Int. j. adv. multidisc. res. stud. 2025; 5(2):106-111

Received: 14-01-2025
Accepted: 24-02-2025

International Journal of Advanced Multidisciplinary Research and Studies

ISSN: 2583-049X

A Systematic Taxonomy of ETL Activities for Modern Data Pipelines

Basheer Mohammed
Consultant, TCS, Telangana, India

DOI: https://doi.org/10.62225/2583049X.2025.5.2.3816

Corresponding Author: **Basheer Mohammed**

Abstract

ETL (Extract, Transform, Load) is a critical process in data management, enabling organizations to extract data from multiple sources, transform it into a usable format, and load it into target systems for analysis and reporting. However, as data pipelines become more complex with advancements in cloud computing, big data, and real-time processing, a structured framework for understanding ETL activities is essential. This article presents a systematic taxonomy of ETL activities, categorizing them into extraction, transformation, loading, and supporting processes. By providing a well-defined classification, this taxonomy helps data engineers, architects, and businesses optimize their ETL workflows, improve efficiency, and enhance data governance in modern data ecosystems.

Keywords: ETL (Extract, Transform, Load); Streaming ETL, Change Data Capture (CDC), ETL Taxonomy, Cloud-Based ETL, AI-Driven ETL

1. Introduction

Extract, Transform, Load (ETL) is a fundamental data integration process that enables organizations to collect data from multiple sources, apply necessary transformations, and load it into a target system such as a data warehouse, data lake, or analytics platform. ETL plays a crucial role in data warehousing, analytics, and business intelligence (BI) by ensuring data consistency, accuracy, and accessibility for decision-making. Effective ETL workflows allow businesses to consolidate structured and unstructured data, enabling advanced analytics, reporting, and machine learning applications.

1.1 Challenges in Managing Complex ETL Workflows

As modern data ecosystems evolve, traditional ETL workflows face significant challenges, including:

- **Scalability:** With the explosion of big data, ETL processes must handle large volumes of data efficiently.
- **Real-time Processing:** Organizations increasingly demand real-time or near-real-time data ingestion, making batch-based ETL insufficient in many cases.
- **Data Quality and Governance:** Ensuring data accuracy, consistency, and compliance with regulatory standards is a growing concern.
- **ETL Performance Optimization:** Optimizing ETL jobs to minimize processing time and system resource consumption is essential for maintaining operational efficiency.
- **Complex Data Transformations:** As data sources become more diverse (e.g., APIs, streaming data, unstructured data), transformation logic becomes more sophisticated and difficult to manage.

1.2 Motivation for Developing an ETL Taxonomy

Given these challenges, a structured taxonomy of ETL activities can help streamline process understanding, optimization, and automation. By classifying ETL activities into well-defined categories, organizations can:

- Standardize ETL design and development across teams.
- Improve efficiency by identifying reusable components and best practices.
- Enhance automation efforts by mapping activities to AI-driven or low-code/no-code ETL solutions.
- Facilitate troubleshooting, monitoring, and optimization by clearly defining process stages.

2. Methodology for Taxonomy Development

The taxonomy of ETL activities in this study is developed based on three primary classification criteria:

- **Functionality-Based Classification:** ETL processes are divided into distinct categories based on their role in data integration workflows—extraction, transformation, loading, and supporting activities.
- **Execution Stage:** Each activity is mapped to its stage in the ETL pipeline, ensuring a logical flow from data ingestion to final storage.
- **Industry Best Practices:** The classification aligns with widely accepted ETL frameworks used in enterprise data warehousing, cloud data platforms, and big data processing.

By combining these three perspectives, the taxonomy provides a structured yet flexible framework that accommodates traditional ETL as well as modern ELT (Extract, Load, Transform) methodologies.

2.1 Sources Considered for Taxonomy Development

The proposed taxonomy is based on insights gathered from multiple sources:

- **Academic Research:** Existing literature on ETL workflows, data integration techniques, and data engineering principles.
- **Industry Frameworks:** Best practices from leading data warehousing and ETL frameworks such as Kimball's dimensional modeling, Inmon's data warehousing architecture, and Data Vault modeling.
- **Case Studies & Real-World Implementations:** Practical observations from enterprise ETL implementations across various industries, including finance, healthcare, e-commerce, and cloud computing.
- **Modern ETL & Data Engineering Trends:** Insights from cloud-based ETL tools (e.g., AWS Glue, Google Cloud Dataflow, Azure Data Factory) and real-time data integration platforms (e.g., Apache Kafka, Spark Streaming).

