AI-Powered Business Intelligence Copilot Using Multi-Model Deep Learning and Large Language Model Analytics

D. Karishma and Dr. N. Kalaichelvi,Mca,M.phil.,ph.D

Advanced Computing & Analytics, Vels Institute of Science and Technology, Chennai, India

Abstract**—** The Business Intelligence Copilot has been created by utilizing multiple technologies such as multi-modal deep learning and Groq's large language model (LLM) implementation. The new copilot will allow users to take action on their business data through a Long Short-Term Memory (LSTM) network analyzing historic sales records and providing the revenue forecasts. Groq's LLM provides natural language querying for data; FastAPI supplies back-end application services; and Streamlit provides an interactive front-end dashboard integrating all trends, predictions, and conversational insights into one location. Comparative analysis shows that actual revenues reported during the same periods as generated forecasted revenues have been consistently accurate and have provided significant delays in deriving either of those insights from historical sales records. Combining deep learning predictive models with conversational AI represents a meaningful technological leap for all BI applications available today

.***Keywords*****—**Business Intelligence; LSTM; Large Language Model; Groq; Sales Forecasting; Natural Language Processing; FastAPI; Streamlit

I. INTRODUCTION

It is common for companies today to gather vast amounts of data from multiple sources (sales systems, customer relationship management systems, and website platform) at an ever-accelerating pace. The ability to quickly make sense of this data has become imperative in staying competitive in today's world. Business Intelligence (BI) systems have become the primary means to convert raw, numerical data into usable insights for decision-making. The primary focus of traditional BI systems has been to analyze only historical data. Although traditional BI systems do a good job of displaying past performance and trends, they typically do not provide abilities to predict future business outcomes, and they also require users to possess a certain level of technical expertise (SQL or proprietary query language) so that non-technical users cannot gain direct access to the company's data.Artificial intelligence (AI) and deep learning are making a significant impact on the work being done by traditional BI systems. The introduction of long short-term memory (LSTM) networks for sequence modeling has established LSTM as the preferred architecture for time-series problems since they maintain long-term contextual information, thereby overcoming the vanishing gradient problem associated with earlier RNNs. In addition to the LSTM network, the development of language models for natural language processing (LLMs) provides another significant change in the way users engage with a company's data. Rather than writing a query (i.e., SQL), an LLM-based BI system enables users to ask questions using natural language (i.e. English) and receive useful, contextually-based responses. This study describes the development of a single, unified BI application that combines LSTM-based revenue forecasting and Groq LLM analytics, allowing users to both forecast their revenue and interrogate their data through conversation without needing a data scientist.

Research Objectives

• Develop revenue forecasting based on deep learning

II. LITERATURE SURVEY

The last ten years have seen great advancements in Business Analytics, specifically in how businesses can use statistical analyses (regression models, ARIMA, and exponential smoothing) to analyze and interpret data. Initially, the use of these methods provided good results when working with relatively simple and clean datasets. However, as more companies continue to acquire an increasing amount of data from a variety of data sources (including sources that produce data in non-linear formats), it has become evident that these statistical analyses do not work well with the types of information previously described. On the other hand, machine learning algorithms such as Random Forests or Support Vector Machines can recognize complicated/non- linear relationships in the data and therefore provide better approaches to evaluating and analyzing historical data. Nonetheless, machine learning algorithms have not had a high degree of success when attempting to evaluate longer- term relationships between individual data points (i.e., how the variance of past data (covariance) may impact future data points). In contrast, Deep Learning models provide a different approach to interacting with and interpreting data from machine learning; most notably, the use of Recurrent Neural Networks for performing time-series analysis on sequential datasets and Long Short-Term Memory datasets have provided solutions for the problem of how to evaluate data across time.

