ISSN 2581 – 3781

AJCSE
ASIAN JOURNAL OF COMPUTER SCIENCE ENGINEERING

Available Online at www.ajcse.info
Asian Journal of Computer Science Engineering 2019;4(3):1-12

RESEARCH ARTICLE

Enterprise Data Lakes for Credit Risk Analytics: An Intelligent Framework for Financial Institutions

Pushpalika Chatterjee

Independent Researcher

Received on: 11-07-2019; Revised on: 19-08-2019; Accepted on: 12-09-2019

ABSTRACT

Financial institutions face unprecedented challenges in managing massive, heterogeneous datasets for credit risk analytics while ensuring regulatory compliance and real-time decision-making capabilities. This paper introduces an intelligent enterprise data lake framework (IEDLF) designed to address these challenges through a unified, scalable architecture that integrates data engineering, machine learning, and metadata-driven governance. By applying the schema-on-read principles, the framework integrates structured, semi-structured, and unstructured data from various sources, including credit bureaus, transactional systems, and alternative data streams. The IEDLF transforms conventional static reporting systems into dynamic intelligence centers by integrating AI-driven credit scoring models, real-time processing capabilities utilizing Apache Spark, and automated ingestion pipelines leveraging Apache Kafka and NiFi. The architecture encompasses multiple layers: source, ingestion, validation, storage, and consumer – each optimized for specific functions within the credit risk analytics workflow. Implementation strategies incorporate comprehensive data quality frameworks using Great Expectations and Deequ to ensure reliability and transparency. The framework demonstrates how financial institutions can achieve scalable, compliant, and insight-driven credit risk management while overcoming limitations of legacy systems and siloed infrastructures, ultimately enabling enhanced predictive modeling, portfolio stress testing, and automated decision-making aligned with Basel III and International Financial Reporting Standard 9 regulatory requirements credit risk management while overcoming limitations of legacy systems and siloed infrastructures.

Keywords: Credit risk analytics, Credit scoring, Enterprise data lake, Financial data management, Machine learning

INTRODUCTION

A significant digital revolution in the financial sector is being fuelled by data-centric innovation, evolving regulations, and the increasing complexity of risk environments.[1] Financial institutions today handle massive, heterogeneous datasets from multiple sources – spanning client demographics, transactional records, industry trends, and other data sources, such as social media or geographic data. In such a data-intensive ecosystem, efficient data management and analytics are crucial for maintaining a competitive advantage, ensuring compliance, and enabling predictive insights.[2] However, legacy systems and siloed infrastructures limit the capacity to instantly process, combine, and evaluate these enormous amounts of data, creating barriers to effective risk assessment and strategic decision-making.

Many organizations are adopting enterprise data lakes (EDLs) as foundational platforms for modern data-driven analytics. EDLs provide a unified architecture that enables the large-scale intake, archiving, and processing of unstructured, semi-structured, and structured data.[3] Unlike traditional data warehouses, EDLs employ schema-on-read principles, allowing for flexibility and adaptability in analytical workflows. This architecture facilitates the direct integration of distributed computing, machine learning models, and big data technologies into the data ecosystem. For financial institutions, this shift represents a paradigm change transforming static reporting

Address for correspondence:
Pushpalika Chatterjee
E-mail: pushpalika.chatterjee@gmail.com

systems into dynamic intelligence hubs capable of supporting real-time analytics, anomaly detection, and automated decision-making.

In our data-driven world, one of the most crucial applications of advanced analytics is credit risk management.[4] Accurate assessment of borrower creditworthiness, default prediction, and regulatory reporting under frameworks such as Basel III and International Financial Reporting Standard (IFRS) 9 demand comprehensive, high-quality data integration.[5,6] Traditional credit risk systems, often limited by fragmented data sources and rigid ETL processes, struggle to adapt to the scale and velocity of modern financial data.[7] By leveraging EDLs, institutions can unify diverse credit-related data, including credit bureau records, loan performance data, behavioral analytics, and macroeconomic indicators, into a centralized, analytics-ready environment.[8] This unified data foundation enhances predictive modeling, portfolio stress testing, and compliance reporting while enabling adaptive and explainable AI applications in credit risk analysis.

Building on these advancements, an intelligent EDL framework (IEDLF) has been designed specifically for credit risk analytics in financial institutions.[9] The proposed framework integrates data engineering, machine learning, and metadata-driven governance to deliver a scalable and intelligent analytics environment. It automates data ingestion pipelines, supports real-time processing, and incorporates AI models for credit scoring and risk classification.[10] Moreover, the IEDLF emphasizes data governance, lineage tracking, and security compliance to ensure reliability and transparency in financial analytics. By merging the scalability of data lakes with the intelligence of AI-driven automation, the framework aims to redefine how financial institutions manage, analyze, and act upon credit risk data, fostering a more resilient, compliant, and insight-driven financial ecosystem.

Structure of the Paper

The paper is structured into six sections. The Introduction explains the need for efficient data management in finance. The Proposed IEDLF outlines the Intelligent Enterprise Data Lake Framework and its objectives. The Architecture and Implementation section describes the layered design and financial applications like credit risk and fraud detection. The Technology Stack covers tools, cloud platforms, and ML integration. The Literature Review highlights previous studies and research gaps, while the Conclusion and Future Work summarize key findings and suggest further development of AI-driven EDL systems in finance.

