Deep Science Publishing
https://doi.org/10.70593/978-93-49910-38-6_3

DeepScience
Open Access Books

Chapter 3: Modern data engineering architectures for financial decision systems

3.1. Introduction to Data Engineering in Finance

Financial decision systems have gained great interest over the last two decades, as computerized trading, credit risk management, risk hedging in asset derivatives, and credit card fraud detection became widespread. Among the numerous automated systems being provided, the highly volatile financial markets and the requirements for technology upgradability forced most firms to develop their own financial decision systems. Overall, the financial domain has a significant market demand for automatic decision systems. As such systems can be very complex, there is a need for systematic approaches to their design and programming. This paper presents the new emerging area of data engineering for financial decision systems, discussing its market demand and introducing its new analysis, design, and implementation principles based on model-driven architecture and composite applications. It presents results from an extensive empirical study of a large European investment bank, examining its data engineering processes. This research was inspired by the business requirements of data engineering in this specific financial institution, providing a comprehensive view of the field. It represents novel academic scholarship based on rigorous detailed empirical research.

This paper focuses on data engineering for financial decision systems, an area not yet investigated in the literature, indicated as a new important emerging area of research. Data engineering has gained importance with the emergence of data warehousing and data mining technologies. Over the last two decades, there has been great interest in decision systems in the financial domain, propelling the commercial development and academic study of financial decision systems. The explosive growth of these two new emerging application areas has sparked the need for a systematic approach to the engineering of complex financial decision systems, thereby paving the way for the

emergence of data engineering for financial decision systems, a new area of research and practice.

Thus, the specter of outperformance by years by the shunning of data still begs to be conjured. For financial decision systems requiring procedural analyses of real-time or sequential data, this paper proposes a new systems-level architectural framework together with semi-aggregative methods to reveal characteristics of the data to aid financial decisions. With this framework and these applicable methods, new class-level decision systems can be made manifest. The proposed framework and methods define a synergistic yet modularized way to help build a better understanding of the inferential interaction between covariates (features) and responses (targets) within financial data.

Fig 3.1: Modern Data Engineering

3.1.1. Background and Significance

Recent decades have witnessed a rapid evolution of data associated with financial systems. Overnight, some companies have grown to be worth trillions of dollars based on mass amounts of financial data gathering. This massive growth in data comes with a massive growth in opportunities to extract value, which represents a change in how the financial world operates, making it more important than competitive advantages and strong capitalization. A collection of recent methods that use data effectively to extract value from observational data in economic systems is sometimes called “data

engineering". Although the term is sometimes associated with other techniques, there is no ambiguity about what is meant regarding the need for systems to marry this massive new amount of data with the desired value.

The financial community has long been struggling with how to better incorporate data into their decision-making systems. Well-recognized risks in this arena include misinterpretation of data-driven results, over-reliance on trend-predicting methods, naïve normalization and bins, pouring data over over-fit models, and playing black-box systems without sufficient understanding. Too often, sophisticated work is hampered and sometimes destroyed by failing to account for missing knowledge of representation, meaning, or complex inferential relationships. It is called Fisher's ire where data is viewed as "independent and identically distributed".

3.2. Key Concepts in Data Engineering

Technologies – or tools – are defined as accepted practices or systems to deliver a structured configuration. "Data engineering architecture" refers to integrated software frameworks to provide computing modules. A data engineering architecture consists of a computing architecture to model the delivery of algorithms preparing data (extraction, transformation, data enrichment), a storage architecture to store simulated data and control its delivery, a committee architecture to distribute backtest results, the currencies used, and additional meta-information surrounding events of interest. All these architectures consist of clouds hosting micro-economies. An information architecture defines the putative structure and interfaces of a system. The design of micro-economies as well as of funds to host events of interest and upper-layer applications take the form of ontologies. An integration architecture consists of interface components connected to other systems that interface with third-party applications. These components also include a cloud-based weather service interfaced at an exchange level.

