Group 3 – Financial Forecasting and explainable risk assessment

1. Proposed Idea

We propose a **hybrid intelligent framework** for **financial forecasting, risk prediction, and decision support**, integrating both **structured numerical data** and **unstructured textual information** to enhance predictive accuracy and interpretability.

The framework consists of three main modules:

1. **ESN-based Financial Forecasting Module:**
   Utilizes **Echo State Networks (ESNs)** to learn complex temporal dependencies and market interactions, enabling robust and adaptive stock value forecasting.
2. **NLP-driven Risk Analysis Module:**
   Processes **textual data sources** such as financial news and reports to extract sentiment, key events, and other linguistic cues. This information is then transformed into a **quantitative risk index**, which complements the structured market data input to the ESN model.
3. **Explainability and RAG-based Decision Support Module:**
   Integrates outputs from both the ESN and NLP components using a **Retrieval-Augmented Generation (RAG)** approach. This layer provides **interpretable insights and justifications** by linking model predictions with relevant news and data sources—enabling transparent, explainable decisions and actionable insights for end users.

2. Scope & Purpose

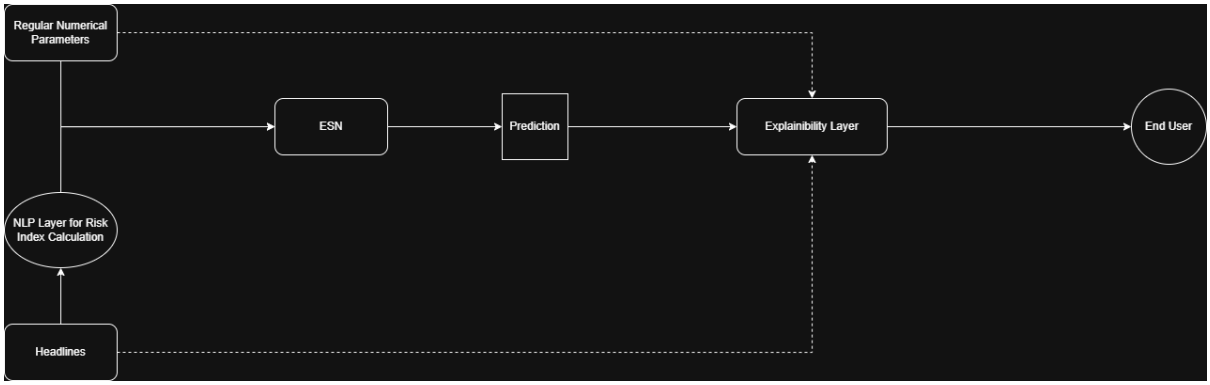
Traditional financial forecasting models rely heavily on quantitative indicators such as market trends, stock prices, and economic ratios, often overlooking the influence of real-time news and sentiment that significantly impact market behavior. To address this gap, the system introduces a **Natural Language Processing (NLP) layer** that extracts sentiment and risk cues from financial headlines, converting them into a quantifiable **risk index**.

This risk index, along with conventional numerical parameters, serves as input to an **Echo State Network (ESN)** — a recurrent neural model capable of efficiently capturing temporal dependencies and complex nonlinear patterns in financial time series. The ESN produces predictive outputs such as market risk levels or asset performance forecasts.

To ensure interpretability and trustworthiness, an **Explainability Layer** is incorporated, offering transparency into how individual features and news factors influence predictions.

Overall, the project aims to build an explainable, data-driven forecasting system that merges market statistics with contextual sentiment from news to support **more informed, transparent, and timely financial decision-making**.

This diagram shows a **hybrid financial forecasting or risk assessment architecture** that integrates numerical data, textual (news) data, and explainability for end users.



2.1 The NLP module

The proposed NLP module is designed to quantify the qualitative impact of financial news on market risk through an automated sentiment and embedding-based analysis pipeline. It focuses on extracting, processing, and encoding recent news headlines into structured numerical features suitable for integration with traditional financial models.

1. Data Acquisition

Recent financial news headlines were programmatically retrieved using the *Yahoo Finance* API (yfinance). For each target asset (e.g., S&P 500 index ticker “SPY”), the module collected metadata including the headline text, publication timestamp, and publisher. A temporal filter retained only news items published within the last *N* days (default: N = horizon selected for ESN output) to ensure that sentiment information remained relevant to current market dynamics.

2. Text Preprocessing and Cleaning

Headline data underwent preprocessing to remove missing or noisy entries and standardize textual format. Basic normalization operations—such as case-folding, removal of non-alphanumeric characters, and whitespace trimming—were applied to improve downstream model performance. Timezone alignment ensured temporal consistency between textual data and corresponding market indicators.

3. Sentiment and Semantic Representation

To capture both affective and contextual aspects of the news data, two complementary NLP techniques were employed:

- **Lexicon-Based Sentiment Analysis:** The *VADER (Valence Aware Dictionary and sEntiment Reasoner)* analyzer provided sentiment polarity scores (compound, positive, neutral, negative) for each headline. This method was chosen for its effectiveness in handling short, domain-relevant text such as financial news titles.
- **Transformer-Based Semantic Embeddings:** Sentence embeddings were generated using the pre-trained **all-MiniLM-L6-v2** model from the *Sentence-Transformers* library. These

embeddings encode high-dimensional contextual representations of headline meaning, capturing subtle linguistic and semantic variations beyond surface sentiment.