PROPOSED IEDLF

An IEDLF is an architectural and analytical approach for integrating, storing, processing, and analyzing large-scale, heterogeneous data for enterprise decision-making. It extends the traditional data lake concept by incorporating intelligence at multiple levels, including data ingestion, real-time or near real-time decision assistance, analytics, and governance, to satisfy business demands. An enterprise's ability to acquire, refine, archive, and explore raw data is enhanced by "Data Lake," a process made possible by a large data warehouse built using low-cost technology. An organization's data lake is a collection of raw, unstructured, or semi-structured data that appears to have no immediate use.

Figure 1 illustrates the process of integrating corporate analytics ecosystems with data warehouses and data lakes. Using Extract and Load (EL) procedures, data are ingested into the Data Lake from both internal sources (such as transactional systems, business applications, and operational databases) and external sources (such as social media, Internet Application Programming Interfaces [APIs], and streaming data). The data lake encompasses semi-structured, unstructured, and raw data, providing flexibility for future processing.

Data Lake Architecture Classification

Data lake architecture does not include architectural technology; rather, it refers to its conceptual arrangement at the greatest level of abstraction. With significant advancements over time, data-lake architecture has changed. Data-lake designs have been presented in a variety of ways, each with certain advantages for data analysis, data storage, and consumers (end users).

Mono zone architecture

To enable organizations to leverage the fundamental concept of a data lake, which aims to create a single, integrated pool of data from multiple sources for later use cases.

The data lake's initial architecture, as described in this early document, consisted of a single zone with a simple, flat shape. Figure 2 illustrates this mono-zone design, which stores all raw data in its original format. This configuration is often associated with the Hadoop setting, which enables the inexpensive loading of large-scale datasets.

Lambda architecture

To handle both batch and real-time data processing simultaneously, the Lambda concept was initially devised. Therefore, data processing and consumption are given precedence over data storage in its architecture. The speed layer no longer has access to the data after it has been put in the permanent memory. These two levels work together to deliver the data to end users through the serving layer's views.

Figure 3 illustrates the three layers that comprise the Lambda architecture, each with its own distinct levels: batch, speed, and serving. The data in permanent memory can be accessed and used in the batch layer, which provides a historical summary of the data. Only incremental data that has not yet been saved in persistent memory is processed by the speed layer, as opposed to the batch layer.

Kappa architecture

Figure 4 depicts Kappa architecture, a simplified variant of Lambda architecture that retains only the speed layer for simplicity, eliminating the batch layer.

The main objective is to complete nearly all of these processes through the speed layer or in real-time, thereby eliminating the need to constantly

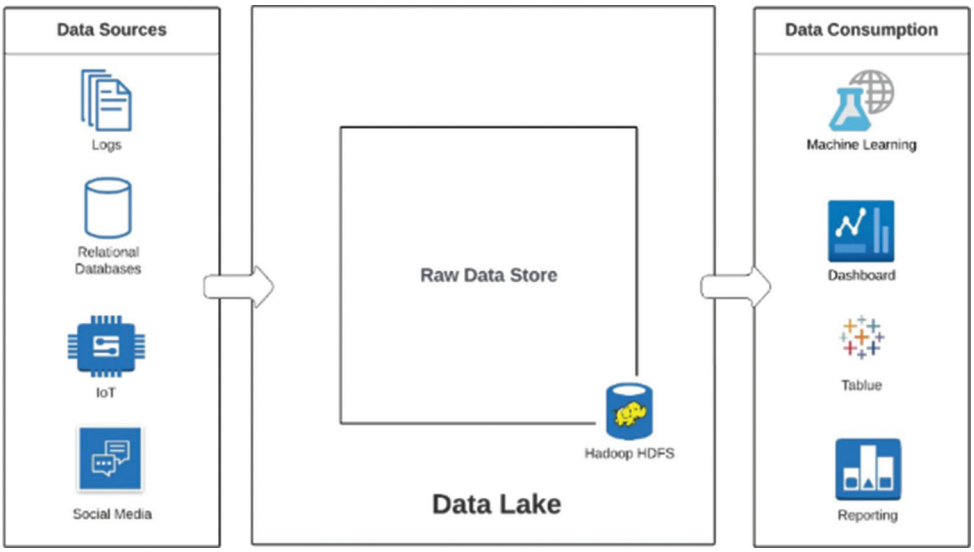

Figure 1: Mono zone architecture

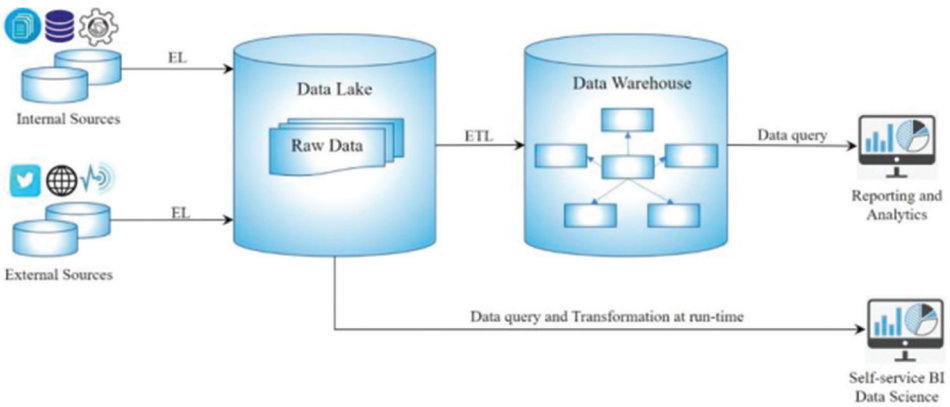

Figure 2: Architecture of data lake and data warehouse integration for enterprise analytics

recalibrate a batch layer from scratch. The Lambda architecture's drawback of needing to write and run the same logic twice has been circumvented by the Kappa design.