By synthesizing information from these sources, the taxonomy ensures a balance between theoretical rigor and practical applicability.

2.2 Alignment with Modern Data Engineering Practices

The taxonomy is designed to align with contemporary data engineering trends, including:

- **Cloud-Native ETL & ELT:** Supports modern data pipelines that leverage cloud-based storage, computing, and serverless ETL processes.
- **Real-Time & Streaming Data Processing:** Incorporates event-driven data ingestion and processing for real-time analytics.
- **Automation & AI-Driven ETL:** Accounts for machine learning-powered data transformations, anomaly detection, and automated data quality checks.
- **Data Governance & Compliance:** Integrates security, lineage tracking, and regulatory compliance considerations into the ETL workflow.

By structuring ETL activities within this taxonomy, organizations can achieve greater consistency, efficiency, and scalability in their data integration processes.

3. A Systematic Taxonomy of ETL Activities

This section presents a structured classification of ETL activities, categorizing them into extraction, transformation, loading, and supporting activities. This taxonomy helps organizations standardize their ETL workflows, optimize performance, and ensure data quality and governance.

3.1 Extraction Activities

The extraction phase is responsible for gathering data from various sources and preparing it for transformation. Key activities include:

- **Data Source Identification:** Determining and cataloging data sources such as relational databases, APIs, flat files, cloud storage, or streaming sources.
- **Data Connection and Access:** Establishing secure and efficient connections to data sources using protocols such as JDBC, ODBC, REST APIs, or message queues.
- **Data Ingestion Strategies (Batch, Real-Time, Streaming):** Choosing an ingestion method based on business needs:
  - **Batch Processing:** Extracting large volumes of data at scheduled intervals.
  - **Real-Time Processing:** Capturing data changes continuously for low-latency analytics.
  - **Streaming:** Processing event-driven data using tools like Apache Kafka, Spark Streaming, or AWS Kinesis.
- **Data Quality Checks at Source:** Performing initial data validation to detect missing, inconsistent, or erroneous records before ingestion.

3.2 Transformation Activities

The transformation phase involves cleaning, enriching, and restructuring data to ensure consistency, accuracy, and usability. Key activities include:

- **Data Cleansing and Standardization:** Removing duplicates, correcting inconsistencies, filling missing values, and ensuring uniform formats.
- **Data Enrichment and Augmentation:** Enhancing data by integrating additional attributes from other sources, such as third-party APIs or reference datasets.
- **Data Aggregation and Summarization:** Consolidating raw data into meaningful summaries, such as total sales per region or average customer transaction values.
- **Schema Mapping and Data Type Conversions:** Aligning source data structures with target schema requirements and converting data types for compatibility.
- **Business Rule Implementations:** Applying predefined rules to derive new values, filter records, or enforce calculations based on organizational policies.
- **Data Masking and Anonymization:** Protecting sensitive information (e.g., PII, financial data) by applying encryption, tokenization, or obfuscation techniques.

3.3 Loading Activities

The loading phase moves transformed data into target storage systems, ensuring efficiency, integrity, and performance. Key activities include:

- **Destination System Selection (Data Warehouse, Data Lake, Database):** Determining the appropriate storage solution based on analytical needs and data volume.
- **Loading Strategies (Incremental, Full Load, Change Data Capture)**
  - **Incremental Load:** Transferring only new or modified data to improve efficiency.

- **Full Load:** Replacing entire datasets in target systems, often used for initial migrations.
- **Change Data Capture (CDC):** Tracking and applying changes dynamically in real time.

- **Error Handling and Rollback Mechanisms:** Implementing checks to identify failed transactions and ensuring rollback procedures to maintain data integrity.
- **Indexing and Performance Optimization:** Applying indexing strategies, partitioning, and caching mechanisms to accelerate data retrieval and query performance.