Data is said to be processed if it is delivered ─ individuals, companies, or instruments whose metadata is expressed in just a few bytes, used by one or multiple data abstractions. To filter price time series yields an array of products and instruments whose recorded price ticks are updated; it uses various services to filter price data. Products whose quotes vary less than a fixed percentage are screened out. Ordinarily, any managed data is accumulated and delivered to be archived, transformed, enriched by calculated observations, predicted using machine learning procedures, or even used in a simulated market. Various tasks throw, store and deliver attempts at a lower application layer. Access and service interfaces are put in application processes, which preclude the top level to access lower layers.

3.2.1. Data Pipelines

The growth in the diversification and complexity of data sources combined with the rapid growth of data volumes and types has made the construction of a Data Pipeline an important step for many companies in a wide variety of industries (Konsisme, 2023; KPMG, 2024; Aladebumoye, 2025). The Data Pipeline is utilized to improve the efficiency and consistency of the organization's daily data processing workflow and is of crucial importance to analysts and management for solving efficiency problems, urgent problems, complex problems, and design decisions.

A data pipeline can be seen as a system that is responsible for the automated processing and transmission of data. The data pipeline can extract the data from the source system, through cleaning, conversion, loading and a series of other steps, and finally transfer the data to the target system. In addition, the data pipeline can also generally monitor the safety, reliability, and completeness of the data in the pipeline and record logs of the conduct and alert notifications when a fault occurs. The main function of the data pipeline is to improve the efficiency of data processing, ensure the accuracy of data, and ensure the security of data. Overall, the data pipeline plays an important role in the modern enterprise.

Building a data pipeline generally consists of five steps: 1. data source: determine which source(s) to extract the data from; 2. data extraction: construct the SQL statements or file-paths for the extraction of raw data; 3. data transmission: set the protocol between data source and target; 4. data processing: set the processing logic; 5. target: choose which target system to store the processed data on.

3.2.2. Data Warehousing

An increasing number of enterprises use Data Warehousing (DW) techniques to obtain better descriptions of their activity for decision making. DW is often seen as a 'frozen' set of data and not of dynamic characteristics able to adapt to the rapid changes of the Web. It is acknowledged that researchers should provide more work focused on managing dynamic environments, but only few research works are completely dedicated to this issue. In addition to the replenishment of the databases, this paper adds to actual data loads the continuous changes of the data sources that generate the online operational database. The problem of changing data sources is seen in many light from the availability of new sources in the DW, policy modification at data source, change in data source format, etc. These problems become more complex when applied to the Internet world. New sources become available, which could have been ignored due to the potential overload of information. A thorough analysis of changing sources for DWs and decision on updates to the source and not only to the data could provoke a great impact

on the overall system architecture. DWs become an essential ingredient for companies regardless of their market size, main business, or geographic location.

Decision Support (DS) systems are one of the core components of Information Technology (IT) systems used for providing additional information and knowledge that help to solve common operational or strategic activities. They use data stored in Data Warehouses (DW) previously transformed from Operational Information Systems (OIS). This paper addresses the main principles and technologies used for designing and evolving Data Warehousing architectures for Data Warehouses. It focuses on On-Line Analytical Processing (OLAP) engines and Data Source Conformity tools since they are often adopted decision systems in the Finance domain.

3.3. Architectural Patterns in Financial Systems

Architectural patterns facilitate architecture design by providing reusable solutions for recurrent problems (Paulson and Partners, 2025; Wolters Kluwer, 2025). Structural domain types and transactional types arising from the domain types confer four architectural patterns that differ in computational resource distribution and autonomy. Each architectural pattern has two alternative configurations that differ in relations among scattered components. Based on a client-server cooperation style, a pipe-filter architecture design pattern for financial decision systems using transactional types is described. Careful choice of a data transport format for different systems allows disequilibrium, subjective interim results, and client diversity. A rich set of data transformation types enables a wide range of functional enrichment, including locations and possible duration of trades, as well as generic classification into basic handling and alarm generation operations. In general use, spill flow of iron ore and indispensable conditions for trade execution are analyzed. For specific purposes, visual and written processing of a trader’s diary in view of the participant’s regulation violations is shown. For realistic tests of scalar hedge performance measurement systems, historical data that creates usage records of different types is modeled.