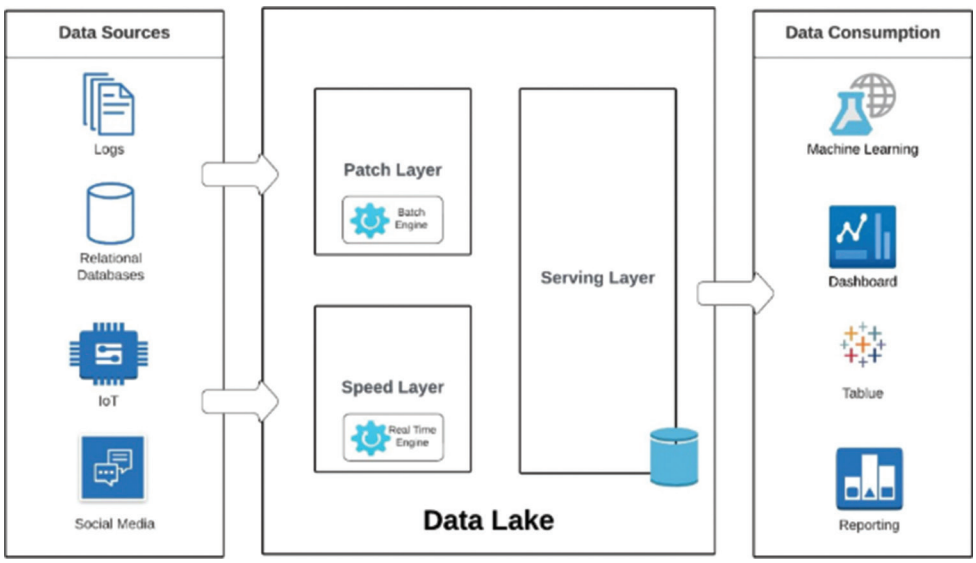

Figure 3: Lambda architecture

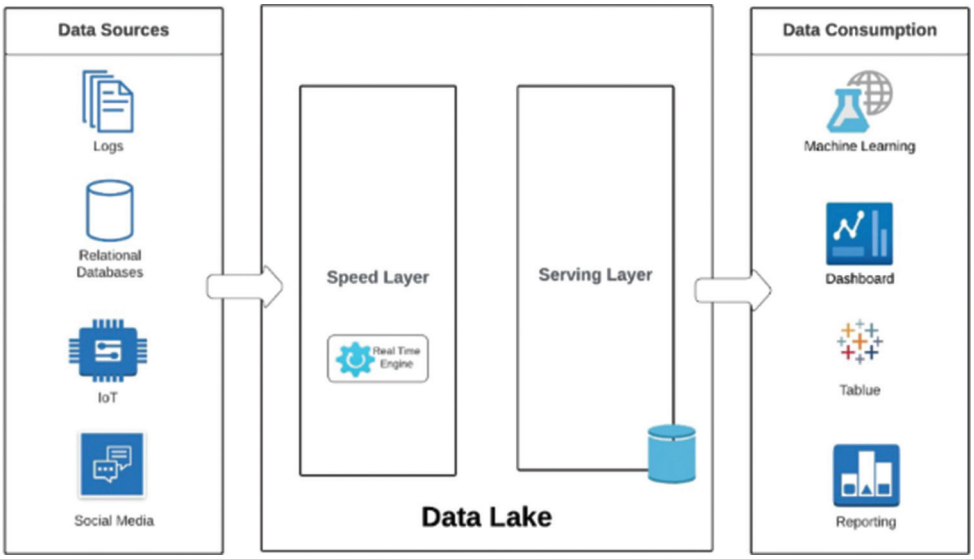

Figure 4: Kappa architecture

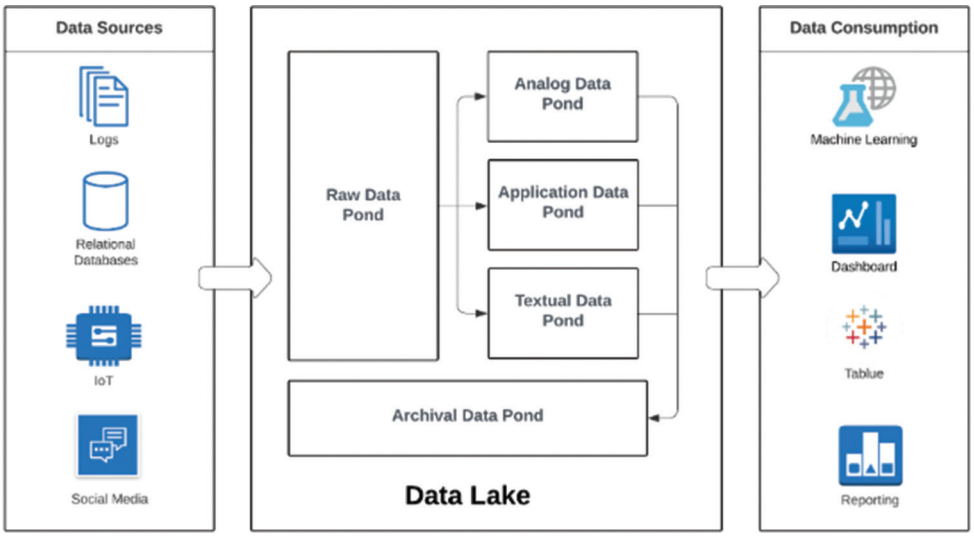

Figure 5: Data pond architecture

Data pond architecture

A data-lake design paradigm proposed by Bill Inmon is the data pond. The data-pond design, as shown in Figure 5, consists of five conceptually divided ponds, each with a specific function. The raw data absorbed from sources is stored in the first pond, which also serves as a staging area for the succeeding ponds and as a backup for the current applications of other companies.

This pond stores raw data until it is transferred to another pond, at which point it is purged and rendered unusable for further processing. IoT devices and APIs are among the high-velocity, semi-structured data sources stored in the analogy-data pond. Extract-transform-load (ETL) procedures fill the application data pond and operate similarly to a data warehouse.

Zone-based architecture

Their focus (processing versus governance) reveals significant differences in the quantity of zones they support, the user groups they cater to (business users and data scientists, or just data scientists), and the zones they include. The basic concept is still the same, though. For instance, well-liked data lake models during the past few years.

The concept in Figure 6 comprises four main zones and a sandbox zone, each with its own data structures and applications. The data first lands in the transitory landing zone, where it is momentarily held in its unprocessed state. Indexing and adding relevant metadata to augment records. Finally, data exploration and ad hoc analysis may be tested in the sandbox.

The Capabilities of A Data Lake

In the era of big data, businesses must constantly gather and evaluate new kinds of data.[11] Following the development of the first data lake example for online data management at an internet firm, several additional types of data suites were discovered. Data lakes gained popularity in the business data management environment for this reason.

The data lake supports the following features:

- Affordably collect and store massive amounts of raw data: Data lakes enable the scalable and efficient storage of enormous amounts of raw data without requiring a lot of pre-processing, which has raised the cost of data storage due to the fast expansion in data volume.
- Store many types of data in the same repository: Multi-structured data, which has several unknown characteristics, structured data, and multimedia data, including text, graphs, and videos, are all provided by traditional DBMSs. To handle these diverse types of data, different techniques must be applied.
- Perform transformations on the data: Data lakes and pre-processing are primarily utilized for ETL data processing to conduct additional system analysis.
- The "schema on read" describes the data's structure right before its used: Complex, costly data modeling and integration are no longer necessary thanks to the data lake.
- Analyze a particular issue using highly narrow use cases: To find out what the data lake's usable data is, individuals must create specialized analytics because the values of the data are unclear.

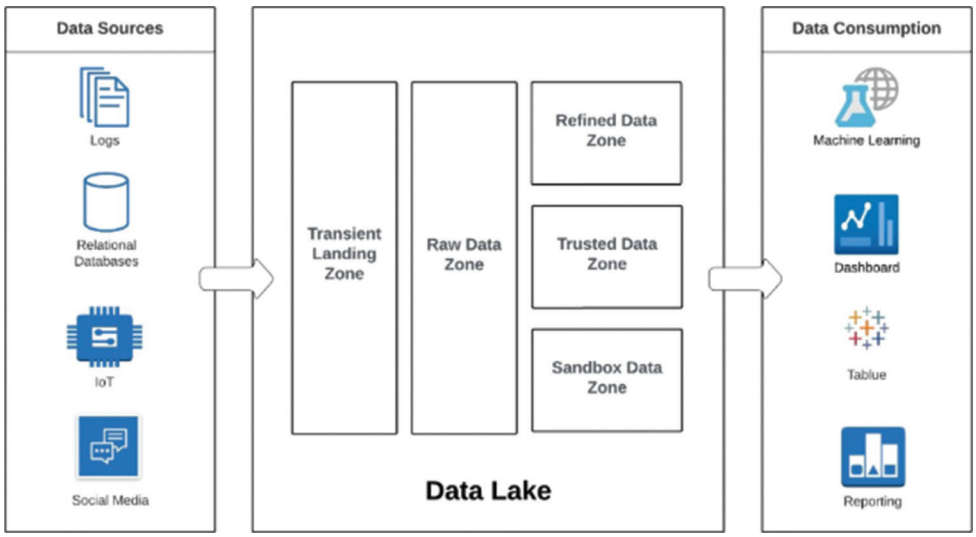

Figure 6: Zone-based architecture

The Data Lake and Data Warehouse

To manage the volume of web data and perform novel transformations to support critical business applications, such as web indexing and page searching, internet corporations have developed a data lake. However, when the big data tsunami approaches, businesses that have invested heavily in building corporate data warehouses begin to build data lakes. To provide a single, replicable version of the truth, a corporate data warehouse was developed. An extremely well-designed system is the data warehouse. Complex data models are often created before the data are loaded into the data warehouse.[12] In addition, data warehouses are built to handle tasks that are executed in batches and Support Hundreds to thousands of people working on analytics or reporting jobs simultaneously.

The key differences between an enterprise data warehouse (EDW) and a data lake are listed in Table 1. It clearly compares the two strategies by highlighting factors like workload handling, schema design, size, benefits, query techniques, data type, cost, and complexity.