3.4 Supporting Activities

Supporting activities ensure ETL pipelines are reliable, efficient, and compliant with security and governance requirements. Key activities include:

- **Logging and Monitoring:** Capturing logs, tracking ETL execution metrics, and setting up alerts to detect failures or performance bottlenecks.
- **Data Lineage and Metadata Management:** Maintaining a record of data transformations, source-to-target mappings, and metadata to enhance traceability and compliance.
- **Performance Optimization and Parallel Processing:** Implementing parallelization, workload balancing, and memory management techniques to optimize ETL execution.
- **Error Handling and Recovery Mechanisms:** Developing failover strategies, retry mechanisms, and contingency plans for system failures or data inconsistencies.
- **Security and Compliance Considerations:** Ensuring adherence to data protection regulations (e.g., GDPR, CCPA), implementing role-based access controls, and securing ETL workflows against cyber threats.

4. ETL Taxonomy in Modern Data Pipelines

As organizations transition from traditional data warehousing to modern cloud-based architectures, ETL processes must evolve to meet new challenges and opportunities. This section explores how the proposed ETL taxonomy aligns with contemporary data engineering practices, including cloud-based ETL, ELT workflows, automation, AI-driven pipelines, and scalability considerations.

4.1 Application of ETL Taxonomy in Cloud-Based ETL Solutions

Cloud computing has transformed ETL by enabling scalable, on-demand data processing with fully managed services. Cloud-based ETL solutions, such as AWS Glue, Google Cloud Dataflow, and Azure Data Factory, leverage the same fundamental ETL activities but introduce new paradigms:

- **Serverless ETL:** Cloud platforms manage infrastructure, allowing users to focus on data transformation logic rather than hardware provisioning.
- **Distributed Processing:** ETL jobs run on distributed frameworks (e.g., Apache Spark, Google BigQuery) to handle large-scale datasets efficiently.
- **Integration with Cloud Storage:** Instead of traditional relational databases, ETL now integrates with data lakes (e.g., Amazon S3, Azure Data Lake Storage) for flexible data storage.
- **Cost-Efficiency:** Pay-as-you-go pricing models optimize resource utilization based on workload demand.

Despite these advantages, cloud ETL solutions must address data security, compliance, and governance to ensure proper handling of sensitive data across different cloud providers.

4.2 Comparison: Traditional ETL vs. Modern ELT (Extract, Load, Transform)

While traditional ETL has been the standard for decades, many modern architectures now favor the ELT (Extract, Load, Transform) approach:

Feature	Traditional ETL	Modern ELT
Processing Location	Data is transformed before loading into the target system	Raw data is loaded first, then transformed within the target system
Computing Resource	Requires a dedicated ETL server or on-premise infrastructure	Uses cloud-native computing power for on-demand transformations
Data Storage	Structured data is stored in a data warehouse	Both structured and unstructured data can reside in a data lake before transformation
Performance	Can be limited by on-premise infrastructure	Leverages distributed cloud-based processing for scalability
Flexibility	Schema must be defined before data is loaded	Schema-on-read allows for flexible data exploration and transformation

4.2.1 ETL vs. ELT Trade-offs

- **ETL is better suited for structured, compliance-heavy data workflows** (e.g., finance, healthcare) where strict data governance is required before loading.
- **ELT is ideal for big data and analytics applications**, where raw data can be loaded into cloud warehouses (e.g., Snowflake, BigQuery) and transformed dynamically as needed.

Both approaches rely on the same fundamental ETL activities, but the sequencing and execution environments differ.

4.3 Impact of Automation and AI-Driven Data Pipelines on ETL Activities

Advancements in AI and automation are reshaping ETL by reducing manual effort, improving efficiency, and enhancing data quality. Key innovations include:

- **AI-Powered Data Cleansing & Anomaly Detection:** Machine learning models detect and correct data inconsistencies, missing values, and anomalies in real time.
- **Automated Schema Detection & Mapping:** AI-driven ETL tools can infer schema changes, suggest mappings, and automate schema evolution for dynamic datasets.
- **Metadata-Driven Pipelines:** Automated metadata management improves data lineage tracking, version control, and governance.
- **Low-Code & No-Code ETL Tools:** Platforms like Informatica, Talend, and dbt enable business users to design ETL workflows with minimal coding,

democratizing data integration.

As AI adoption grows, ETL processes are shifting from rule-based transformations to intelligent, self-optimizing pipelines that adapt based on real-time insights.

