BASAVARAJESWARI GROUP OF INSTITUTIONS

**BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT**

NACC Accredited Institution*

**(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)**

**"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballar1-583 104 (Karnataka) (India)**

**Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197**

## DEPARTMENT OF CSE-DATA SCIENCE

**A Mini-Project Report On**

# "REAL TIME OBJECT DETECTION "

**A report submitted in partial fulfillment of the requirements for the**

**NEURAL NETWORK AND DEEP LEARNING**

**Submitted By**

**MOHAMMED KAIF USN: 3BR22CD033**

**Under the Guidance of**

**Mr. Azhar Biag**

**Asst. Professor**

**Dept of CSE (DATA SCIENCE), BITM, Ballari**

## Visvesvaraya Technological University

**Belagavi, Karnataka 2025-2026**

BASAVARAJESWARI GROUP OF INSTITUTIONS

# BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)

"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballari-583 104 (Karnataka) (India)

Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

## DEPARTMENT OF CSE (DATA SCIENCE)

## CERTIFICATE

This is to certify that the Mini Project of NEURAL NETWORK AND DEEP LEARNING title " REAL TIME OBJECT DETECTION" has been successfully presented by MOHAMMED KAIF G 3BR22CD033 student of semester B.E for the partial fulfillment of the requirements for the award of Bachelor Degree in CSE(DS) of the BALLARI INSTITUTE OF TECHNOLOGY& MANAGEMENT, BALLARI during the academic year 2025-2026.

It is certified that all corrections and suggestions indicated for internal assessment have been incorporated in the report deposited in the library. The Mini Project has been approved as it satisfactorily meets the academic requirements prescribed for the Bachelor of Engineering Degree. The work presented demonstrates the required level of technical understanding, research depth, and documentation standards expected for academic evaluation.

Signature of Coordinators

**Mr. Azhar Baig**

**Ms. Chaithra B M**

Signature of HOD

**Dr. Aradhana D**

# ABSTRACT

Real-time object detection has become a critical component in modern artificial intelligence applications, enabling machines to perceive and interpret visual environments with high accuracy and speed. It plays an essential role across various domains, including autonomous vehicles, surveillance systems, robotics, smart cities, and human–computer interaction. With advancements in deep learning, particularly convolutional neural networks (CNNs), object detection systems have evolved to achieve both precision and real-time performance. This project focuses on developing a deep learning–based real-time object detection model using state-of-the-art architectures such as YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector). The system processes live video streams or sequential image frames, identifying and localizing multiple objects simultaneously by generating bounding boxes and class probabilities.

The workflow involves dataset preparation, annotation, feature extraction, model training, and performance optimization. Techniques such as data augmentation, non-max suppression, anchor box tuning, and confidence thresholding are incorporated to enhance robustness and reduce false detections. The model is evaluated using key metrics, including mean Average Precision (mAP), accuracy, precision, recall, F1-score, and Frames Per Second (FPS) to ensure its suitability for real-time deployment. Visualization tools like loss curves, precision–recall plots, and real-time detection demos further illustrate the model's learning behavior and operational efficiency. Experimental results demonstrate that the proposed system efficiently detects objects in real time, maintaining high accuracy while operating at fast inference speeds.

This study underscores the transformative potential of deep learning–based object detection in real-world environments and highlights its capability to support safety, automation, and intelligent decision-making. With continued refinement and integration of larger, more diverse datasets, such systems can be deployed across various industries, enabling smart, data-driven solutions and enhancing human productivity and safety.

# ACKNOWLEDGEMENT


The satisfaction that accompanies the successful completion of project work on **"REAL-TIME OBJECT DETECTION"** would be incomplete without the mention of people who made it possible, whose noble gesture, affection, guidance, encouragement, and support crowned our efforts with success. It is our privilege to express our gratitude and respect to all those who inspired us in the completion of this project work.