Table 1: Comparison of EDW and data lake

Dimension	EDW	Data Lake
Workload	Hundreds to thousands of users at once Executing interactive analytics Enhanced capacity for task management Batch processing	Large-scale batch data processing Its powers are still being improved Encourage more engaged users
Schema	A schema is usually established before data storage Offers performance, security, and integration but necessitates initial effort	A schema is usually defined following data storage. Provides exceptional agility and makes data collection simple
Scale	Large data volumes at moderate cost	Large data volumes at affordable prices
Benefit	Fast response Consistent performance Easy to use Data integration Cross-functional analysis	Superb scalability Programming support Radically change
UERY	SQL	Programming
Data	Cleansed	Raw
Cost	Efficient use of CPU/IO	Low cost of storage and processing
Complexity	Complex joins	Complex processing

EDW: Enterprise data warehouse

ARCHITECTURE AND IMPLEMENTATION OF EDL IN FINANCIAL

The combined design of the ecosystem of a financial institution's data warehouse and data lake. EL pipelines are used to add data to sources for the Data Lake, both within and outside the company. The Data Lake retains raw financial and transactional data, which is later processed and transferred to the Data Warehouse through ETL operations for structured analytics and reporting, as shown in Figure 7. Some of the layers in IEDLF are discussed below:

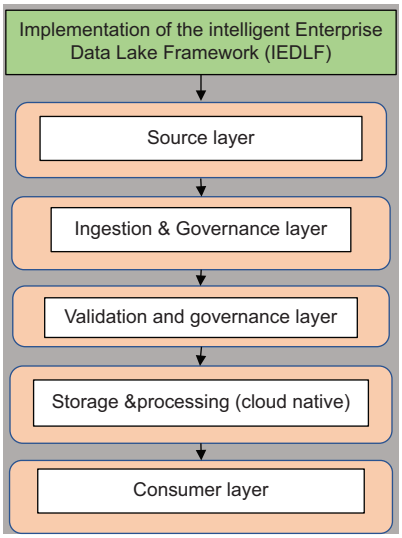

Figure 7: Implementation of the intelligent enterprise data lake framework

Source Layer

Capture diverse input streams from multiple credit bureau exchanges, consumer service systems, financial APIs, and unstructured sources. There are several examples of source layers.

- Legacy Exchanges: Flat file through SFTP.
- Consumer Service Systems: REST APIs, relational databases (Oracle, SQL Server).
- Banking/FinTech APIs: ISO 20022 transaction feeds, JSON over HTTPS.
- Unstructured Sources: PDFs, CSVs, XML documents.

Tech stack

- API Gateways (Apigee/AWS API Gateway)
- SFTP servers + NiFi processors
- Cloud Storage connectors

Ingestion and Migration layer

Consolidate all incoming data into the platform using streaming and batch ingestion, a component of the layer.

Multi-exchange migration engine

- Apache Kafka/AWS Kinesis for event-driven ingestion.
- Apache NiFi pipelines for batch file transfers.
- Schema normalization scripts.

Consumer data aggregator

- SQL-based transformations to harmonize consumer service data
- Merge/join logic across 4+ sources

Schema normalizer

- The canonical model applied to all credit bureau entities

Sample (Kafka producer for Bureau Feed)

```
import json

producer = KafkaProducer(
    bootstrap_servers='broker:9092',
    value_serializer=lambda v: json.dumps(v).
    encode('utf-8')
)
exchange_data = {
    "exchange_id": "EX123",
    "customer_id": "CUST001",
    "account_status": "active",
    "balance": 4500
}
producer.send("bureau-exchange-topic",
value=exchange_data)
producer.flush()
```

Validation and Governance Layer

Ensure data quality, compliance, and modernization of legacy debt-monitoring workflows. Component

- Automated Validation Rules

Great Expectations/Deequ for checks

- Data Quality Index (DQI)

Composite scoring across timeliness, completeness, validity, uniqueness

- Legacy Modernization Service

Re-engineered Perl UDM → Spring Batch + Spark

- Metadata Lineage & Compliance

Apache Atlas/AWS Glue Catalog

Sample (Great Expectations Rule for DQI)

```
Import great_expectations as ge

df=ge.read_csv("consumer_service_data.csv")
df.expect_column_values_to_not_be_
null("customer_id")
df.expect_column_values_to_be_
between("balance", min_value=0)

# Generate validation report
results = df.validate()
print(results).
```

A. Storage & processing(Cloud-Native)

Store, organize, and process data at scale with clear lifecycle zones.

Zones

- Raw Zone: Immutable landing in Parquet/ ORC.
- Curated Zone: Normalized, cleaned data for analytics.
- Feature Zone: Business-ready datasets for downstream scoring and compliance.
- Data Fabric Layer: Orchestrated ingestion + Spark jobs + BigQuery tables.