4.4 Scalability and Real-Time Data Processing Challenges

Modern data pipelines must handle high-velocity, high-volume data while ensuring low-latency processing. The ETL taxonomy must adapt to:

- **Scalability Challenges:**
  - Distributed ETL frameworks like Apache Spark enable parallel processing of large datasets.
  - Microservices-based ETL architectures decouple data processing steps for modular scalability.
- **Real-Time Data Processing:**
  - Streaming ETL tools like Apache Kafka, Flink, and Spark Streaming allow continuous ingestion and transformation.
  - Event-driven architectures process data as it arrives, reducing latency for real-time analytics.
- **Balancing Latency vs. Data Accuracy:**
  - Some real-time pipelines perform minimal transformations (ELT-like approach) to reduce latency, while others implement mini-batch processing to ensure data accuracy.

The proposed ETL taxonomy remains relevant in modern data pipelines, whether for traditional on-premise ETL, cloud-based ELT, or AI-driven automation. By structuring ETL activities into extraction, transformation, loading, and supporting processes, organizations can optimize performance, improve data quality, and scale efficiently in today’s data-driven world.

5. Case Study / Example Implementation

To illustrate the practical application of the proposed ETL taxonomy, this section presents a real-world case study of a global e-commerce company optimizing its ETL workflows for data analytics, operational intelligence, and business decision-making.

5.1 Background and Business Problem

A global e-commerce retailer collects vast amounts of data from multiple sources, including:

- **Customer transactions** (e.g., website and mobile app purchases)
- **Product catalog updates** (e.g., inventory levels, supplier data)
- **Marketing and user engagement data** (e.g., clickstream analytics, ad performance)
- **Logistics and fulfillment data** (e.g., shipping, returns, delivery tracking).

The company initially relied on a traditional ETL process running on an on-premises data warehouse. However, as business operations expanded, they faced several challenges:

- **Scalability Issues:** The ETL process struggled to handle increasing data volumes, leading to delayed reports.
- **High Processing Costs:** Running ETL workloads on dedicated servers was expensive and inefficient.
- **Limited Real-Time Capabilities:** Decision-making relied on batch processing, making it difficult to react to real-time trends (e.g., fraud detection, dynamic pricing).
- **Data Quality & Governance Issues:** Inconsistent data formats and missing records led to unreliable analytics.

To address these challenges, the company migrated to a cloud-based ETL and ELT approach, leveraging the proposed ETL taxonomy to standardize and optimize workflows.

5.2 Applying the ETL Taxonomy to Optimize Workflows

The company restructured its ETL pipeline using the taxonomy framework, aligning each activity to the modern cloud-based environment:

5.2.1 Extraction Activities

- **Data Source Identification:** Cataloged structured (SQL databases) and unstructured (JSON logs, APIs) data sources.
- **Data Connection & Access:** Used AWS Glue and Amazon Kinesis to ingest data from different systems.
- **Ingestion Strategies:**
  - Transactional data (orders, payments): Processed in near real-time via Kafka streaming.
  - Product and inventory updates: Scheduled batch jobs.
  - User activity logs: Streamed continuously to a data lake (Amazon S3).
- **Data Quality Checks at Source:** Implemented schema validation and deduplication before ingestion.

5.2.2 Transformation Activities

- **Data Cleansing & Standardization:** Used AWS Glue and dbt to remove duplicates, correct inconsistent formats, and enforce data integrity rules.
- **Data Enrichment & Augmentation:** Integrated external pricing APIs and customer demographics to enhance product analytics.
- **Schema Mapping & Data Type Conversion:** Standardized formats across different systems, ensuring compatibility with Snowflake (data warehouse).
- **Business Rule Implementation:** Applied dynamic pricing algorithms and customer segmentation rules for targeted marketing.
- **Data Masking & Anonymization:** Protected sensitive PII (personally identifiable information) using encryption and tokenization.

5.2.3 Loading Activities

- **Destination System Selection:**
  - Operational dashboards: Data loaded into Snowflake for BI reports.
  - Historical data: Stored in Amazon S3 for archival and machine learning use cases.
- **Loading Strategies:**
  - Incremental Load: Updated only modified records in Snowflake using Change Data Capture (CDC).
  - Full Load: Weekly batch processing for large analytical models.
- **Error Handling & Rollback Mechanisms:** Implemented **d**ata validation rules to prevent corrupt data from entering analytical systems.
- **Indexing & Performance Optimization:** Used clustering keys and caching in Snowflake to accelerate query performance.