III. PROPOSED SYSTEM

The proposed system is a BI Copilot structured around six core components that collectively handle both predictive analytics and natural-language business queries within one unified application:

- Data Collection Module
- Data Preprocessing Module
- Deep Learning Forecasting Model (LSTM)
- API-Based Model Deployment (FastAPI)
- Groq LLM Analytics Engine
- Interactive Visualization Dashboard (Streamlit)

Raw incoming data is processed through a cleaning and transformation pipeline before any model receives it. Monthly sales figures are extracted from the cleaned records to form the training series for the LSTM. Once trained, the model is wrapped in a FastAPI service, exposing a standard

HTTP interface for on-demand predictions. The Groq LLM handles conversational queries independently, enriching each question with relevant dataset statistics before generating a plain-language response. Streamlit ties all components together on the front end.

IV. SYSTEM ARCHITECTURE

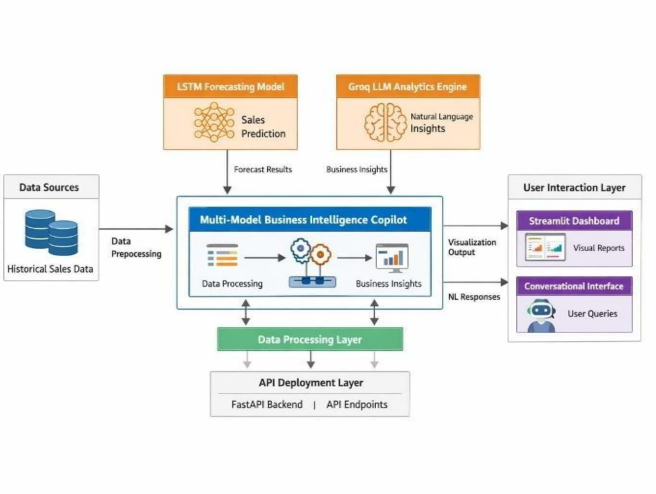

Fig. 1. Overall System Architecture of the BI Copilot

A. Data Source Layer

This layer stores and provides access to raw historical sales data, including order dates, product details, customer identifiers, geographic data, and transaction amounts. All downstream components depend on the integrity of data at this layer.

B. Data Preprocessing Layer

Real-world sales data rarely arrives in a usable state. This layer handles missing-value imputation, duplicate removal, datetime parsing, and Min-Max normalization to ensure that machine learning components receive consistent, well-formed input.

C. Feature Engineering Layer

Individual transaction records are aggregated into monthly totals, producing the time-series sequence on which the LSTM trains. We have chosen a monthly granularity as the most clear indication of overall business activity. For example, you can see seasonality and patterns of growth very plainly in monthly data.

D. Deep Learning Forecasting Layer

The LSTM neural network takes in the monthly time series, develops a deep understanding of the governing patterns, and makes revenue forecasts for future months. The architecture includes stacked LSTM layers configured with dropout regularization to reduce the potential for overfitting.

E. LLM Analytics Layer

The LLM generates a worded answer to an incoming natural-language user query by identifying an enriched list of relevant statistics from one or more data sets and using that information to generate a worded response to the question posed. This layer creates a new method for users to interact with the forecast model and LLM.

F. API Deployment Layer

FastAPI provides a RESTful API endpoint for the predictive model (/predict) and LLM Analytics engine (/insight). This provides a consistent interface to handle incoming requests and return the results of the request in a consistent format.

G. User Interface Layer

The Streamlit dashboard brings together a visual representation of the sales trend charts, LSTM forecast output, and conversational use of the dashboard by non-technical users, all in one view on a single screen and able to be used by a person with no specialized training.

V. PROPOSED METHODOLOGY & MATHEMATICAL MODEL

Four problems need to be solved in sequence by the system - namely, data cleaning, training the LSTM model, deploying the trained model, and integrating the LLM to the trained model. The problems are described further below with a mathematical formulation for each one.

A. Data Preprocessing

Denote that the full set of sales observations is given by

D = $\{x_1, x_2, ..., ..., x_n\}$ where x_i is the sales amount at time i. From these sales observations, a time series will be created from the monthly totals for further training of the model. Again, the sales observations will be normalized to [0, 1] using Min-Max normalization, so that the magnitude of values does not create bias for training the model.

$$X_s^{caled} = (X - X_i^{mn}) / (X^{max} - X_i^{mn})$$

B. LSTM Deep Learning Model

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network designed to forecast long sequences. They are designed this way because the early context of some data is essential to forecasting within the long sequence. Each time step in the model has three gates that will allow information to be processed.