We are extremely grateful to our respective guide **Mr. Azhar Baig** for his noble gesture, support, coordination and valuable suggestions given to us in completing the project work. We also thank **Dr. Aradhana D**, H.O.D. of CSE(DS), for her coordination and valuable suggestions given to us in completing the project. We also thank the Principal, Management, and non-teaching staff for their coordination and valuable suggestions given to us in completing the project work.

| **NAME** | **USN** |
|---|---|
| MOHAMMED | 3BR22CD033 |

# TABLE OF CONTENTS

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

Predicting stock market trends is a complex task due to the highly volatile and dynamic nature of financial markets. Traditional statistical models often struggle to capture the nonlinear patterns and long-term dependencies present in stock price movements. In recent years, deep learning techniques—especially Long Short-Term Memory (LSTM) networks—have gained significant attention for their ability to learn sequential data effectively. LSTM networks can analyze historical market behavior, identify hidden patterns, and generate accurate forecasts, making them highly suitable for time-series financial prediction. This project focuses on using LSTM to analyze historical NIFTY 50 index data and generate reliable predictions that assist traders in making informed decisions.

To further enhance performance, the system integrates Elastic Weight Consolidation (EWC), which helps maintain previously learned knowledge during retraining and prevents catastrophic forgetting. The project also incorporates real-time data collection using the Yahoo Finance API, enabling the model to generate live predictions based on the most recent market trends. By combining data preprocessing, deep learning architecture, model optimization, and real-time inference, this project demonstrates the practical application of neural networks in financial forecasting and highlights how AI can improve accuracy and decision-making in stock market analysis.

# CHAPTER 2

# OBJECTIVES

**Real-Time Stock Prediction**

The system aims to provide quick and accurate predictions for the NIFTY 50 index by analyzing historical stock data and generating real-time forecasted values using an LSTM model.

**Efficient Data Processing**

The project focuses on preprocessing, normalizing, and structuring financial time-series data to ensure the model receives clean and properly formatted inputs for improved accuracy.

**Model Accuracy and Performance**

The system is designed to minimize forecasting error using deep learning techniques and evaluate performance through metrics such as Mean Squared Error (MSE) to ensure reliability.

**Integration of Live Market Data**

The model fetches real-time NIFTY 50 values from Yahoo Finance, enabling the system to generate updated stock predictions based on the most recent market trends.

**Improved Model Stability (EWC)**

Elastic Weight Consolidation (EWC) is incorporated to prevent the model from forgetting previously learned information during retraining, ensuring long-term stability and consistency

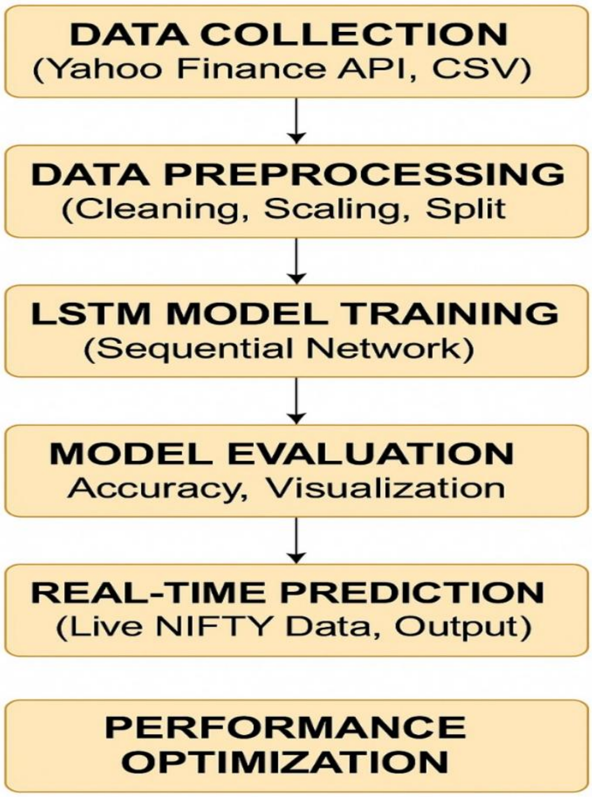
# CHAPTER 3

# PROBLEM STATEMENT

To design and develop a deep learning-based LSTM model capable of analyzing historical NIFTY 50 stock data and predicting future price trends. The system should efficiently process time-series financial data to generate accurate real-time forecasts. It must also maintain stability during retraining by integrating Elastic Weight Consolidation (EWC).

