Dr. Bayomock Linwa Andre Claude, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

An AI-Driven Budget Revenue Projection System for Public Finance: Architecture, Data Model and Implementation

Dr. Bayomock Linwa Andre Claude
Head of Computer Science Department,
International University of Grand-Bassam bayomock.a@iugb.edu.ci
Mr. Traore Cheick Oumar
Bachelor of Computer Science, International University of Grand-Bassam

Abstract- Determining the budget revenue in big organizations, multinational company or government institutions, is a difficult tasks considering incertitude's of some input items. In Sub-Saharan Africa countries, many governments still rely on manual, spreadsheet-based methods that are time- consuming, inconsistent, and error-prone to establish budget projections. This paper presents the design and implementation of a web-based budget revenue projection protocol system using artificial intelligence, developed for a general public organization responsible for national budget management. The goal of this research is to minimize the production time of the budget projection, minimize the projected data' incertitude. The proposed protocol automates the projection of state budget revenues using statistical models (SARIMAX) and AI-driven analysis (Prophet), replacing previously manual forecasting methods.

Index Terms- Budget Revenue, Projection, Protocol, Artificial Intelligence, SARIMAX, Prophet, N-tier Architecture, Data Model, Public Finance.

I. INTRODUCTION

Budget revenue forecasting determines future expenses and revenues of an organization. It guides financial experts on the process activities of the budget estimation as financial planning, using historical data, market trends, and economic drivers to predict performance. It informs executive managers on budgeting, hiring, and investment metrics, and then, allow them to take decisions on evidence-based projections data. Key methods include linear regression, time-series analysis, and pipeline modeling, often updated through continuous monitoring.

Projection budget results allow the government to determine allocated budget of each of ministry department. The quality of data projection should be trustful by minimizing the margin error between the real data and the forecast data. However, the budget projection process, in many Africa countries, is still determined manually, and this increase considerably the budget margin error. The main budget forecast tool used by financial experts in Africa is the Microsoft EXCEL tool and generally without the activation of the macro function.

In Côte d'Ivoire, the budget framework process, also called "cadrage budgetaire", is the central mechanism through which governments plan future revenue projections. It follows a well- defined workflow involving macroeconomic framework transmission from the Ministry of Economy and Finance, framework creation with projection horizons, distribution to financial departments (revenue collection agencies), data submission of historical and projected values, and consolidation into a unified document.

However, several significant challenges hinder this process: the manual nature of projections makes them time-consuming and inconsistent; there is no system capable of automatically analyzing historical revenue data; the existing data models do not sufficiently detail budget lines, their associated factors, or the hypotheses used for projections; there is no scenario simulation capability; and consolidating projections from multiple agencies is cumbersome.

To address these gaps, this paper proposes a comprehensive web-based system that automates budget revenue projections using artificial intelligence. The system integrates data collection, AI-driven analysis, and interactive visualization to enhance the accuracy, speed, and reliability of the forecasting process. It serves multiple stakeholders by offering relevant information and functionalities tailored to their roles.

This paper is structured as follows: after the introduction, the motivations behind this work are explained, followed by a review of related work. Then, the system requirements are presented, followed by the data model description. Afterwards, the architectural solution is described, the AI forecasting approach is detailed, and the implementation results are presented. Finally, the conclusion is given.

II. MOTIVATIONS

The motivations behind this research are classified into four main categories that justify the implementation of an AI-driven budget revenue projection system.

2.1 Operational

Budget revenue projections relied heavily on spreadsheets and the personal expertise of agents, making the process time-consuming and inconsistent. There was no system capable of automatically analyzing historical revenue data and producing projections based on statistical models. The existing methods did not allow analysts to easily simulate different budget scenarios by adjusting projection factors.

2.2 Data Structuring

The existing data model did not sufficiently detail budget lines, their associated factors, or the hypotheses used for projections. A well-structured data model is essential for ensuring traceability and coherence in the projection process.

2.3 Consolidation

Aggregating projections from multiple financial agencies into a unified view was a cumbersome manual process. A centralized platform is needed to consolidate projections from all departments efficiently.

2.4 Technological

In a context marked by the emergence of new technologies, increasingly powerful data analysis and processing tools are now available, particularly thanks to Artificial Intelligence. Integrating these technologies into the public finance ecosystem represents a major opportunity to improve the performance and productivity of budget services.

III. RELATED WORK

Recent years have seen growing interest in applying artificial intelligence and machine learning to financial forecasting and public finance management. This literature review is organized around key conceptual streams: time series forecasting methods, AI applications in public finance, and budget management systems.

Time series forecasting has been extensively studied using both classical statistical methods and modern machine learning techniques. The ARIMA family of models, including SARIMAX (Seasonal AutoRegressive Integrated Moving Average with eXogenous factors), has been widely used for economic and financial forecasting (Seabold & Perktold, 2010). SARIMAX extends the traditional ARIMA model by incorporating seasonal patterns and external variables, making it particularly suitable for budget revenue data that may be influenced by macroeconomic indicators.

Prophet, developed by Meta (Taylor & Letham, 2018), offers an alternative approach designed for business time series forecasting. Prophet handles missing data, trend changes, and outliers effectively, making it suitable for real-world financial data. Its ability to incorporate domain knowledge through configurable parameters aligns with the needs of budget analysts who possess expert knowledge about revenue trends.

In the domain of public finance, several countries have developed information systems for budget management. However, most existing systems focus primarily on budget execution tracking rather than predictive analytics. The integration of AI-driven forecasting into budget management platforms remains limited, particularly in Sub-Saharan Africa where digital infrastructure and data availability present additional challenges.

The authors Nguyen, L. & Lee, J. (2021) examine the limitations of manual budgeting and forecasting processes. It distinguishes between budgets (plans for achieving goals) and forecasts (short-term snapshots of expected outcomes), arguing that forecasts are often more valuable due

to their adaptability. The paper highlights key disadvantages of manual methods: they are time-consuming, prone to human error, lack data security, and cannot adapt quickly to changing economic conditions. The authors advocate for rolling forecasts (monthly/quarterly) over static annual budgets, noting that manual approaches create a deliberate focus on profitability at the expense of customer value. The paper directly supports the article's motivation for automating the forecasting process through AI-driven tools.

This comprehensive OECD (2019) report presents good practice principles for performance budgeting across member countries. It examines how governments can link budget allocations to measurable results and outcomes. The report covers the evolution of performance budgeting frameworks, the role of spending reviews, and the importance of program evaluation. It emphasizes that modern budgeting should incorporate data-driven decision-making and performance metrics rather than relying solely on incremental approaches. The report provides international benchmarks for budget management systems, which serve as a reference framework for the article's proposed system.

The Ministry of Finance, Ghana (2019). manual details Ghana's public finance management (PFM) cycle, including program-based budgeting, policy formulation, the Medium-Term Expenditure Framework (MTEF), and budget execution. It provides a comprehensive operational framework covering budget preparation, performance indicators, costing programs, and the relationship between policy and budget formulation. The manual illustrates how a Sub-Saharan African country structures its budget cycle, from policy formulation through execution to evaluation, and highlights the challenges of implementing systematic budget management in developing contexts.