Some basic concepts are presented underlying financial decision systems that implement direct and indirect transactions in three domains: buying and selling raw materials, money currency exchange, including mis-selling, and stock restitution support. They have been encapsulated into domain types describing the fundamental entities interacting in these domains and into transactional types implementing operations on the domain types. Based on classification of the transactional types into control, access, updating, and report types, typical message flows and data transport formats, as well as data transport protocols and access models for processing files in drain piles are given. It is shown how greater autonomy for accessibility financial decision is achieved by keeping the subject of financial transactions apart from their processing. One possible structural

domain type of a medium of exchange, a foreign money currency, encapsulates an origin country, its three-letter code according to the international standard, and two terminologies. It is implemented as an abstraction identical in all three domains. Each domain uses standard data transport format for this currency type. A data structure of a transport file type containing two currency amounts with the paths to exchange rates and calculations, in a knowledge base, is given.As indicated by the rapid evolution of social media networks, locations of epidemic diseases, and pathological states of machines, the rapid growth in volume, velocity, and variety of data, in turn, presents huge challenges for internet-based giants and ICT innovators. Nevertheless, for those massive demands of data-driven applications, the current technologies are not sufficient to cope with their processing demand. Batch data processing frameworks are able to provide substantial processing ability for massive static input data. But they are confronted with long processing latency, and as a result, increasingly falling behind the pace of data generation rate and failing real-time view of information. At present, fast real-time insights generally rely on in-memory or on-disk short-term static windowed batch processing technology.

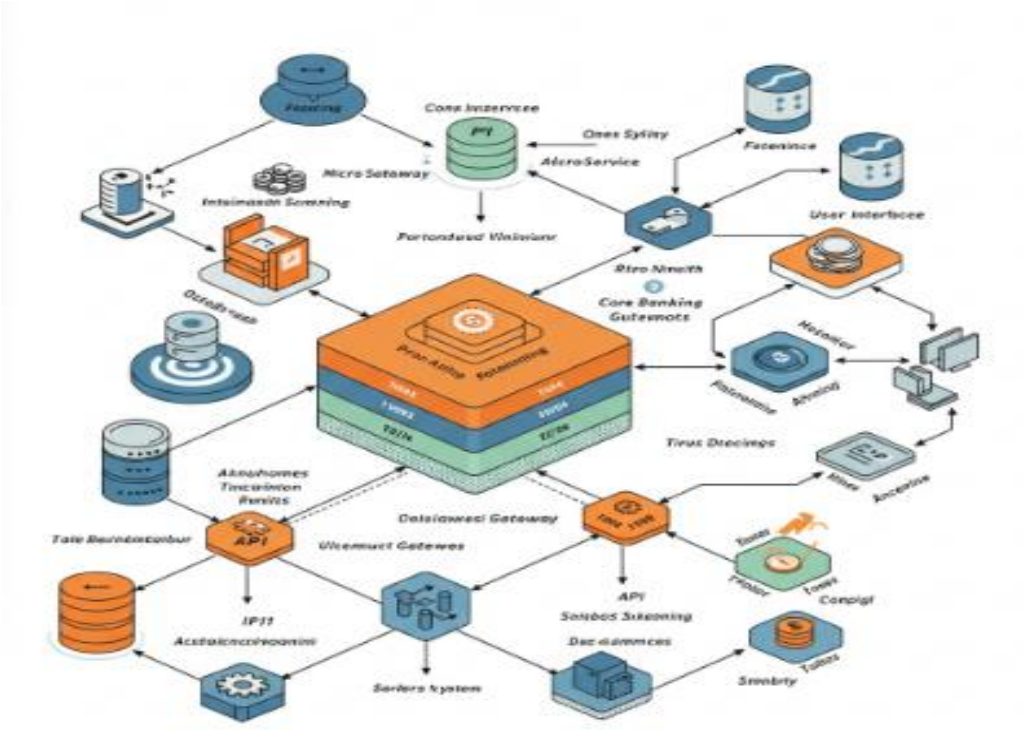

Fig 3.2: Architectural Patterns in Financial Systems

3.3.1. Batch Processing Architectures

For financial decisions processing, numerous big data and machine learning techniques have been widely and deeply investigated. Electronic trading has attracted more and more attention along with the boom of global financial markets and the fast-growing data volume. A major trading system is composed of a number of decision subsystems which usually run on cluster and for the sake of robustness and speed some of the systems are even programmed to run on GPU. With the overwhelming increase of trading volume, pricing and execution decisions need to be made faster and more accurately, and also new alpha models need to be generated more efficiently.