5.2.4 Supporting Activities

- **Logging & Monitoring:** Integrated AWS CloudWatch and Datadog to track ETL performance and detect anomalies.
- **Data Lineage & Metadata Management:** Used Apache Atlas to maintain end-to-end lineage tracking.
- **Performance Optimization:** Applied parallel processing and auto-scaling in Spark-based transformations.
- **Security & Compliance:** Ensured GDPR & CCPA compliance by implementing role-based access control (RBAC) and data masking.

5.3 Results & Benefits of Using the ETL Taxonomy

By aligning ETL processes with the taxonomy framework, the company achieved significant improvements:

Challenge	Solution using ETL Taxonomy	Benefit
Scalability Issues	Migrated ETL to a cloud-based, distributed architecture	Improved data processing speed and flexibility
High Processing Costs	Switched to serverless ETL (AWS Glue, Snowflake ELT)	Reduced infrastructure costs by 40%
Limited Real-Time Capabilities	Adopted Kafka for streaming ETL	Enabled real-time fraud detection & dynamic pricing
Data Quality Issues	Implemented automated cleansing & validation rules	Enhanced data accuracy and trustworthiness
Complex Maintenance	Standardized workflows with metadata management	Improved ETL maintainability & governance

This case study demonstrates how a systematic ETL taxonomy can transform data workflows, enabling businesses to operate at scale while maintaining high-quality, reliable, and timely data insights.

6. Conclusion

The systematic taxonomy of ETL activities proposed in this paper provides a structured approach to understanding, optimizing, and automating modern ETL workflows. By categorizing ETL processes into extraction, transformation, loading, and supporting activities, the taxonomy helps data engineers, architects, and businesses standardize and streamline their data pipelines.

Through the case study, we demonstrated how applying this taxonomy to a real-world ETL implementation led to improved efficiency, scalability, and data quality, particularly in cloud-based and AI-driven environments. Organizations leveraging this framework can enhance maintainability, optimize costs, and ensure data reliability in their analytics and business intelligence workflows.

As ETL continues to evolve, emerging trends like AI-driven optimizations, serverless architectures, real-time processing, and enhanced data governance will shape the future of data integration. Businesses that adopt taxonomy-driven ETL strategies will be better positioned to navigate these advancements, ensuring their data pipelines are agile, scalable, and future-proof.

Ultimately, this taxonomy serves as a foundation for further research and innovation, paving the way for more automated, intelligent, and resilient data integration frameworks in the ever-growing world of data engineering.

7. Conflict of Interest

None.

8. References

1. Van der Putten Chiara. Transforming data flow: Generative AI in ETL pipeline automatization. PhD diss., Politecnico di Torino, 2024.
2. Simitsis Alkis, Spiros Skiadopoulos, Panos Vassiliadis. The history, present, and future of ETL technology. In DOLAP, 2023, 3-12.
3. Seenivasan Dhamotharan. ETL (extract, transform, load) best practices. International Journal of Computer Trends and Technology. 2023; 71(1):40-44.
4. Raj Aiswarya, Jan Bosch, Helena Holmström Olsson, Tian J Wang. Modelling data pipelines. In 2020 46th Euromicro conference on software engineering and advanced applications (SEAA), pp. 13-20. IEEE, 2020.
5. Zarate Gorka, María Jose Lopez Osa, Ana I Torre-Bastida, Urtza Iturraspe, Jordi Arjona, Benjamín Navarro, *et al*. Evolution of Extract-Transform-Load (ETL) processes towards data product pipelines. In Proceedings of the 4th Eclipse Security, AI, Architecture and Modelling Conference on Data Space, 2024, 25-32.
6. Shojaee Rad, Zahra, Mostafa Ghobaei-Arani. Data pipeline approaches in serverless computing: A taxonomy, review, and research trends. Journal of Big Data. 2024; 11(1):82.
7. Walha Afef, Faiza Ghozzi, Faiez Gargouri. Data integration from traditional to big data: Main features and comparisons of ETL approaches. The Journal of Supercomputing. 2024; 80(19):26687-26725.
8. Pham Phuong. A case study in developing an automated ETL solution: concept and implementation, 2020.
9. Seenivasan Dhamotharan. Improving the Performance of the ETL Jobs. International Journal of Computer Trends and Technology. 2023; 71(3):27-33.
10. Schejtman Nicolas. Semantic ETL. An extract, transform, load pipeline implementation using semantic technologies, 2020.
11. Gueddoudj, El Yazid, Azeddine Chikh, Abdelouahab Attia. Os-ETL: A High-Efficiency, Open-Scala Solution for Integrating Heterogeneous Data in Large-Scale Data Warehousing. Ingénierie des Systèmes d'Information. 2023; 28(3).
12. Trajkovska Aneta, Tome Dimovski, Ramona Markoska, Zoran Kotevski. Automation and Monitoring on Integration ETL Processes while Distributing Data, 2023, 212-219.
13. Mohamed Azlinah, Maryam Khanian Najafabadi, Yap Bee Wah, Ezzatul Akmal Kamaru Zaman, Ruhaila Maskat. The state of the art and taxonomy of big data analytics: View from new big data framework. Artificial intelligence review. 2020; 53:989-1037.
14. Sikeridis Dimitrios, Ioannis Papapanagiotou, Bhaskar Prasad Rimal, Michael Devetsikiotis. A Comparative taxonomy and survey of public cloud infrastructure vendors, 2017. arXiv preprint arXiv:1710.01476.
15. Seenivasan Dhamotharan. Optimizing Cloud Data Warehousing: A Deep Dive into Snowflakes Architecture and Performance. International Journal of Advanced Research in Engineering and Technology.