The AF-TERG (2024) guidance notes from the Adaptation Fund's Technical Evaluation Reference Group addresses the principles and practices of evaluation budgeting in international development contexts. It covers when and how to develop evaluation budgets, the relationship between evaluation planning and financial allocation, and best practices for ensuring that evaluation activities are adequately funded. While focused on climate adaptation projects, the methodological principles of structured budgeting, resource allocation, and evidence-based financial planning are broadly applicable to public finance management.

Prakash, K.O., Abdullah, and al (2024) research examines how budgeting and forecasting impact strategic decisions and financial health at National Finance Oman. Using both quantitative surveys and qualitative interviews, the research reveals that financial management plans and precise forecasting help manage economic fluctuations, allocate resources more effectively, and achieve sustainable financial outcomes. Critically, the study finds that current forecasting models need to better adapt to market changes and unforeseen events. The authors explicitly recommend employing AI-driven forecasting and cloud-based financial management systems to enhance accuracy and efficiency- a recommendation that directly aligns with the article's proposed solution.

In Sahu, N. & Metta, A. (2025), the authors examine budgeting and forecasting practices in 40 small and medium-sized enterprises (SMEs) associated with Chartered Accountants in India. Key findings include: 75% of firms use incremental budgeting, trend-based forecasting dominates at 70%, but AI-based tools are gaining traction at 22% adoption. Major challenges identified are inaccurate/delayed data (62%), market volatility (58%), and limited expertise (50%). The study finds that CA-guided firms demonstrate better financial discipline and that digital transformation is reshaping budgeting models. The authors recommend adopting advanced budgeting techniques (rolling budgets, zero-based budgeting) and AI-based forecasting tools.

The Olamijuwon, O.J. & Zouo, S.J.C. (2024) paper explores the conceptual integration of machine learning and data analytics into public health budget forecasting. It discusses regression models, decision trees, and neural networks as ML approaches for budget prediction, and highlights the benefits of real-time monitoring and dynamic resource allocation. The paper addresses ethical considerations including data privacy, algorithmic bias, and equitable resource distribution. While focused on health budgets, the methodological framework — using ML models to replace static, incremental budgeting — is directly applicable to general public finance forecasting. The paper emphasizes that ML can continuously learn from data, improving accuracy over time, which aligns with the article's SARIMAX and Prophet approach.

In Kruger, J. (2017), the fundamentals of World Bank planning and budgeting is presented, It is emphasizing the connection between financial planning and organizational creditworthiness. It introduces key concepts: the importance of knowing your current position before planning, the need for synchronized financial and physical planning, and the roadmap to a Capital Improvement Plan (CIP). The presentation establishes that effective planning requires both understanding current data and projecting future needs — principles that underpin the article's AI-driven projection system

While these studies offer substantial advancements, there remain critical gaps. Few systems integrate time series forecasting, exogenous factor management, role-based workflows, and interactive visualization within a single platform tailored to the operational realities of public finance in developing regions. Building upon these insights, the current paper proposes a comprehensive platform that addresses these gaps.

IV. REQUIREMENTS

4.1 Functional Requirements

The system must address the following functional requirements organized by service domain:

1) User Authentication and Access Control

- FR1 - Authentication: The system must allow users to authenticate using a unique username and a secure password in order to access the application.
- FR2 - Authorization: The system must control access to functionalities based on user roles, ensuring that each user can only access features corresponding to their permissions.
- FR3 - Session Management: The system must manage user sessions using JWT-based authentication and should ensure secure session handling, including proper logout mechanisms.
- FR4 - User Management: The system must allow administrators to create, update, and delete user accounts, as well as manage their roles and access rights.

2) Budgetary YearManagement

- FR5 - Create Budget Exercises: The system must allow authorized users to create new budget exercises by providing a unique code and a descriptive label.
- FR6 - List Budget Exercises: The system should provide a complete list of all existing budget exercises to facilitate consultation and management.
- FR7 - Update Budget Exercises: The system must allow users to modify the information

of existing budget exercises when necessary.

- FR8 - Delete Budget Exercises: The system should allow authorized users to delete budget exercises while ensuring data consistency and integrity.

3) Budget Framework Management

- FR9 - Create Frameworks: The system must allow users to create budget frameworks associated with a specific budgetary year.
- FR10 - Configure Projection Horizon: The system must allow users to define the projection horizon by specifying the number of years to be considered for forecasting.
- FR11 - Framework Status Management: The system must manage the lifecycle of a framework by allowing it to be set to different statuses, such as EN_ELABORATION or FERME, depending on its progression.
- FR12 - Analyst Notification: The system should notify analysts when a framework becomes available, enabling them to start working on projections.

4) Budget Line(Rubrique) Management

- FR13 - Create Categories: The system must allow users to create Budget Lines by defining a code, a label, a type, and hierarchical relationships.
- FR14 - Hierarchical Structure: The system must support a parent-child hierarchy between categories in order to represent structured financial data.
- FR15 - Department Assignment: The system must allow each Budget Lineto be assigned to a specific financial department responsible for its management.

5) Projection Factor Management

- FR16 - Define Factors: The system must allow users to define projection factors by specifying a code, a label, and a unit of measurement.

- FR17 - Link Factors to Categories: The system must allow factors to be associated with specific Budget Lines in order to influence projections.
- FR18 - Manage Factor Values: The system must allow users to input and manage both historical and projected values of factors for each year.

6) AI Projection Generation

- FR19 - Generate Projections: The system must generate projections using statistical and forecasting models such as SARIMAX or Prophet, based on historical data.
- FR20 - Use Exogenous Factors: The system should incorporate external or exogenous factors into the projection models to improve the accuracy of forecasts.
- FR21 - Display Reliability: The system should display reliability indicators and observations associated with each projection to support decision-making.
- FR22 - Manual Adjustment: The system must allow users to manually adjust projected values when necessary, in order to incorporate expert judgment.

7) Consolidation and Reporting

- FR23 - Consolidated View: The system must provide a consolidated view of projections across all financial entities or agencies.
- FR24 - Data Visualization: The system should provide interactive dashboards with charts to visualize both historical and projected data.
- FR25 - Data Export: The system must allow users to export projection data in order to facilitate reporting and external analysis.

V. DATA MODEL

The data model for this project is designed to support the management, projection, and consolidation of budget revenues. The model is implemented in MySQL and captures detailed information about budget exercises, frameworks, budget lines, factors, hypotheses, and users.

5.1 Entities and Their Roles

The data model includes the following key entities:

- ExerciceBudgetaire (Budget Exercise): Represents a fiscal year with code, label, and status.
- Cadrage (Budget Framework): Defines the projection period, horizon, and lifecycle status linked to an exercise.
- Rubrique (Budget Line): Represents a budget category with hierarchical parent-child relationships, assigned to a department.
- TypeRubrique (Category Type): Classifies budget lines (e.g., tax revenues, non-tax revenues).
- Regie (Financial Department): Represents revenue collection agencies (e.g., Directorate of Taxes, Directorate of Customs).
- Facteur (Projection Factor): External variables influencing projections (e.g., GDP, inflation, rainfall).
- FacteurValeur (Factor Value): Stores factor values for each year and budget line combination.
- Hypothese (Projection Hypothesis): Stores AI-projected values, manual overrides, reliability scores, and observations.
- RecetteHistorique (Historical Revenue): Records actual revenue amounts per year per budget line.
- Utilisateur (System User): User accounts with roles and department assignments.