For macro trading decisions, batch pricing and risk evaluation systems, on which most of the currently used approaches rely on, are not sufficient because of the fundamental physics underlying them as complex, highly non-linear, and time-sensitive. Instead, microstructure level and agent-based approaches allow more accurate nature behaviors, however, they also need longer time to simulate and calibrate which are not suitable for the increasing trading volume. Undoubtedly, enormous data are produced nonstop at HFTs, and if appropriate processing can be accomplished, an alpha model that can output abnormal occluded behavior will be generated, leading to attracting a great amount of orders and benefiting enormous profits. It is worth noting that for such a situation, batch treatment will lead to huge waste and may result in loss, and thus a new stream treatment framework is needed for processing and generating real time alpha strategies.

3.3.2. Real-Time Processing Architectures

The evolution of data processing frameworks as commercial real-time processing systems aims to meet increasing workload Big-Data levels. Hence, open ecosystem stream processing frameworks, Apache Kafka and Apache Pulsar, are implemented as a main broker for connector and messaging functions. While executing queries is done, e.g., by open related query languages SQL-like, as Apache Flink, PostgreSQL, and Apache Ignite Streaming addition to the to-be-tested system separately, the existing high-level stream systems, archiving requests and controller tasks, store no related matured technologies. There are design and optimization methods, but complexity in decision frames still remains open. The first analysis allows to formulate the main components of the framework, especially in the domains of multi-criteria quality aspects. By means of an evolutionary algorithm based on these components' formulae and a popularity metric, desired properties in the pathways of this population can be efficiently improved and controlled.

The applicability of relevant instantiations with Apache Kafka and PRESTO outlier candidates are presented, especially with the data quantity since the last midnight books

object [8]. Even, based on general quality aspects, the decision frames of questions and values with a ‘yes or no’ approach are drawn. Out of an on-asp-lapsing time off-day time zone group (e.g., 008-122 and 122-008), atypically few daily tweets were always queried in Test-Scope 3c. With respect to expected query loads, peak moments more stringent to the storage and processing duration should be anticipated. All over templates with a variety of broker architectures are included, as well as instances specific for the data volume of items and the scalar extreme value. Because of genre-dependent quantities of data sources and ingest locations, a perfect partition strategy and adapted augmenters could not be envisaged. At last, the introduction of a ranking variant into a pipeline would be small, on average only between 0.41 to 0.61. On an increasing number of queries, bottlenecks on some intermediate pipelines could surface heavily under real-timed conditions. So apart from query precondition selectors or prior unprepared data, the pipelining and re-running patterns’ problems should also be examined.

3.4. Data Governance and Compliance

To be compliant with the laws and regulations, financial companies’ risk assessment methodologies must comply with regulation. A documented governance process distinguishes those algorithms and methodologies that are treated as guidelines and do not require validation with specific documentation. The same governance framework must account for external stress-test scenarios, controls and scenarios, and the corresponding documentation requirements. The documentation procedure should enable both manual and automatic updates of regulatory documentation across financial companies. Proper continuity of care must be ensured for the delegation of methodologies to be acknowledged as documentation or models. New systems and new transactions can increase storage costs if not discarded properly, so an automatic storage strategy is crucial. Very old data may be recycled or discarded.

External Data Management holds external non-regulated data with external feeds from both banks and non-banks, likewise with coverage marking preferably used in a manner similar to Risk Data. The existing manual data governance processes need to be automated. A defined general-margin ERC to automate data governance processes identified needs to monitor both compliance and business risk. Governance artefacts covering business use, data lineage, and active monitoring initially are treated as proof of processes. Findings such as impacts in the model development phase are tracked by connecting the model blueprint to changes in monitored artefacts. Feedback messages reclassify findings or escalate tickets and annotate diagrams. Relevant proposals are extracted from production tickets to approximate asset-based RCA and feed into the Forum; occurrences are classified central, semantic, or structural. For full automation of

document updates, a framework is needed, as well as tools to specify, monitor, and visualize processes.

