23] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos, “On the Logical Modeling of ETL Processes,” in Proc. International Conference on Advanced Information Systems Engineering, 2002, pp. 782–786. Available: <https://link.springer.com/chapter/10.1007/3-540-47919-1_53>.

[24] H. Zhou, D. Yang, and Y. Xu, “An ETL strategy for real-time data warehouse,” in Practical Applications of Intelligent Systems, Springer, 2011, pp. 329–336. Available: <https://link.springer.com/chapter/10.1007/978-3-642-25658-7_40>.