5.2 Conceptual Data Model

The conceptual data model illustrates the relationships between entities. The central entity is the Rubrique (Budget Line), which is linked to a Cadrage (Framework), a TypeRubrique (Type), and a Regie (Department). Each Rubrique can have multiple Hypotheses (projections), RecetteHistoriques (historical data), and FacteurValeurs (factor values).

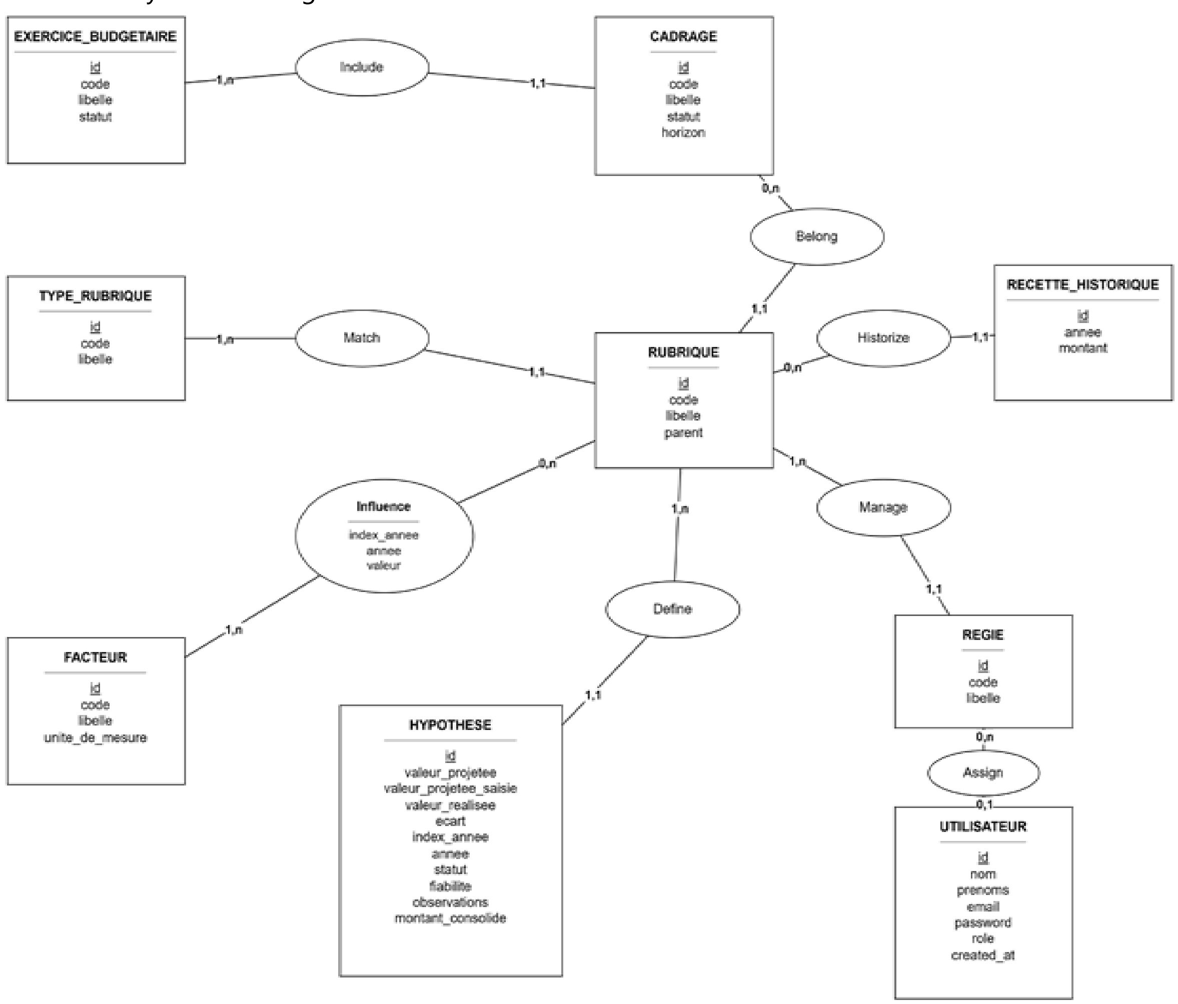

Figure 1: Conceptual Data Model

5.3 Data Dictionary

The following table presents the key entities with their attributes and constraints:

Field	Description
Id	Unique identifier of the budget exercise
Code	Short code for the exercise (e.g., "EX-2026")
Libelle	Year label of the exercise (e.g., 2026)
Statut	Current status of the exercise: under development or closed

Table 1 : Exercice Budgétaire (Budget Exercise)

Field	Description
Id	Unique identifier of the budget framework
Code	Short code for the framework
Libelle	Full name of the framework (e.g., "Cadrage DPBEP 2026-2030")
Statut	Current status: under development or closed
Horizon	Number of projection years

Table 2: Cadrage (Budget Framework)

Field	Description
Id	Unique identifier of the Budget Lines
Code	Short code for the category
Libelle	Full label of the Budget Lines
Parent	Reference to the parent category for hierarchical organization

Table 3: Rubrique (Budget Lines)

Field	Description
Id	Unique identifier of the category type
Code	Type code
Libelle	Type label (e.g., "Recettes fiscales")

Table 4: Type Rubrique (Category Type)

Field	Description
Id	Unique identifier of the financial department
Code	Department code (e.g., "DGI")
Libelle	Full name of the department

Table 5: Régie (Financial Department)

Field	Description
Id	Unique identifier of the factor
Code	Unique factor code
Libelle	Factor label (e.g., "PIB nominal")
unite_de_mesure	Measurement unit (e.g., "Milliards FCFA", "Pourcentage")

Table 6: Facteur (Exogenous Factor)

Field	Description
Id	Unique identifier of the factor value entry
index_annee	Year index within the framework (1, 2, 3...)
Annee	Absolute year value (e.g., 2027)
Valeur	Numeric value of the factor for the given year

Table 7: Facteur Valeur (Factor Value)

Field	Description
Id	Unique identifier of the hypothesis
valeur_projetee	AI-computed projected revenue amount
valeur_projetee_saisie	Manually entered projected value by the analyst (overrides AI value)
valeur_realisee	Actual realized revenue after the year ends
Ecart	Deviation between projected and realized values
index_annee	Year index within the projection (1, 2, 3...)
Annee	Absolute year (e.g., 2027)
Statut	Current status: projected, manually entered, or pending
Fiabilite	AI reliability score as a percentage (0-100)
Observations	AI-generated textual observations about the projection
montant_consolide	Final consolidated amount after central review

Table 8: Hypothèse (Projection Hypothesis)

Field	Description
Id	Unique identifier of the historical revenue record
Annee	Year of the historical revenue (e.g., 2020)
Montant	Revenue amount in FCFA

Table 9: Recette Historique (Historical Revenue)

Field	Description
Id	Unique identifier of the user account
Nom	Last name of the user
Prenoms	First name(s) of the user
Email	Email address used for login
Password	Encrypted password
Role	User role: Administrator, Analyst, or Central Agent
created_at	Account creation timestamp

Table 10: Utilisateur (System User)

VI. AI BUDGET FORECAST PROTOCOL

6.1 Protocol Description

The AI forecasting engine is a Python-based microservice (FastAPI) that receives historical budget revenue data and external factors from the Quarkus backend, processes them through either SARIMAX (Statsmodels) or Prophet (Meta), and returns projected values with reliability scores and observations.