3.4.1. Regulatory Requirements

In terms of regulatory requirements, the architects were sensitive to general legislative expectations and specific regulatory recommendations. Jury-supported recommendations of regulatory authorities for diverse applications were taken into account. For the general category of information-based manipulation, the regulators recommended transparency of adaptive management. Regulatory authorities had emphasized specific processes such as outlier detection, investigation of transaction, and escalation to REMIT if necessary. RECACs had a legal hold, and they needed to preserve these records comprehensively with the ability to retrieve and prove background. For trade-based manipulation, authorities had recommended monitoring for finding algorithmic and marketable trading related suspicious activity. Such recommendations were angry for a pre-execution, intrusive approach that may raise competition concerns. This practical need of RECACs was juxtaposed with the academic expectation that governance by a set of robust models is effective with the caveat of its interpretability and adaptability. Model interpretability, as argued, was relevant but often with potential competition concerns that the RECACs had to deal with. The measurements complied with various regulators' recommendations including the European Securities and Markets Authority’s guidelines for internet news and social media analytics. Regulators often recommended detecting unwanted price movements. Such recommendations were in line with the general expectation of message reception and the extreme value theory proposed. To ensure that the models of the architects aligned with regulatory expectations, two prongs of requirement elicitation were adopted. General requirements, independent of application-specific considerations, were collected from the regulatory recommendations. The number of requirements was large due to heterogeneous regulatory expectations. A top-down prioritization was undertaken. The quantifiable properties, which were fundamental and objective alerts, were declared essential.

3.4.2. Data Quality Management

In today's world, organizations are using data as a commodity to improve decision-making processes and optimize the company’s operational processes. Data is usually generated from different organizations in different formats, speeds, schemes, and sources. Managing this data has become an incredibly challenging task. Organizations have to tackle these data from different sources, which cause challenges in understanding multiple formats and speeds of incoming data, and the ability to find, access and share

data between different sources. In addition, when managing huge, complex, fast, and varied data, there is a high effort needed from the business side of the system in order to provide requirements to process the data. The need for expertise, and knowledge of the data semantics in the business point of view should be present in the area of interest to ensure that important data domains are not missed during processing and monitoring. Understanding the overall landscape of the various architectures, technologies, formats, and norms of the used data sources is crucial to addressing the above-mentioned challenges. However, the architecture would be composed of lots of components, exchanging data among them in a different formats representation and speeds, which causes the capability of monitoring the quality of that exchanged data to be present. The amount of data being consumed and generated from operations is increasing dramatically, therefore detecting that the validity of the data is going to decrease.

To this framework MDE technique has been applied to design a generic UML-based Architecture Description Language (ADL). The generic MDE-based framework allows architecture modeling apart from the implementation code in the same way used in Enterprise Architecture. Thus, it could act as a blueprint for a wide spectrum of Data-Intensive applications. The whole entire architecture would be monitored on the data, databases, and the whole system with some predefined quality metrics. By moving the discussed architecture into the cloud, a more scalable architecture could be obtained. That, in turn, leads to the ability for increasing the monitoring coverage, becoming cloud agnostic and achieving high redundancy. Monitoring can be consumed in a chargeable way for these organizations that provide resources in the cloud environment.

3.5. Technologies and Tools

Modern data engineering architectures for financial decision systems should cover a two-fold perspective: the integral structures supporting the continual data capturing, modeling and analysis as well as the component parts and technologies making this possible. The paper will cover both aspects, focusing on the latter. The defined data engineering architectures and their technological implementations is a dynamic field. It is vital to keep updating academic research on the topic of industrial digital and data architectures as technologies and commercial needs change rapidly and new solutions are introduced.

A full stack implementation of the financial digital architecture has recently been constructed and went through industrial use case testing. Various components implementing the solution and part of the overall data architecture were extensively tested in simulation. These components are: Modern data lake architecture supporting the continual data capturing, storage and direct convenient querying of large defined dataset implementations. Group of static analysis solutions on large datasets and multiple

ML and AI implementations for dynamic and evolving datasets and streaming or extremely high volume data observations. An interactive notebook interface among which specialized components for the analysis of streaming data observations and deep risk analysis was tested in industrial operation.