Tech stack

- Storage: AWS S3/Azure Data Lake/GCP Cloud Storage
- Processing: Spark on Dataproc/EMR/ Databricks
- Orchestration: Apache Airflow

Sample (Spark Aggregation in Curated Zone)

```
from pyspark.sql import SparkSession

spark = (
   SparkSession
   .builder
   .appName("curated_aggregation")
   .getOrCreate()
)

df = spark.read.json("gs://finance-raw-zone/
bureau_data/")

agg_df = df.groupBy("customer_id").
sum("balance")
```

```
# or: agg_df = df.groupBy("customer_id").
agg({"balance": "sum"})

agg_df.write.mode("overwrite").parquet(
  "gs://finance-curated-zone/customer_balances/"
)
```

Airflow DAG Orchestration (Data Fabric Layer)

```
from datetime import datetime, timedelta

from airflow import DAG
from airflow.contrib.operators.dataproc_operator
import DataProcPySparkOperator

default_args = {
   'owner': 'data-eng',
   'depends_on_past': False,
   'start_date': datetime(2019, 2, 2),
   'email_on_failure': False,
   'email_on_retry': False,
   'retries': 1,
   'retry_delay': timedelta(minutes=5),
}

with DAG(
   dag_id='finance_data_fabric',
   default_args=default_args,
   schedule_interval='@daily',
   catchup=False,   # often added later, but safe to
keep
) as dag:

                        run_spark_curated_job =
DataProcPySparkOperator(
      task_id='run_spark_curated_job',
      main='gs://scripts/curated_job.py',
      cluster_name='finance-dataproc',
      region='us-central1',
      project_id='finance-gcp',
      arguments=[],
    job_name='finance-curated-job-{{ ds_nodash
}}',
  )
```

Consumers Layer

Deliver curated insights and compliance-ready reports to stakeholders.

Consumer

- Risk Officers & Underwriters: Dashboards (Power BI, Tableau).
- Regulators & Compliance Teams: Lineage reports, DQI dashboards, automated compliance APIs.
- Business Teams: Consumer service analytics for decision-making.

TECHNOLOGY STACK AND IMPLEMENTATION STRATEGIES FOR EDL

The successful deployment of an EDL for credit risk analytics requires a robust technology stack and carefully designed implementation strategies. The technology landscape for big data was dominated by open-source frameworks and distributed computing systems that could efficiently handle large-scale, heterogeneous datasets.[13] This section outlines the core technological components and implementation approaches relevant to financial institutions

Big Data Ecosystem Components

- Hadoop Distributed File System (HDFS): Supports the data lake as its fundamental storage layer, enabling distributed storage of large-scale datasets across commodity hardware. High-throughput access to data, scalability, and HDFS are ideal for storing financial data, both structured and unstructured, due to its tolerance for failure.
- Apache Spark: A distributed computing platform that runs in memory and provides fast data processing for batch and real-time analytics.[14] Spark is ideal for credit risk analytics because it offers MLlib, graph processing (GraphX), and structured data processing (Spark SQL).
- Apache Hive: Enables SQL-like querying of big datasets kept in the data lake by providing a data warehousing layer on top of Hadoop. Hive facilitates data exploration and ad-hoc analytics for credit risk assessment models.

Data Integration Tools

- ETL/ELT Framework: Incorporating diverse credit information from transactional systems, credit bureaus, APIs, and other sources requires the use of ETL or ELT procedures. Tools like Informatica, Talend, and Apache NiFi.

- Data Ingestion Frameworks: Apache Flume and Kafka are commonly employed for ingesting streaming data in real time. Kafka, with its publish–subscribe model, allows financial institutions to capture and process live transactional data for timely credit risk assessment.

Machine Learning and Analytics Platforms

The integration of ML and analytics platforms with EDLs was a critical enabler for advanced credit risk analytics. These platforms provided the computational power and algorithmic flexibility required to process large-scale, heterogeneous financial datasets and produce actionable insights for risk management.

Machine learning libraries

Machine learning (ML) tools, such as Scikit-learn, Apache Spark MLlib, and TensorFlow, have been widely adopted for developing predictive models in the credit risk domain.

- Spark MLlib offered distributed ML algorithms for classification, regression, clustering, and collaborative filtering, enabling scalable model training directly on data stored in Hadoop clusters
- Scikit-learn provided a rich set of supervised and unsupervised learning algorithms suitable for smaller datasets and rapid prototyping of credit scoring models
- TensorFlow, though primarily a deep learning platform, supported custom neural network architectures for advanced risk prediction and anomaly detection in credit transaction data.

Predictive modeling for credit risk

Machine learning platforms enabled institutions to build and deploy models for:

- Credit Scoring: Predicting borrower creditworthiness using historical financial behavior
- Default Prediction: Anticipating potential loan defaults through pattern recognition
- Fraud Detection: Identifying anomalous transactions using classification and clustering techniques.

Analytical tools

Business Intelligence (BI) and data exploration tools complement ML platforms by allowing analysts to interact with model outputs and interpret results effectively. Commonly used analytical tools included:

- Tableau and QlikView for interactive dashboards and visual analytics.
- Apache Zeppelin and Jupyter Notebooks for exploratory analysis, model development, and sharing of reproducible research.

Benefits and limitations of EDL

EDLs provide a unified platform for storing and analyzing vast amounts of heterogeneous data, revolutionizing data management in financial organizations. Some of EDL's main advantages and drawbacks are covered here:

Scalability

EDLs typically expand horizontally to handle and store petabytes of financial data, which can be in any format: semi-structured, unstructured, or structured. This makes them ideal for credit risk analytics where data volume grows continuously due to transactional systems, IoT devices, and external credit bureaus.

Cost efficiency

By leveraging commodity hardware and open-source frameworks such as HDFS and Apache Spark, data lakes significantly reduce infrastructure and storage costs compared to traditional EDWs. This enables financial institutions to manage large-scale credit datasets economically.