2021; 12(3).
16. Ozyurt Ibrahim Burak, Jeffrey S Grethe. Foundry: A message-oriented, horizontally scalable ETL system for scientific data integration and enhancement. Database 2018: bay130, 2018.
17. Mondal Kartick Chandra, Neepa Biswas, Swati Saha. Role of machine learning in ETL automation. In Proceedings of the 21st international conference on distributed computing and networking, 2020, 1-6.
18. Karagiannis Anastasios, Panos Vassiliadis, Alkis Simitsis. Scheduling strategies for efficient ETL execution. Information Systems. 2013; 38(6):927-945.
19. El Hafyani Hafsa, Mohammad Abboud, Yehia Taher. A microservices based architecture for implementing and automating etl data pipelines for mobile crowdsensing applications. In 2021 IEEE international conference on big data (Big Data), pp. 5909-5911. IEEE, 2021.
20. Seenivasan Dhamotharan. Real-time data processing with streaming ETL. International Journal of Science and Research. 2023; 12(11):1-10.
21. Uddin, Md Kazi Shahab, Kazi Md Riaz Hossan. A Review of Implementing AI-Powered Data Warehouse Solutions to Optimize Big Data Management and Utilization. Academic Journal on Business Administration, Innovation & Sustainability. 2024; 4(3):10-69593.
22. Nambiar Athira, Divyansh Mundra. An overview of data warehouse and data lake in modern enterprise data management. Big data and cognitive computing. 2022; 6(4):132.
23. Oliveira Bruno, Orlando Belo. ETL standard processes modelling-a novel BPMN approach. In International Conference on Enterprise Information Systems, vol. 2, pp. 120-127. SCITEPRESS, 2013.
24. Sheta Sagar Vishnubhai. A Comprehensive Analysis of Real-Time Data Processing Architectures for High-Throughput Applications, 2022.
25. Khan Muhammad Jahanzeb, Ruoyu Wang, Daniel Sun, Guoqiang Li. Data Provenance Based System for Classification and Linear Regression in Distributed Machine Learning. In International Workshop on Structured Object-Oriented Formal Language and Method, pp. 279-295. Cham: Springer International Publishing, 2019.
26. Ataei Pouya, Alan Litchfield. The state of big data reference architectures: A systematic literature review. IEEE Access. 2022; 10:113789-113807.
27. Seenivasan Dhamotharan. Critical Security Enhancements for ETL Workflows: Addressing Emerging Threats and Ensuring Data Integrity. International Journal of Innovative Research in Computer and Communication Engineering, 2024:1301-1313.
28. Anchoori Santoshkumar. Auto-scaling Distributed ETL Systems with Serverless Platforms. System 250, no. 500: 100.
29. Otoo-Arthur David, Terence L van Zyl. A scalable heterogeneous big data framework for e-learning systems. In 2020 international conference on artificial intelligence, big data, computing and data communication systems (icABCD), 2020, 1-15. IEEE.