The necessary tools and technologies apply not only to covering finance, but also to wider domains dealing with complex decision making and large scale data observations. The software tools and components include a Data lake architecture covering the storage of different types of observations with streaming and wider data storing tools along with automatic data capturing technologies from sources offering very diverse data format types. A set of tailormade procedures analyzing the captured observations for well defined static periods of time that can be scheduled with intense computational efficiency and compatibility with parallel processing frameworks. A set of integrated ML, AI and stochastic analysis technologies targeting the data observations for providing stability and future generations forecasts on wider or diverse market risks applying cloud computing as well as on shrinking dimensions or speed prepositions aiming for both the selection of explanatory variables and the decision making.

Fig : Financial big data management and intelligence based on computer intelligent algorithm

3.5.1. ETL Tools

Data-warehousing, data mining and OLAP-like tools are leveraged in modern data management architectures for creating financial decision supporting systems in the context of industrial technologies and commercial banking. The purely chosen data warehousing and data mining/data analyzing tools, the in-house developed ETL Tool, the OLAP-tool with respect to recent ontology enhancing applied technologies for complex customer grouping are presented. This work illustrates the need for modern techniques for signal and time series processing in the scientific bank systems. Decision support systems dealing with banking transaction data bases in commercial banks consider a very large amount of data. Those are continuously enriched with new information while in parallel growing older. This manifold lengthens or hides past patterns in data. The use of classical statistical techniques does not allow the analysis of data with respect to temporal dependency.

Such a process should be done in case of data mining or data analyzing voices/applications. In banking systems on the level of OLAP data warehousing is done with respect to monthly basis reconciliation since vendor built-in tools have been utilized. This was also important, since the time elapsed in OLAP data mining commands was on the average lower than 30 seconds, which is very fast compared to the characteristic transaction durations. However, it decentralized the actual OLAP data in more data warehouses according to payment practices. To have a better architecture and know the data origin, those data have been stored into a central data warehouse with the use of ETL processes and tools.

Two criteria have been used for the selection process of commercial tools for this newly built data warehouse. On the one hand, the best data mining, data analyzing and OLAP tools were chosen, on the other hand, the most suited ones with respect to the in-house developed ETL tools were picked out. The data mining and data analyzing tools for creating interesting/future knowledge are on the level of the data warehouse with hundreds of tools implemented. Those are Powerful Statistical and Data Mining, Data Mining Web Services and OLAP Tools which are usually called Data-mining OLAP Tools and are specified for a fixed engine and data mining methods only.

3.5.2. Data Integration Platforms

Deploying a data engineering architecture integrated with various data sources, different consumers and data types is a challenging task. Nevertheless, a plethora of products and services exist that fulfil these roles and mandates with more or less degree. Many data engineering architectures run on cloud services that speed up deployment, code adaptation, handling scalability, necessary maintenance and incline towards lower costs.

However, a deeper understanding of the place of each architecture component, their inter-connections and trade-offs can provide valuable knowledge, with which systems can be designed, even if existing components cannot be deployed.

Regarding cloud storage, it is very important to choose an appropriate bucket type. The data retrieved can be time-series data, which usually is retrieved frequently and lasts long, or it can be short-term and ad-hoc data that needs to be analysed almost instantly. On the other hand, so-called holding buckets can drop or retain low-priority partitions, whereas long-term storage options have strict data retention contracts.

Managing data pipelines and ingestion requires carefully selecting the tiers for use. From on-the-fly, fast ingestion technologies to full-fledged ELT systems, many options exist [11] that integrate multiple sources. When choosing data format, stop to think whether it is even necessary to convert to a custom one. If it is necessary, common formats exist for time-series, tabular data and geo-data. SQL engines and their ecosystems are vast. Clustering methods need to be selected with great care. When low latency is required, frameworks exist that accommodate micro-batch processing. Otherwise, thorough database design is paramount.