Step-by-Step Protocol

Step 1 - User triggers projection

The Department Analyst selects a budget line (rubrique), chooses a forecasting model (PROPHET or STATSMODELS), and clicks the "Generate Projection" button on the React frontend.

Step 2 - Frontend sends request to backend

The React frontend sends an HTTP POST request to the Quarkus backend (e.g., POST /api/projections/generate). The request contains: the rubrique ID, the selected model type, and optionally, future factor hypotheses entered by the analyst.

Step 3 - Backend assembles historical data

The Quarkus backend retrieves from the MySQL database:

- Historical revenues (RecetteHistorique) for the selected rubrique — typically 2015 to 2025
- Associated projection factors (FacteurValeur) — e.g., GDP growth, inflation rate, rainfall index

- The projection horizon from the active budget framework (Cadrage) — e.g., 5 years
- Future factor hypotheses (if provided by the analyst)

Step 4 - Backend calls the AI microservice

The Quarkus backend constructs a JSON payload conforming to the PredictRequest schema and sends an HTTP POST request to the Python AI microservice at POST /predict. The payload includes:

- modelType: "PROPHET" or "STATSMODELS"
- y: list of historical data points [{ds: "2015-01-01", y: 50000000}, ...]
- exogenous: dictionary of factor series {"PIB": [{ds: ..., y: ...}], "INFLATION": [...]}
- horizon: number of future years to predict (e.g., 5)
- futureExogenous: analyst-provided hypotheses for future factor values (optional)

Step 5 - AI microservice preprocesses data

The Python microservice performs the following preprocessing:

1. Converts the historical series (y) into a Pandas DataFrame sorted by date
2. Merges exogenous factor data onto the historical DataFrame by date, using forward-fill for missing values
3. Builds a future date DataFrame (e.g., 2026-01-01 to 2030-01-01)
4. Applies future exogenous values: uses analyst hypotheses if available, otherwise extends the last known value

Step 6A - Prophet model execution (if modelType = PROPHET)

If Prophet is selected:

5. A Prophet model is instantiated with default configuration (linear growth, additive seasonality)
6. Each exogenous factor is added as a regressor via model.add_regressor()
7. The model is fitted on the training data (historical revenues + aligned factors)
8. A future DataFrame (with exogenous columns populated) is passed to model.predict()
9. Prophet returns yhat (predicted value), yhat_lower and yhat_upper (80% confidence interval)

Step 6B - SARIMAX model execution (if modelType = STATSMODELS)

If STATSMODELS is selected, the process is more defensive:

10. Exogenous variable limitation: maximum n_observations / 4 factors are retained to avoid overfitting (e.g., 10 years → max 2 factors)
11. Normalization: exogenous data is standardized (mean=0, std=1) so that variables of different scales (GDP in billions vs. rainfall in mm) are treated equally
12. Adaptive order selection:
    - With factors → SARIMAX(1,1,1) — captures autoregressive patterns and factor correlations
    - Without factors → SARIMAX(0,1,0) — simple linear drift projection
13. Fixed parameters: seasonal_order=(0,0,0,0) (no seasonality — annual data), trend='c' (constant/drift), enforce_stationarity=False, enforce_invertibility=False
14. Model fitting: model.fit(disp=False) — silent optimization
15. Forecasting: get_forecast(steps=horizon, exog=exog_future) with 80% confidence interval (alpha=0.2)
16. Sanity check: each predicted value is verified — if any value is negative, exceeds 500% of the last known value, or drops below 10%, the model is considered divergent
17. Fallback mechanism: if divergence is detected, the model automatically falls back to SARIMAX(0,1,0) without factors (simple linear drift), ensuring a reasonable prediction is always returned

Step 7 - Reliability calculation

For each predicted year, a reliability score (0–100%) is computed:

$\text{fiabilité} = (1 - (\text{upper} - \text{lower}) / |\hat{y}|) \times 100$

Where upper and lower are the confidence interval bounds and $\hat{y}$ is the predicted value. A narrow interval relative to the prediction yields high reliability; a wide interval yields low reliability.

Step 8 - Observation generation

The AI automatically generates textual observations for each projected year:

- "Fiabilité élevée" — if reliability ≥ 80%
- "Fiabilité faible" or "Incertain" — if reliability < 50%

- "Hausse prévue (+X%)" — if year-over-year growth exceeds +5%
- "Baisse prévue (−X%)" — if year-over-year decline exceeds −5%
- "Facteurs ignorés (modèle instable)" — if the SARIMAX fallback was triggered

Step 9 - AI microservice returns response

The AI microservice returns a JSON response containing a list of ForecastPoint objects, each with: ds (date), yhat (predicted value), fiabilite (reliability score), and observations (textual notes).

Step 10 - Backend stores results

The Quarkus backend receives the AI response and stores each projection as a Hypothese entity in the MySQL database, linked to the rubrique, cadrage, and exercise. The status is updated to "PROJETE" (projected).

Step 11 - Frontend displays results

The React frontend receives the updated data and displays: the projected values in a data table with reliability scores and observations, interactive Chart.js visualizations (bar charts for historical data, line charts for projections), and the option for manual override (saisie manuelle) if the analyst disagrees with the AI prediction.

6.2 Activity Sequence

The sequence diagram illustrates the chronological sequence of interactions between the various actors and the system during the budgeting and forecasting process.

It highlights the order of exchanges, the responsibilities of the actors, and the processing performed by the system.