The Forget Gate: which destroys parts of the long-term stored memory.

$$f_t = \sigma(Wf[h_{t-1}, x_t] + bf)$$

The Input Gate: which restricts how much new information is written to the cell state.

$$i_t = \sigma(W^I[h_{t-1}, x_t] + b^I)$$
$$\tilde{c}_t = tanh(W^c[h_{t-1}, x_t] + b^c)$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{c}_t$$

The Output Gate: which determines which information from the updated cell state is transferred to the hidden state.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \odot tanh(C_t)$$

C. Sales Forecasting

The LSTM forecasting engine performs a one step ahead forecast using a window of n historical sales values, as formalized by: $y_{t+1} = f(x_{t-n+1}, \ldots, x_t)$ Where: - y_{t+1} is the forecasted future sales value - f = the trained LSTM forecasting engine - x_t = Previous sales value(s) contained in the input window.

D. Groq LLM Analytics Engine

After the user submits their request through the Groq API, their query is enhanced with database statistics. The entire pipeline for each query consists of the following five steps:

1) Extract statistical information from the database

2) Construct the inputted LLM prompt

3) Send the request to the Groq API

4) Receipt of return plain-language business insight Enriched LLM Business Insight is Equal to

LLM(Query + Summary of Data).

E. API-Based Deployment

Each of the LSTM models is published via FastAPI app. There are two endpoints that expose the forecast and business insights functionality: /predict and /insight.

VI. SYSTEM INPUT

A. Dataset Input

The forecasting component relies on historical sales data. Each transaction record provides a training point; the more history, the greater the reliability of the forecast.

Field	Description
Order Date	Date when the transaction occurred
Customer Name/ID	Identifies the purchasing customer
Product Details	Information about the purchased product
Country/Region	Customer geographic location
Sales Amount	Total transaction value

TABLE I. Dataset Input Fields

B. User Query Input

The second input type can accept unstructured, natural language questions without requiring knowledge of SQL or any other query language. Examples of supported questions include: "Who is the customer that paid us the most?" "What have my total sales been during year N?" "Which country produced the most revenue?"

VII. PSEUDOCODE & IMPLEMENTATION

Here are the algorithms that convey the logic of the major parts of the BI Copilot system.

Algorithm 1: Data Pre-processing & Monthly Sales Grouping

```
Input (Sales Data): sales_data.csv
Output: Monthly Sales Time Series
1. Load CSV file containing sales data
2. Remove all whitespace from column names
3. Convert the ORDERDATE field to a datetime format
4. Remove all orders without an ORDERDATE
5. Create a new field called YearMonth, which is based on the ORDERDATE field
6. Group records by YearMonth; sum up SALES for each Month
7. Sort results chronologically
8. The output will be a time series of Monthly Sales
```

Algorithm 2: LSTM Model Training

```
Input:  Monthly Sales Time Series
Output: Trained LSTM Model, Evaluation Metrics
1.  Normalize values with MinMaxScaler; save scaler
2.  Set window_size = 6
3.  For i = window_size to len(data):
        X[i] ← previous 6 months; Y[i] ← current month
4.  Split 80/20 train/test
5.  Build LSTM(100) → Dropout(0.2) → LSTM(100) → Dropout(0.2) → Dense
6.  Compile: Adam optimizer, MSE loss
7.  Train with EarlyStopping; save model
8.  Evaluate: MAE, RMSE, MAPE, R²
```

Algorithm 3: Sales Forecasting Prediction (FastAPI)

```
Input:  Trained LSTM Model, Monthly Sales Data
Output: Next Month Sales Prediction
1. Load an already-trained LSTM model and scaler
 2. If the length of monthly_sales is less than the window size, return an error
3. Isolate the last 6 months of sales data
4. Reshape these values into an array that can be fed to the LSTM model as input
5. Run the prediction (thank you to the LSTM model); then apply inverse-scaling to the prediction
6. Return the resulting predicted sales value
```