4. Feature Scaling and Dimensionality Reduction

The combined feature space—comprising sentiment scores and sentence embeddings—was standardized using **StandardScaler** to normalize feature magnitudes. Principal Component Analysis (**PCA**) was then applied to reduce dimensionality while preserving key variance components, thereby enhancing computational efficiency and mitigating redundancy among correlated textual features.

5. Risk Index Construction

A composite **risk index** was derived by aggregating the transformed features. This index reflects the overall sentiment-driven market tone and serves as an additional explanatory variable for the forecasting model (e.g., the Echo State Network in the full architecture). The output can be interpreted as a numerical proxy for news-driven risk intensity over a defined lookback period.

6. Experimental Outlook

This initial experiment establishes the feasibility of integrating sentiment and semantic representations for risk quantification. Future work will focus on:

- **Hyperparameter tuning**, including embedding dimensionality, PCA components, and lookback window size.
- **Model exploration**, such as fine-tuning domain-specific transformer models (e.g., FinBERT) or experimenting with attention-based fusion layers.
- **Temporal aggregation strategies**, to weight headlines based on recency or publisher credibility.

By embedding this NLP module into the broader forecasting pipeline, the system aims to capture real-world market psychology alongside quantitative trends, thereby enhancing both predictive accuracy and interpretability.

2.2 RAG Module Description

The proposed **Retrieval-Augmented Generation (RAG)** module serves as an intelligent retrieval and summarization layer that connects analytical model artifacts with natural-language interpretability. Its central objective is to enable **on-demand, explainable forecasting summaries** by dynamically retrieving relevant evidence—such as model outputs, evaluation metrics, and visualizations—and synthesizing them into coherent explanations using a large language model (LLM).

This module bridges the gap between complex quantitative modeling (ESN forecasting) and qualitative interpretability (NLP sentiment and narrative summaries).

1. Architectural Overview

The RAG system comprises two principal components:

1. **Retrieval Layer** – responsible for structured storage and efficient querying of model artifacts.

2. **Generation Layer** – powered by an LLM that composes context-aware summaries and conclusions based on retrieved content.

Together, these layers form a self-contained explainability framework capable of responding to user queries such as *"Explain the market risk prediction for week 42"* or *"Summarize performance metrics for the latest ESN model."*

2. Artifact Storage and Indexing

A dedicated **artifact store** is maintained to capture all key outputs generated during model development and experimentation.
This includes:

- **Model artifacts:** trained weights, hyperparameter configurations, and version metadata.
- **Evaluation outputs:** numerical metrics (e.g., MSE, RMSE, F1), confusion matrices, and time-series forecasts.
- **Visual artifacts:** charts showing predicted vs. actual performance, sentiment distributions, and feature importances.
- **Experimental metadata:** timestamp, training data period, feature set version, and associated NLP risk indices.

All artifacts are stored in a **vector-search-compatible database** (e.g., Pinecone, FAISS, or Chroma), where metadata and embeddings derived from textual annotations or file descriptions are indexed. This enables semantic retrieval—allowing the system to locate relevant experiments not just by keyword but by conceptual similarity (e.g., "latest ESN run using 7-day sentiment window").

3. Retrieval Mechanism

When a user requests a report or analysis for a specific **forecasting period** or **model version**, the retrieval layer executes the following steps:

1. **Query Parsing:** The natural-language query (e.g., "Show summary for Q3 forecast") is converted into an embedding using the same text encoder employed for artifact metadata.
2. **Similarity Search:** The system retrieves the top-*k* most relevant artifacts—charts, metrics, or configurations—from the vector store.
3. **Context Compilation:** The retrieved data are converted into a structured context package containing:
    - Key statistics and evaluation summaries
    - Associated visual evidence (if available)
    - Provenance information (date, model ID, dataset window)

This context forms the **retrieval grounding** for the generative stage.

4. Generation and Summarization

The **Generation Layer** employs a large language model (LLM) to synthesize a **concise, justified narrative** based on the retrieved context.
The model prompt follows a structured format:

> ***[System Instruction]***
>
> ***Summarize the forecasting performance for the specified period.***
>
> ***Include key numerical trends, interpret charts, and explain deviations.***
>
> ***Provide a short conclusion grounded in the retrieved evidence.***
>
> ***[Retrieved Context]***
>
> ***{metrics, charts, model configuration, textual notes}***

The LLM then produces an **interpretable summary** highlighting:

- Major performance observations (e.g., "The ESN outperformed baseline models by 12% RMSE improvement").
- Correlations between sentiment (from the NLP risk index) and prediction accuracy.
- Any notable trends or anomalies during the selected forecasting window.

This narrative output acts as the **explainable layer** of the architecture, allowing analysts or decision-makers to understand model behavior without examining raw metrics.

5. Planned Development Stages

The RAG system will evolve in the following stages:

Stage	Objective	Description
Stage 1	**News-Only Prototype**	Train and test the RAG system using stored news sentiment outputs (from the NLP module). The goal is to validate retrieval and LLM summarization based solely on textual and sentiment artifacts.
Stage 2	**Integration with NLP + ESN**	Extend retrieval to include ESN forecasting artifacts—metrics, predictions, and explainability charts. The RAG will combine both sentiment trends and temporal forecasting results in its summaries.
Stage 3	**Explainability & Decision Support Layer**	Deploy the RAG pipeline as a user-interactive module. Users can query any time period (e.g., "Explain market behavior during week 45") and receive an automatically generated summary that integrates sentiment signals, ESN forecasts, and SHAP-based explanations.