Flexibility in data storage

Data lakes natively support various types of data, including semi-structured (JSON, XML), unstructured (log files, PDFs), and structured (relational databases). This flexibility enables financial institutions to incorporate diverse sources, such as bank transactions, social media feeds, and third-party credit information, for more comprehensive credit risk analytics.

Real-time and historical analysis

Data lakes facilitate both batch and stream processing, enabling simultaneous access to

Table 2: Comparative analysis of EDLs and credit risk analytics in the financial environment

Author (year)	Focus area	Key findings	Limitations	Future work/research direction
Llave (2018)	Business Intelligence and EDLs	Data lake implementation in enterprises supports three major purposes as staging areas for data warehouses, platforms for data scientist experimentation and direct resources for self-service business intelligence.	Lack of large-scale empirical evidence; limited insights into governance and metadata management challenges.	Conduct comprehensive empirical studies to analyze performance, scalability, and data governance of EDLs.
Addo, Guegan, and Hassani (2018)	Credit Risk Analytics using Machine Learning	Developed binary classifiers (tree-based and deep learning) for loan default prediction; tree-based models show better stability and interpretability than deep learning models.	Dataset limited to specific financial context; generalization across geographies not evaluated.	Extend analysis to ensemble and hybrid models; include explainable AI (XAI) approaches for credit risk interpretability.
Ravi and Kamaruddin (2017)	Digital Transformation in BFSI (Banking, Financial Services, and Insurance)	Described transition from traditional banking to data-driven digital banking with IoT, Blockchain, Chatbots, and Robotics integration.	Lacked quantitative validation and performance comparison of emerging technologies.	Develop analytical models to evaluate the impact of new-age technologies on customer experience and operational efficiency.
Ferreira, Almeida, and Monteiro (2017)	Data Warehouse and BI for Financial Sector	Microsoft SQL Server Integration and Analysis Services are used in the suggested BI solution; OLAP cubes improve data aggregation, automation, and decision-making.	Focused on a single organizational context; scalability for larger data ecosystems was not analyzed.	Extend architecture for real-time analytics and integration with unstructured financial data sources.
Guo *et al.* (2016)	Evaluation of Credit Risk in P2P Lending	Proposed data-driven investment framework using instance-based credit risk modeling; optimized loan portfolio allocation using boundary-constrained portfolio optimization.	Limited to specific P2P lending datasets; external market dynamics not incorporated.	Enhance model with macroeconomic indicators; apply deep learning to improve cross-market generalization.

EDL: Enterprise data lake

historical data and real-time transaction streams. This capability enables dynamic credit risk assessment and facilitates timely decision-making, thereby enhancing the responsiveness of risk management processes.

Data quality management

Without strong metadata management and schema governance, EDLs can suffer from inconsistent or low-quality data. This may lead to unreliable analytical outcomes unless proper data profiling and cleaning processes are implemented.

Governance complexity

Large-scale data lakes require robust governance structures to ensure adherence to internal and industry standards. Managing data lineage, ownership, and usage rights across diverse datasets is a complex challenge for financial institutions.

LITERATURE REVIEW

This section presents earlier studies on EDL architectures and their applications in financial analytics and credit risk management. Table 2 presents a structured comparison of prior research, highlighting the limitations of big data–driven research and discussing future work.

Llave (2018) suggests that businesses can significantly enhance their business intelligence by leveraging data technology. However, there has been little actual study on this subject regarding how businesses utilize the data lake concept. The findings of a preliminary investigation aimed at enhancing comprehension of the data lake methodology's implementation in companies adopted this strategy across several businesses. They determined three key goals for implementing data lakes: as data warehouse sources or staging areas, for data scientists' and analysts' experiments, and as a direct source of self-service business knowledge.[15]

Addo, Guegan, and Hassani (2018) emphasized that credit risk projections, monitoring, reliable models, and efficient loan processing are essential for ensuring transparency and effective decision-making. Their study utilized real-world data to train machine learning and deep learning models capable of predicting the probability of loan default through binary classification. By incorporating selected model features into customized classifiers, they assessed the robustness of various binary models and evaluated their performance

across multiple datasets. The findings revealed that multilayer artificial neural network models were less stable compared to tree-based models.[16] Ravi and Kamaruddin (2017) note that the financial services sector is rapidly eschewing the conventional paradigm in favor of advanced digital client and transaction methods. The paradigm of journal and ledger entry has given way to one that is fueled by data and analytics in the digital banking sector. Big data analytics has significantly impacted consumer behavior in the financial services, insurance, and banking (FIs) industries. This is one of the most significant advantages of new technologies, such as blockchain, chatbots, robots, and the Internet of Things (IoT).[17]

Ferreira, Almeida, and Monteiro (2017) conducted a multifaceted study of corporate data and a data warehouse for a financial holding firm. The goal is to create a business intelligence system that enables easy, quick, and flexible access to current, aggregated, actual, and/or forecasted information about bank accounts, utilizing the Integration Services and Analysis Services features in Microsoft SQL Server. Data from operational databases that support cash management data are also extracted and processed by it. Examining current and preliminary aggregated financial data using online analytical processing cubes enhances performance and provides a more automated and reliable approach.[18]

Guo *et al.* (2016) propose an alternative funding source that is independent of conventional financial institutions. Since P2P lending markets lack an explicit asset allocation mechanism, individual investors must precisely estimate the credit risk of each loan to allocate their money among multiple loans. A data-driven approach for making investment decisions in this emerging field is a methodology for assessing credit risk based on specific instances, it can calculate the potential loss and gain from every given loan. Furthermore, in-depth tests on actual datasets from two prominent peer-to-peer lending marketplaces demonstrate that P2P loan investment decisions are an optimization problem for a portfolio with boundaries.[19]

To guarantee the scalable, dependable, and safe deployment of AI/ML models for crucial applications such as fraud detection, risk assessment, and real-time consumer analytics, the financial sector is progressively using cloud-native technology. Agility and scalability are often limited by conventional on-premise or monolithic deployment techniques, particularly in environments that require high-frequency data processing and adherence to regulations. A thorough framework for deploying cloud-native models designed especially for financial applications that includes microservices architecture, orchestration, CI/CD pipelines, and containerization.