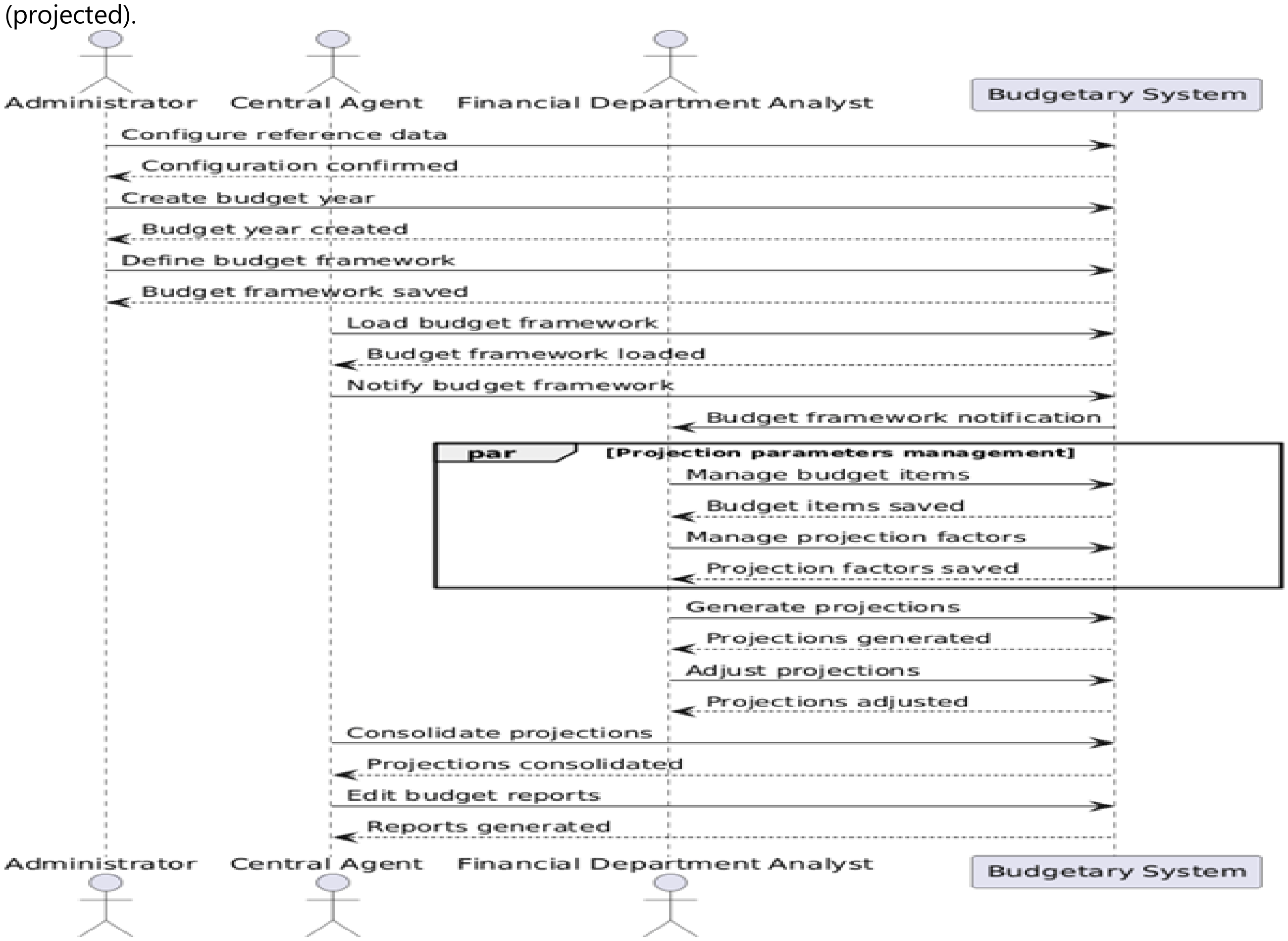

Figure 2: Sequence Diagam

VII. ARCHITECTURE SOLUTION

The architecture chosen for the system is an N-tier architecture organized around four main layers, ensuring separation of responsibilities, maintainability, scalability, and system evolution.

7.1 Infrastructure Architecture

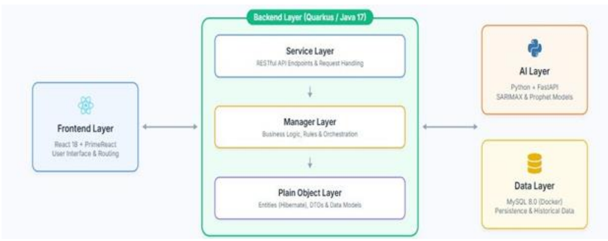

Figure 3: Infrastructure Architecture Diagram

7.2 Layer Description

Presentation Layer (Frontend): Built with React 18, TypeScript, and PrimeReact, providing a modern, responsive interface. It handles data entry, visualization, and user input validation. No complex business logic resides in this layer; processing is delegated to the backend via API calls.

Application Layer (Backend): Built with Quarkus/Java, exposing RESTful APIs and processing business logic. It is responsible for business rules processing, management of budgetary processes, securing exchanges, communication with the database, and orchestration of calls to the AI modules.

Data Layer: MySQL 8.0 deployed via Docker, ensuring data persistence, integrity, traceability, and consistency.

AI Layer: Python/FastAPI microservice with Statsmodels (SARIMAX) and Prophet for time series forecasting. This layer functions as a specialized service interfaced with the application layer.

7.3 Deployment architecture

Figure 4: Deployment diagram

7.4 Technology Stack

Layer	Technology	Purpose
Frontend	React 18 + TypeScript	UI framework
	PrimeReact + PrimeFlex	Component library
	Axios	HTTP client
	Chart.js	Data visualization
Backend	Java 17	Programming language
	Quarkus 3.x	Application framework
	Hibernate ORM (Panache)	Database ORM
Database	MySQL 8.0	Relational database
	Docker	Container platform
AI	Python 3 + FastAPI	AI service framework
	Statsmodels (SARIMAX)	Statistical forecasting
	Prophet	Alternative forecasting

Table 11: Technologie Summary Table

VIII. USE CASE

8.1 Use Case Diagram

The use case diagram describes the system's functionalities from the users' perspective. It highlights the actors, their interactions with the system, and the various services provided. This diagram offers a comprehensive view of functional requirements and facilitates understanding user expectations.