3.6. Machine Learning in Financial Decision Systems

Unprecedented advances in data availability, analytic techniques, and computing capabilities are transforming financial markets and open new possibilities for increasingly sophisticated decision support. Novel machine learning methods and algorithms provide unique modeling opportunities but also create challenges implicitly due to model complexity. These developments call for new, more systematic, and broader approaches and frameworks for data engineering in the financial industry. Such considerations are especially relevant for geometrically more complex data types in the financial domain, such as text or images. While financial texts constitute an important part of modern trading strategies, statisticians and algorithm researchers largely ignore the financial disclosure dimension of sentiment analysis in a forecasting context. Models specifically considering the peculiar nature of the disclosure events and financial texts are lacking, despite the fundamental differences from standard sentiment analysis tasks. Completing these challenges is a prerequisite for real-time analysis and decisions. Contemporary financial texts predict stock price movements better than conventional articles written by financial journalists. Advancements in new modeling concepts for decision systems relying on data engineering and machine learning cover major aspects of modern financial decision systems. Combinations of machine learning techniques and decision analytics ensuring accurate predictions of market-moving events are discussed. Matched with suited data engineering architectures, data preprocessing, algorithmic considerations, and concept features are elaborated. Some new proposals have been put

into practice by fintech companies, and others have been tested in reception labs via market crawlers and data managers. Five major groups of approach areas or algorithms with relevant papers are summarized. Growing literature contributions have attempted to forecast the behavior of financial time series by means of machine learning algorithms, improving on traditional time series models. Recent studies have extended classical models, not only for the actual time series, but also the extracted features forming the vectors mapped via neural networks to take into account complex interdependencies and reduce dimensionality. Advanced modeling strategies are key parts of successful decision systems, since naive forecasting often yields high out-of-sample errors and substantial short-term forecast biases.

3.6.1. Predictive Analytics

Predictive analytics refers to the use of analytical processing of historical data in an effort to infer characteristics of current or future events and incorporate that information into managerial decisions in an automated fashion. In this sense, predictive analytics are digital devices of financial decision systems which have as their output a prediction of future events together with a commentary stating why that prediction holds. Financial data is multifaceted and arises at different frequencies, from transaction records of stocks to annual disclosures of financial statements to daily news headlines. Predictive analytics commonly extracts features from financial data. Structural information features have thus far received the attention of academic research. A growing body of research, however, focuses on the lexical information present in the textual data generated by those predictors. Characterizing those features is a difficult task for traditional decision analytics techniques, which either inherently disallow automatic feature engineering or are computationally demanding. In other words, identifying the terms that are most informative about investor reactions to financial disclosures in a semi-automated manner is not a simple task. Predictive frameworks on financial disclosures, however, could substantiate otherwise costly asset-valuation analyses for institutional investors, or facilitate less sophisticated trading strategies by other actors in the market. When developing decision analytics systems, two machine learning strategies can be employed for the prediction tasks. Traditional machine learning algorithms permit the automated training of predictive frameworks once the features to be utilized as input artifacts have been defined, and the probability of each label outcome can be formulated as a function of those features.

3.6.2. Risk Assessment Models

Modern data engineering architectures have evolved rapidly over the past decades. Data has become an important resource for competitive advantage. Companies are trying to add more data storage and processing capabilities to their systems; however, these attempts often result in a monolithic architecture with multiple layers combined. This architecture can be designed with multiple layers, such as clustered database management systems, distributed storage systems, compute engines, and various tools for business intelligence, machine learning, and data science.

Most data engineers design data architecture to batch process data. In traditional batch processing systems, data is gathered within a period, stored on medium, processed later, and finally delivered. These steps produce remarkable results; however, they also become the bottleneck of modern data engineering. With low-latency architectures, data is stored in a streamed fashion, processed with event-driven calculations, and pushed to periodic companies to deliver models and analyses.

Systems for rapid data update and analysis are a necessity for today's working financial institutions. Even if older financial institutions sometimes resist the design of a new financial data processing architecture, the sheer volume of data produced by trades and the necessity of monitoring "real world" risks force such older institutions to adapt. This section will analyze a modern latency-aware data engineering architecture and applied technologies for this architecture. The design starts with a concrete business problem defined by a quant analyst from a bank's risk forecasting department. It will be piloted from the initial discussions on designing an "on-time" value-at-risk model via a high-latency architecture to a final "real-time" architecture with more discussion on tuning these architectures.