CONCLUSION AND FUTURE WORK

EDL represents a transformative evolution in credit risk analytics, merging the scalability of data lakes with AI-driven automation and robust governance mechanisms. It effectively addresses critical challenges in modern financial data management, including the integration of heterogeneous data sources, real-time analytical processing, and adherence to regulatory mandates such as Basel III and IFRS 9. Through its layered architecture encompassing data ingestion, validation, cloud-native storage, and intelligent consumption layers, the framework establishes a unified, analytics-ready ecosystem that transcends fragmented legacy infrastructures. This integration enables more accurate predictive modeling, portfolio stress testing, and automated credit decision-making within financial institutions. Future research should emphasize enhancing model interpretability through explainable AI to strengthen regulatory transparency, applying federated learning techniques to enable secure collaboration across institutions while maintaining data privacy, and developing real-time anomaly detection mechanisms using streaming analytics for proactive fraud prevention. Moreover, incorporating environmental, social, and governance (ESG) indicators into credit risk models is essential to align with sustainable finance objectives. Ultimately, comprehensive empirical evaluations across diverse financial contexts are necessary to validate scalability, assess performance improvements, and establish standardized governance practices for implementing EDLs in highly regulated environments.

REFERENCES

1. Chen N, Ribeiro B, Chen A. Financial credit risk assessment: A recent review. Artif Intell Rev 2016;45: 1-23.

2. Singh HP, Kumar S. Working capital requirements of manufacturing SMEs: Evidence from emerging economy. Rev Int Bus Strateg 2017;27:369-85.
3. Pomp A, Paulus A, Kirmse A, Kraus V, Meisen T. Applying semantics to reduce the time to analytics within complex heterogeneous infrastructures. Technologies 2018;6:86.
4. Ji C, Shao Q, Sun J, Liu S, Pan L, Wu L, *et al*. Device data ingestion for industrial big data platforms with a case study. Sensors (Basel) 2016;16:279.
5. Pathak P, Shrivastava A, Gupta S. A survey on various security issues in delay tolerant networks. J Adv Shell Program 2015;2:12-8.
6. Kushwaha A, Pathak P, Gupta S. Review of optimize load balancing algorithms in cloud. Int J Distrib Cloud Comput 2016;4:1-9.
7. Nerella VM. MIGRATE: A rollback-enabled framework for automated oracle XTTS-based cross-platform database migrations. J Electr Syst 2018;14:85-95.
8. Crouhy M, Galai D, Mark R. A comparative analysis of current credit risk models. J Bank Financ 2000;24: 59-117.
9. Lai KK, Yu L, Wang S, Zhou L. Credit risk analysis using a reliability-based neural network ensemble model. In: Artificial Neural Networks ICANN International Conference, Athens, Greece 2006; 2006. p. 682-90.
10. Hassan MK, Brodmann J, Rayfield B, Huda M. Modeling credit risk in credit unions using survival analysis. Int J Bank Mark 2018;36:482-95.
11. Fang H. Managing data lakes in big data era: What's a data lake and why has it become popular in data management ecosystem. In: 2015 International Conference on CYBER Technology in Automation, Control, and Intelligent Systems. IEEE-CYBER; 2015. p. 820-4.
12. Gröger C, Schwarz H, Mitschang B. The deep data warehouse: Link-based integration and enrichment of warehouse data and unstructured content. In: 2014 IEEE 18th International Enterprise Distributed Object Computing Conference; 2014. p. 210-7.
13. Vinod Kumar L, Natarajan S, Keerthana S, Chinmayi KM, Lakshmi N. Credit Risk Analysis in Peer-to-Peer Lending System. In: 2016 IEEE International Conference on Knowledge Engineering and Applications (ICKEA); 2016. p. 193-6.
14. Nerella VM. Automated cross-platform database migration and high availability implementation. Turkish J Comput Math Educ 2018;9:823-35.
15. Llave MR. Data lakes in business intelligence: Reporting from the trenches. Proc Comput Sci 2018;138:516-24.
16. Addo PM, Guegan D, Hassani B. Credit risk analysis using machine and deep learning models. Risks 2018;6:38.
17. Ravi V, Kamaruddin S. Big Data Analytics Enabled Smart Financial Services: Opportunities and Challenges. In: Big Data Analytics. Conference: International Conference on Big Data Analytics; 2017. p. 15-39.
18. Ferreira J, Almeida F, Monteiro J. Building an effective data warehouse for the financial sector. Autom Control Inf Sci 2017;3:16-25.
19. Guo Y, Zhou W, Luo C, Liu C, Xiong X. Instance-based credit risk assessment for investment decisions in P2P lending. Eur J Oper Res 2016;249:417-26.