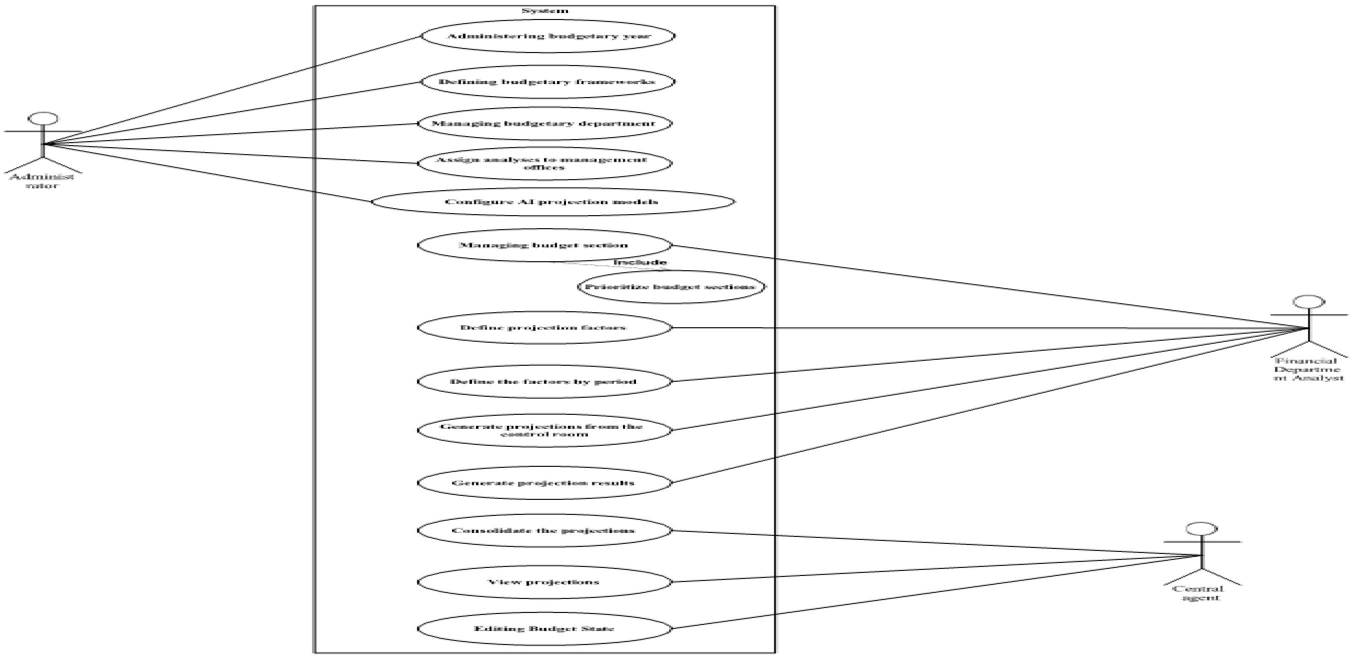

Figure 5: Use Case Diagram

8.2 Description of some Use cases:
UC1

	14. The system returns a confirmation dialog with "Yes" and "No" buttons. 15. In case the administrator selects "Yes", the system deletes the record from the database. 16. The operation is canceled if the administrator selects "No". 17. The system returns to the previous state.
Alternate flows	1. The administrator enters a code that already exists. 2. The system displays an error message "Code already exists, please try another one". 3. The administrator can either try again or cancel

	the operation by clicking on the "Annuler" button. 4. If the administrator clicks on "Annuler", the system returns back to the data table.
Post-condition	If the use case was successful, a new Budgetary Yearis created, updated, or deleted from the system. Otherwise, the state of the system is unchanged by the use case.
Exceptions	• The administrator aborts the create/edit operation by clicking on "Annuler" button in the dialog form. • The administrator aborts the delete operation by clicking on "No" button at step 16 of the Normal Flow.

	18. The system returns a confirmation dialog with "Yes" and "No" buttons. 19. In case the administrator selects "Yes", the system deletes the record from the database. 20. The operation is canceled if the administrator selects "No". 21. The system returns to the previous state.
Alternate flows	5. The administrator enters a code that already exists. 6. The system displays an error message "Code already exists, please try another one". 7. The administrator can either try again or cancel the operation by clicking on the "Annuler" button. 8. If the administrator clicks on "Annuler", the system returns back to the data table.
Post-condition	If the use case was successful, a new Budgetary Yearis created, updated, or deleted from the system. Otherwise, the state of the system is unchanged by the use case.
Exceptions	• The administrator aborts the create/edit operation by clicking on "Annuler" button in the dialog form. • The administrator aborts the delete operation by clicking on "No" button at step 16 of the Normal Flow.

Table 12: Administering Budgetary Year

UC2

Use case Name	**Defining Budgetary Frameworks**
Primary actor	Administrator
Secondary actor	Application System
Brief description	This use case allows the administrator to create, view, update, and delete budgetary frameworks (cadrages) linked to a specific budget exercise. A framework defines the projection period and horizon.
Pre-condition	1. The administrator is logged into the application system. 2. At least one Budgetary Year exists in the system. 3. The application system displays the main dashboard with the sidebar navigation.
Normal flows	1. The administrator clicks on "Données de Référence" in the sidebar navigation. 2. The system expands the sub-menu and the administrator selects "Cadrages". 3. The system returns a data table containing all existing frameworks with columns: Code, Libellé, Exercice, Statut, and Horizon, plus action buttons (Edit, Delete) on each row and a "Nouveau" button at the top left.

	4. To CREATE: The administrator clicks on "Nouveau" button. 5. The system returns a dialog form containing the following fields: • Code (text input, required) • Libellé (text input, required, e.g., "Cadrage DPBEP 2026-2030") • Exercice Budgétaire (dropdown list of existing exercises, required) • Statut (dropdown: EN_ELABORATION / FERME) • Horizon (numeric input, required, default: 5) 6. The administrator fills the information, then clicks on "Enregistrer". 7. The system validates the information, creates the framework record, and returns a successful creation message. 8. The system adds the new framework to the data table. 9. To EDIT: The administrator clicks on the "Edit" button on the corresponding row. 10. The system returns the dialog form pre-filled with existing data. 11. The administrator modifies the desired fields and clicks "Enregistrer". 12. The system updates the record and returns a success message. 13. To DELETE: The administrator clicks on the "Delete" button. 14. The system displays a confirmation dialog with "Yes" and "No". 15. In case the administrator selects "Yes", the system deletes the framework and all associated data. 16. The operation is canceled if the administrator selects "No". 17. The system returns to the previous state.
Alternate flows	1. The administrator tries to create a framework but no Budgetary Yearexists. 2. The system displays an empty dropdown for the exercise field. 3. The administrator must first create a Budgetary Yearbefore creating a framework. 4. The administrator can cancel the operation by clicking on "Annuler".
Post-condition	If the use case was successful, a new budgetary framework is created, updated, or deleted from the system. The framework is linked to the selected budget exercise. Otherwise, the state of the system is unchanged.
Exceptions	• The administrator aborts the operation by clicking on "Annuler" button. • The administrator aborts the delete operation by clicking on "No" at step 16 of the Normal Flow. • No Budgetary Yearexists in the system (creation is blocked).

Table 13: Defining Budgetary Frameworks

UC3

Use case Name	**Managing Budgetary Department**
Primary actor	Administrator
Secondary actor	Application System
Brief description	This use case allows the administrator to create, view, update, and delete financial agencies (régies financières) such as DGI, DGD, and DGT. A Financial departement represents a revenue collection authority.
Pre-condition	1. The administrator is logged into the application system. 2. The application system displays the main dashboard with the sidebar navigation.
Normal flows	1. The administrator clicks on "Données de Référence" in the sidebar navigation. 2. The system expands the sub-menu and the administrator selects "Régies". 3. The system returns a data table containing all existing financial agencies with columns: Code and Libellé, plus action buttons (Edit, Delete) on each row and a "Nouveau" button at the top left. 4. To CREATE: The administrator clicks on "Nouveau" button. 5. The system returns a dialog form containing the following fields: • Code (text input, required, e.g., "DGI") • Libellé (text input, required, e.g., "Direction Générale des Impôts") 6. The administrator fills the information, then clicks on "Enregistrer". 7. The system validates the information, creates the Financial departement record, and returns a successful creation message. 8. The system adds the new Financial departement to the data table. 9. To EDIT: The administrator clicks on the "Edit" button on the corresponding row. 10. The system returns the dialog form pre-filled with existing data. 11. The administrator modifies the fields and clicks "Enregistrer". 12. The system updates the record and returns a success message. 13. To DELETE: The administrator clicks on the "Delete" button. 14. The system displays a confirmation dialog with "Yes" and "No". 15. In case the administrator selects "Yes", the system deletes the Financial departement record from the database. 16. The operation is canceled if the administrator selects "No". 17. The system returns to the previous state.
Alternate flows	1. The administrator enters a code that already exists. 2. The system displays an error message "Code already exists". 3. The administrator can either

	try again with a different code or cancel the operation by clicking on "Annuler". 4. If the administrator clicks "Annuler", the system returns back to the data table.
Post-condition	If the use case was successful, the financial agency is created, updated, or deleted from the system. Otherwise, the state of the system is unchanged by the use case.
Exceptions	• The administrator aborts the operation by clicking on "Annuler". • The administrator aborts the delete operation by clicking on "No". • The Financial departement cannot be deleted if it is referenced by existing rubriques or users.