First of all, it is necessary to discuss and structure the "problem" itself and the subsequent questions. Some basic concepts will be defined, including latency in forecasting, quant finance, architectures, natural query interface. In some domain knowledge, variables that are either easily computable or nearly impossible will be defined by how easy it will be to gather the data. The problem is well-defined and procedures in quant finance, architecture layers, and background knowledge in infrastructure design will be examined before presenting the design itself.

3.7. Conclusion

The competitive edge of a financial institution hinges not only on its analytics but also on how it combines these analytics with humans and their business processes. It is about the complete financial decision support systems the institution builds and how well these systems are run. The main research question was how to design a modern data

engineering architecture to support these financial decision systems in their internal up and downstream data flows. This question is relevant to scholars in the field of data engineering and to industry engineers involved in the design of analytics systems as well.

The architecture is presented that maps onto the various components to consider in a data engineering architecture. A set of nine design choices is identified with these components elaborated upon. Finally, real-world use cases and experiences applying this architectural design are presented, complemented with a reflection on outstanding design changes anticipated in the ever-changing world of IT engineering. The architecture design builds on recent principal developments in the domains of data warehousing, big data, and data lakes. The design choices reflect academic research in the domains of data engineering and warehousing but also reflect currently trending techniques widely used in practice. The paper contributes to the design of a modern data engineering architecture for financial decision systems by providing insights into various components to consider and their implied design choices. Besides academic reflection on current developments in the architectural pillars, a concrete architectural blueprint is provided as a reference for consulting or engineering firms.

3.7.1. Emerging Trends

Financial services play a fundamental role in the functioning of modern economies, affecting both prosperity and economic growth. At the same time, financial markets are among the most complex systems, involving uncertainty and innate hazards. The sector of financial services has undergone major changes over the last 20–30 years, covering new banking, investment, and insurance products, a range of new financing tools, and new corporate finance practices. Nevertheless, these changes were accompanied by reactions, both regulatory and operational. On the path toward improved decision making, risk analysis, monitoring, and reporting, the sector started relying increasingly on new technologies, such as data mining, sampling techniques, laboratorial games, artificial intelligence, and simulation.

All branches of financial services rely extensively on computational approaches and data analytics in their major applications, such as asset management, option pricing, and risk assessment. Research on investment and finance is a very multidisciplinary domain, extending from computational mathematics to algorithmic science and its application in economics and finance. Computational technologies, concepts, and methods have never been straightforwardly applicable in finance (special) task settings. High uncertainties and behaviors represented at aggregation levels remove the explicit representation of the values and the starting intentions. Non-analytic decision rules raise severe computational problems in a dynamic environment of a huge number of heterogeneous and individualized agents.

Prescriptive (or decision) and predictive (or forecasting) systems are crucial for the effective functioning of financial services, since they assist analytical decision making in these services, offering operational guidance or assistance to decision makers. Their engineering implies the development of wide-scope analytic models, yielding solutions within acceptable times. The development of realistic and logical analytic models in such a context is rather complex. On the one hand, several properties of the context pose challenges on the analytical tractability of the models (tackling complex problems, uncertainties, regulatory requirements, and dynamism). On the other hand, the availability of massive old and present data raises scientific computational issues. Considerable computational issues arise, especially in decision support systems that require real time responses, and/or in the transformation of descriptive rules into the standard and explicit format of a prescriptive system.

References

Aladebumoye, T. (2025). *Integrating AI-Driven Tax Technology into Business Strategy*. International Journal of Research and Review, 12(1), 224–231. IJRR Journal

Konsisme (2023). *Embracing the Power of AI: Transforming Corporate Taxation*. Retrieved from https://www.konsise.com/embracing-the-power-of-ai-transforming-corporate-taxation/ konsise.com

KPMG. (2024). *AI in Tax*. Retrieved from https://kpmg.com/xx/en/what-we-do/services/ai/ai-services/ai-in-tax.html KPMG

Paulson and Partners. (2025). *AI in Tax Advisory: A Revolution in the Making*. Retrieved from https://paulsonandpartners.com/ai-in-tax-advisory-a-revolution-in-the-making/ Paulson and Partners

Wolters Kluwer. (2025). *Artificial Intelligence (AI) in Tax and Accounting*. Retrieved from https://www.wolterskluwer.com/en/know/artificial-intelligence-tax-accounting