Algorithm 4: Business Insight Generation (LLM)

```
Input: Business Question, Sales Dataset
Output: Business Insight Generated using AI.
1. Calculate the total sales, top customers, and top countries.
2. Create an enriched prompt.(Stats + User Question)
3. Send the enriched prompt to the Groq LLM API.
 4. Receive, and return, the AI generated business insight.
```

Algorithm 5: Interactive BI Dashboard

```
   Input:  Sales Dataset
   Output: Interactive Streamlit Dashboard
1.  Load and preprocess dataset
2.  Aggregate monthly sales

3. Set up Three Dashboard Components:

  a. Analytics | Forecast | Copilot

4. Analytics: Show Monthly Trends, Top 10
Countries, Top 10 Customers

5. Forecast: Train Long Short Term Memory
Model and Use to Predict Next Month and
Display Metrics

6. Copilot: Receive Query, Generate Insight
Using a language model and Show the Result
```

VIII. SYSTEM OUTPUT

In this system, there are two types of outputs generated by the system. For one output, revenue is forecasted based on the historical revenue patterns of the organization as identified by the LSTM model, called the forecasted revenue output. For the second output, there are headlines that contain answers to the user's requests and are based on actual data from the database (Q&As). A user could request to know who the highest revenue-generating customer was last quarter, which country generated the largest amount of revenue, and to see how much revenue will be generated next month. All of these answers will be provided in the same Streamlit dashboard, where charts, numeric value displays, and textual responses are listed together and aligned accordingly.

IX. RESULTS & DISCUSSION

Both parts of a system were tested to confirm their design is correct. The LSTM was successful at recognizing both seasonal and growth trends using historical data, which showed that the forecasting model produced valid predictions (forecasts) relative to previously unseen test periods. Because many business resource allocation decisions require the use of forecasted data, it is especially important that forecasting models exhibit repeatable patterns across multiple tests. The LLM component of the system allowed for capabilities not available to the forecasting model alone: namely, the ability to ask questions (i.e., "What", "Why", and "Who") and receive associate responses based on the user's own data rather than generalized responses. The reliability of the forecast plus the conversational response layer makes this system significantly more valuable than either a static report or a traditional dashboard.

X. SCREENSHOTS

These are the graphs showing the actual performance of the entire system using data from an actual dataset. These graphs illustrate how the different functionality elements of the overall BI interface interact and work together to support revenue forecasting and natural language data queries on a single platform.