Table 14: Managing Budgetary Department

IX. AI FORECASTING APPROACH

The projection module is based on the analysis of time series of budget revenues. A time series is a sequence of values observed at regular intervals over time, for example the annual revenue amounts of a budget line.

9.1 SARIMAX Model

The SARIMAX (Seasonal AutoRegressive Integrated Moving Average with eXogenous factors) model adapts its configuration based on available data. Without factors, SARIMAX(0,1,0) with linear drift is used for a realistic linear projection. With factors, SARIMAX(1,1,1) with exogenous variables captures correlations between budget revenues and external factors such as GDP or inflation.

The model includes automatic sanity checks that verify whether predictions are reasonable (within 10% to 500% of the last observed value). If the primary model diverges, a fallback to SARIMAX(0,1,0) without factors is automatically applied.

9.2 Prophet Model

Prophet, developed by Meta, is used as an alternative forecasting model. It handles trend changes and outliers effectively. External factors (regressors) can be added to improve prediction accuracy. Prophet generates confidence intervals that are used to compute reliability scores.

9.3 Reliability Calculation

For each projection, a reliability percentage (0-100%) is calculated using the formula:

fiabilite = (1 - (upper - lower) / |yhat|) x 100

Where upper and lower are the confidence interval bounds and yhat is the predicted value. Additionally, AI-generated observations provide contextual information such as "High reliability," "Expected growth +8.3%," or "Factors ignored (unstable model)."

9.4 Exogenous Factor Integration

The system normalizes exogenous variables to prevent scale-related issues and limits the number of variables to max(1, n_observations/4) to avoid overfitting. When analysts provide future hypotheses for factors, these values are used directly; otherwise, the last known value is extended.

X. IMPLEMENTATION RESULTS

The system was successfully developed and tested with real budget revenue data from 2015 to 2025, with projections generated for 2026 to 2030. The following sections present key screenshots of the implemented system.

10.1 Login and Authentication

A clean authentication form with username/password fields, application logo, and role- based redirection after login.

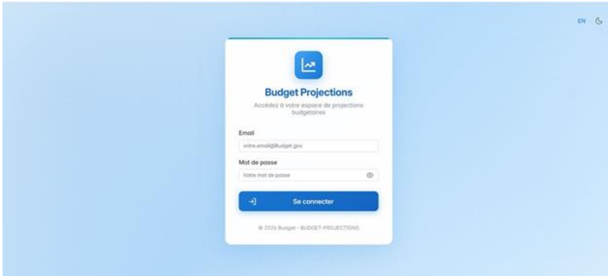

Figure 6: Login Page

10.2 Administration Interface

The administrator can manage budget exercises, frameworks, departments, budget lines, and user accounts. The interface provides CRUD operations with data validation.

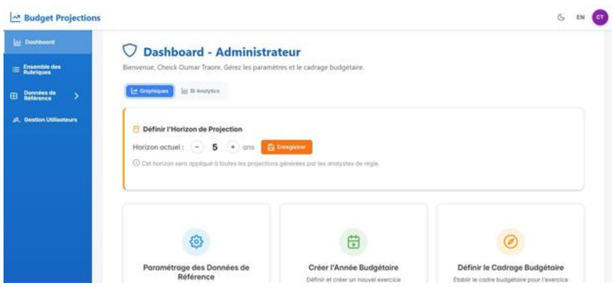

Figure 7: Administrator Main Page

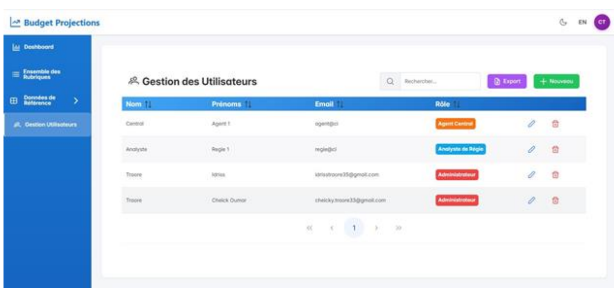

Figure 8: User Management Page

10.3 Reference Data Management

The analyst can configure the reference data used for projections, including budget exercises, frameworks, budget lines, factors, and historical revenue data.

Figure 9: Reference Data Management

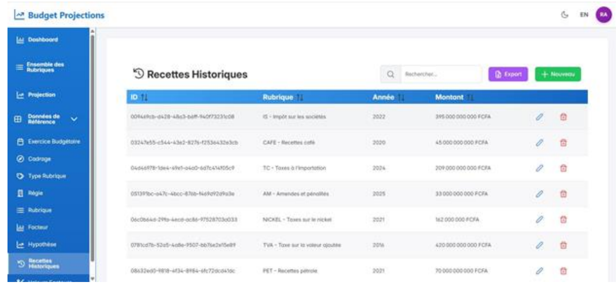

Figure 10: Historical Revenue List

10.4 Department Analyst Dashboard

The dashboard displays combined bar and line charts showing historical amounts (bars) and projected values (line) over time, for total projected revenue and growth rate, a pie chart for revenue distribution by department, and trend comparison charts.

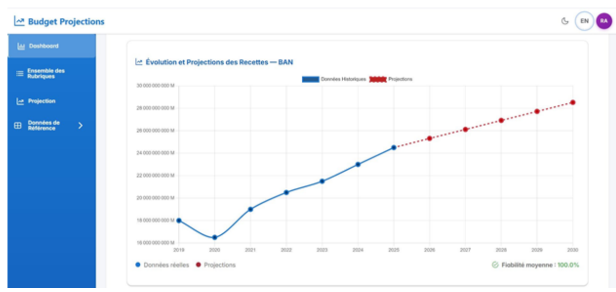

Figure 11: Department Analyst Dashboard

9.5 AI Projection Generation

The core page allows the analyst to view budget lines assigned to their department, trigger AI projection generation, view results with reliability scores and observations, and manually adjust projected values.