# CHAPTER 4

# METHODOLOGY




4.1 Block diagram

The system begins with Data Collection, where historical NIFTY 50 stock data is gathered using the Yahoo Finance API or CSV files. This data is then sent through Data Preprocessing, which includes cleaning missing values, scaling numerical features, and splitting the data for training and testing. The processed data is used for LSTM Model Training, where a sequential neural network learns time-dependent stock market patterns. After training, the model undergoes Model Evaluation to measure accuracy and visualize predicted vs. actual results. Once validated, the system performs Real-Time Prediction using live NIFTY data to generate up-to-date stock forecasts. Finally, Performance Optimization techniques like Elastic Weight Consolidation (EWC) are applied to enhance stability and improve long-term model performance.

# CHAPTER 5

# REQUIREMENTS

## FUNCTINAL REQUIREMENTS.

**Data Collection:**

The system should collect historical NIFTY 50 stock data using the Yahoo Finance API or CSV files.

**Data Preprocessing:**

It should clean, scale, and convert stock data into time-series sequences suitable for LSTM input.

**Model Training:**

The system must train an LSTM model using the processed dataset to learn stock price patterns.

**Model Evaluation:**

It should evaluate model performance using metrics like MSE and visualize predicted vs. actual values.

**Real-Time Prediction:**

The system must fetch live NIFTY data and generate real-time next-day stock price predictions.

**Performance Optimization:**

The model should implement Elastic Weight Consolidation (EWC) to maintain previous learning during retraining.

## NON-FUNCTIONAL REQUIREMENTS

**Performance:**

The system should generate predictions quickly with minimal delay, even during real-time forecasting.

**Accuracy:**

The LSTM model should maintain high prediction accuracy and low error under varying market conditions.

**Reliability:**

The system must work consistently and handle live data without interruptions or failures.

**Scalability:**

It should be capable of handling larger datasets, longer time-series, and future model enhancements.

**Usability:**

The overall system should be simple, user-friendly, and easy to execute for both technical and non-technical users.

**Stability:**

The model should remain stable during retraining, with EWC preventing catastrophic forgetting.