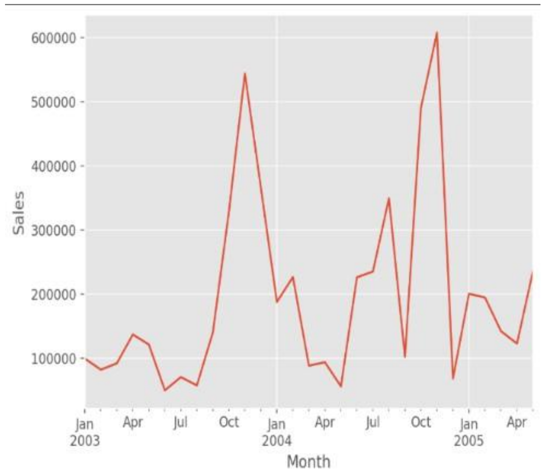

Fig1.. Streamlit Dashboard Interface

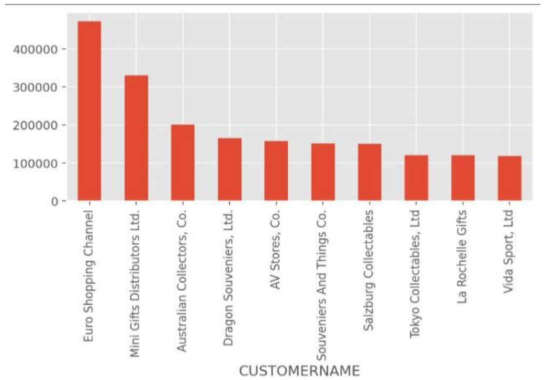

Fig.2. country sales and revenue

Fig. 4. Deep learning forecast

Fig. 4. FastAPI Sales Forecast Endpoint

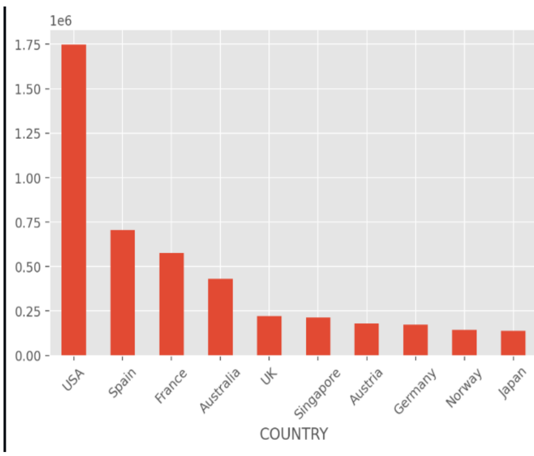

Fig. country sales prediction

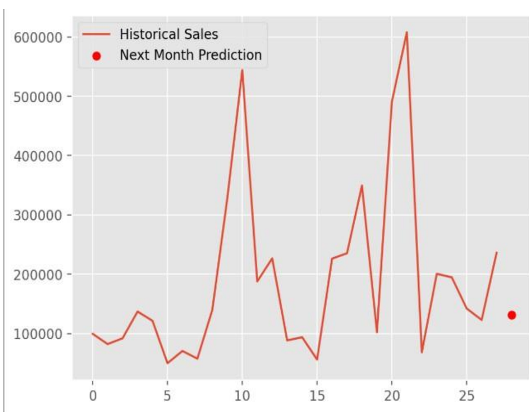

Fig. 7. Historical sales prediction

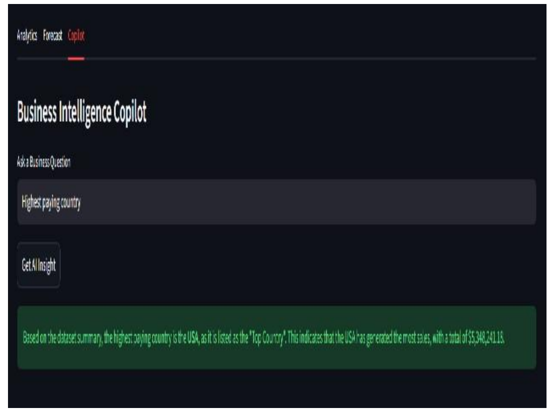

Fig. 6. Groq LLM Business Insight Response

XI. CONCLUSION

The goal of the Business Intelligence Copilot is to allow users of all technical abilities access to both predictive analytic tools (through revenue forecasting) and traditional forms of analysis (using natural language queries) on a single platform. The LSTM forecasting engine has successfully learned patterns in historical sales data, and has generated accurate predictions for each of the test timeframes that were evaluated. Non-technical users can use the Groq LLM to easily query data in order to receive precise and credible answers to their queries, instead of receiving only tables of raw data in response to their queries. The Streamlit dashboard allows users to use all of these functions from one location, with one user interface, which can be navigated easily without any technical expertise. In order to continue building upon the BI Copilot, the team is examining several ways to expand the capabilities of this tool, including connecting to live data feeds to generate updated revenue forecasts, looking at transformer based architectures to complement or replace LSTMs in forecasting techniques, and moving the BI Copilot to be hosted in the cloud as a way to have access to virtually unlimited amounts of data for future analysis.

XII. REFERENCES

[1] A. Vaswani et al., "Attention Is All You Need," in Proc. Neural Information Processing Systems (NIPS), 2017.

[2] T. B. Brown et al., "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems, 2020.

[3] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[4] Google, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," Google Research, 2016.

[5] OpenAI, "GPT-4 Technical Report," 2023.

[6] C. D. Manning and H. Schütze, Foundations of Statistical Natural Language Processing. MIT Press, 1999.