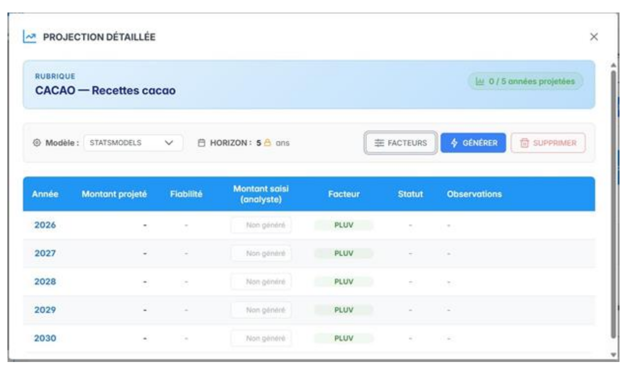

Figure 12: Projection Generation Page

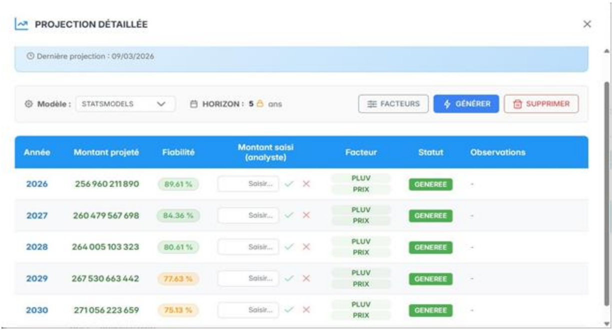

Figure 13: Generated Projections with Reliability Scores

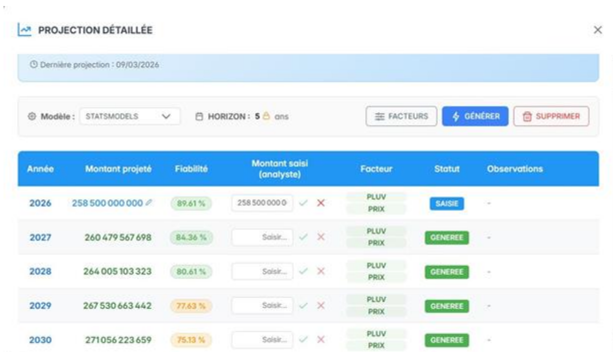

Figure 14: Manual Override of Projected Value

10.6 Consolidation

Accessible to the Central Agent, the consolidation page presents a unified view of all projections across agencies with summary tables, aggregated values, and status indicators. Once consolidated, budget lines are locked from further modification.

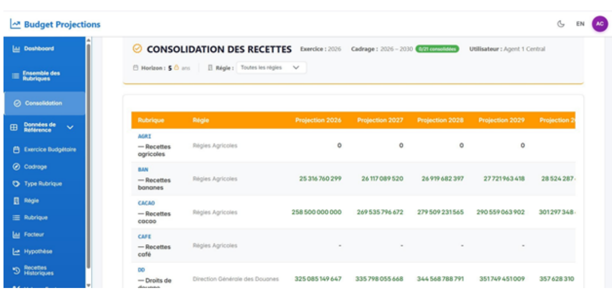

Figure 15: Consolidation Page

Figure 16: Dashboard Analysis Charts

XI. CONCLUSION

This paper presented the design and implementation of a complete budget revenue projection system using artificial intelligence. The project successfully addressed the identified challenges in public finance forecasting:

- A structured data model captures exercises, frameworks, categories, factors, and hypotheses with clear traceability.
- An AI engine using SARIMAX and Prophet generates reliable forecasts from historical data and analyst-defined factors.
- A modern web application was built with React/PrimeReact, Quarkus/Java, MySQL/Docker, and Python/FastAPI.
- Role-based access control serves the different actors in the budget process.
- Interactive Chart.js dashboards provide analytical insights into revenue data.
- A consolidation workflow aggregates projections across departments into a unified view.

The system demonstrates practical utility through its visualization tools, reporting features, and AI-driven projections, allowing stakeholders to monitor revenue trends, detect irregularities, and optimize budget planning strategies. Future work includes more advanced predictive models, expanded datasets, automated report generation, and real-time collaboration features.

REFERENCES

[1] Meta Platforms, Inc. (2024). React - A JavaScript library for building user interfaces. https://reactjs.org.
[2] PrimeTek. (2024). PrimeReact - Rich UI Component Suite for React. https://primereact.org.
[3] Red Hat. (2024). Quarkus - Supersonic Subatomic Java. https://quarkus.io.
[4] Oracle Corporation. (2024). MySQL 8.0 Reference Manual. https://dev.mysql.com/doc
[5] Docker Inc. (2024). Docker Documentation. https://docs.docker.com.
[6] Python Software Foundation. (2024). Python Documentation. https://www.python.org.
[7] Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with Python. https://www.statsmodels.org.
[8] Taylor, S.J., & Letham, B. (2018). Forecasting at Scale. The American Statistician. https://facebook.github.io/prophet.
[9] Tiangolo, S. (2024). FastAPI. https://fastapi.tiangolo.com.
[10] Chart.js Contributors. (2024). Chart.js. https://www.chartjs.org.
[11] Tardieu, H., Rochfeld, A., & Colletti, R. (1983). La methode MERISE: principes et outils.
[12] Object Management Group. (2017). Unified Modeling Language (UML), Version 2.5.
[13] Adamowski, J. F. et al. (2012). Comparison of multiple linear and nonlinear regression, autoregressive integrated moving average, and artificial neural network models. Water Resources Management, 26(2), 433-454.
[14] Herrera, M. et al. (2010). Predicting short-term urban water demand using artificial neural networks. Journal of Hydroinformatics, 12(3), 305-318.
[15] Nguyen, L. & Lee, J. (2021). Manual Budgeting and Forecasting and Its Disadvantages. International Journal on Economics, Finance and Sustainable Development, 3(2), 47–51.
[16] OECD (2019). OECD Good Practices for Performance Budgeting. OECD Publishing, Paris. https://doi.org/10.1787/c90b0305-en.
[17] Ministry of Finance, Ghana (2019). Budget Operations Manual. Government of Ghana.
[18] AF-TERG (2024). Guidance Note: Evaluation Budgeting. Adaptation Fund Technical Evaluation Reference Group (AF-TERG), Washington, DC.

[19] Prakash, K.O., Abdullah, K.A., Daoud, R., Moulana, Y., & Matriano, M.T. (2024). The Budgeting and Forecasting Analysis and the Impact on Company Financial Performance and Strategic Decision-Making: A Case of National Finance, Oman. Global Scientific Journal, 12(6), 2132–2157.
[20] Sahu, N. & Metta, A. (2025). A Study on Budgeting and Financial Forecasting Practices in CA Client Firms. International Journal of Research Publication and Reviews, 6(11), 10041– 10044.
[21] Olamijuwon, O.J. & Zouo, S.J.C. (2024). Budget Forecasting for Public Health Initiatives: A Conceptual Approach Using Machine Learning and Data Analytics. International Journal of Engineering Research and Development, 20(11), 1085–1093.
[22] Kruger, J. (2017). Planning and Budgeting. World Bank Learning Week, March 2017. World Bank Group.

About the authors

Dr. BAYOMOCK LINWA Andre Claude is the chief of computer sciences department at International University of Grand-Bassam (IUGB), Ivory Coast. He has a Ph.D. in Computer Engineering from "Ecole Polytechnique de Montreal" in Canada. In his past experience, he taught at "Ecole Polytechnique de Montreal" and at ETS (Ecole de Technologie Superieure) of Montreal. Dr. Bayomock gets a large professional experience and know-how as software consultant/expert in various industries (e.g. Banking, Government, Insurance, Telecommunications, Manufacturing, or Healthcare). His areas of research interests are mobile applications and networks, business intelligence, artificial intelligence, web services, data models, database and security.

Mr. TRAORE Cheick Oumar holds a Bachelor of Science in Computer Science from the International University of Grand-Bassam (IUGB), Ivory Coast. He completed his capstone internship at the Directorate of Budget Information Systems (DSIB), where he developed the AI-driven budget revenue projection system presented in this paper. His areas of interest include web
application development, artificial intelligence, and data-driven solutions for public administration.