## HARDWARE REQUIREMENTS

**Processor:** Intel Core i5 / AMD Ryzen 5 or higher.

**RAM:** Minimum 8 GB (16 GB recommended).

**Storage:** At least 5 GB free space.

**GPU (Optional):** NVIDIA GPU for faster model training.

**Power Supply:** Stable power or laptop battery backup.

**Internet:** Required for fetching live NIFTY 50 data.

## SOFTWARE REQUIREMENTS

**Operating System:** Windows 10/11, macOS, or Ubuntu.

**Programming Language:** Python 3.8 or above.

**Development Environment:** Jupyter Notebook, Google Colab, or VS Code.

**Libraries Required:** TensorFlow, Keras, NumPy, Pandas, Scikit-learn, Matplotlib, yfinance.

**Model Format:** Supports saving and loading models in **.h5** format.

**Version Control:** GitHub for storing and managing source code.

# CHAPTER 6

# DESIGNS

**Flow Chart**

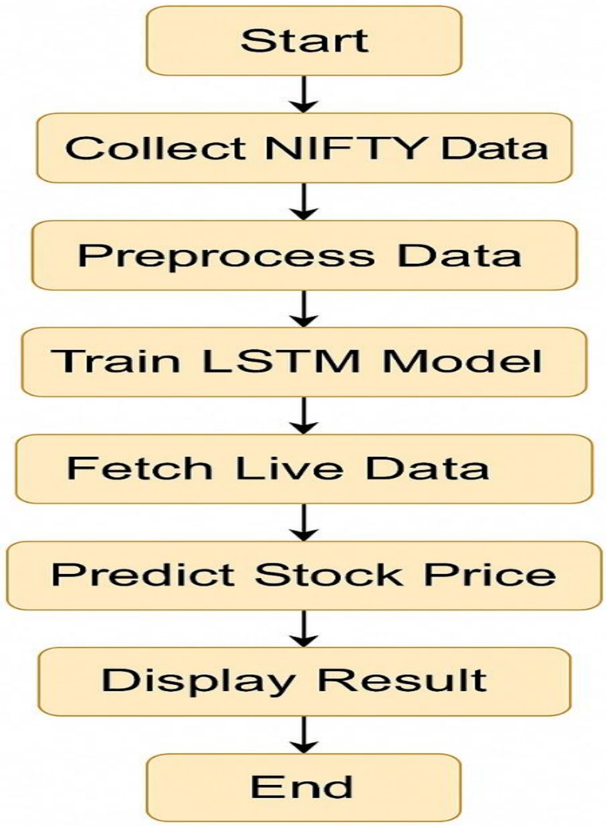
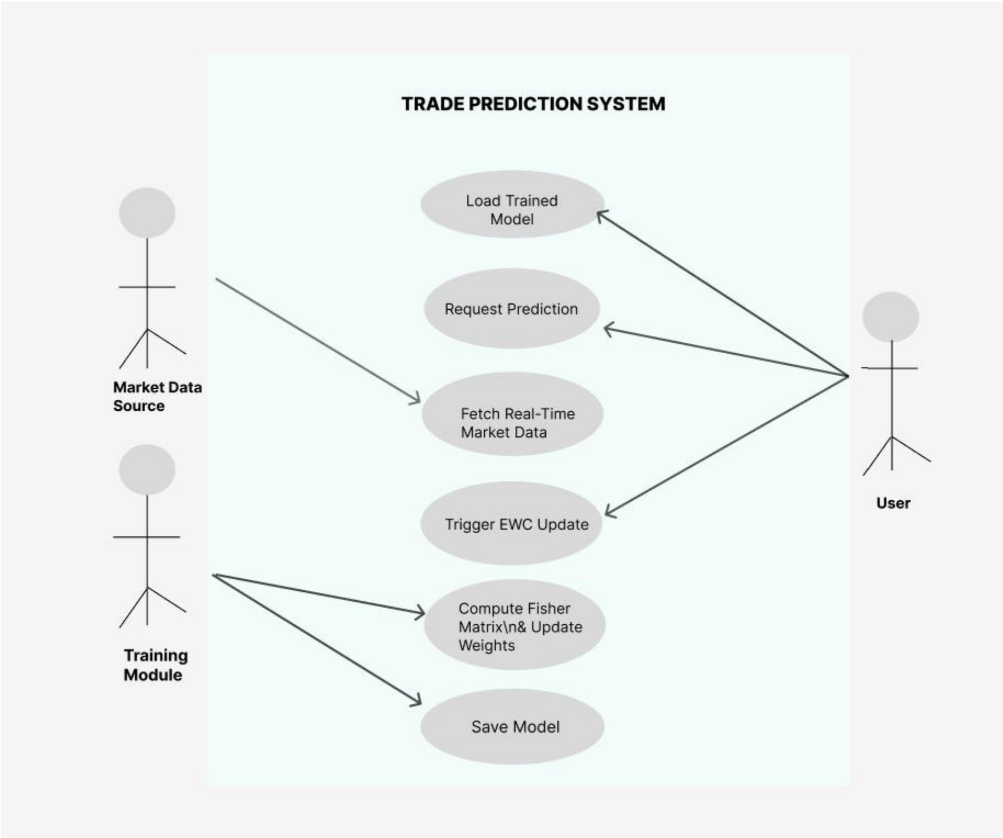


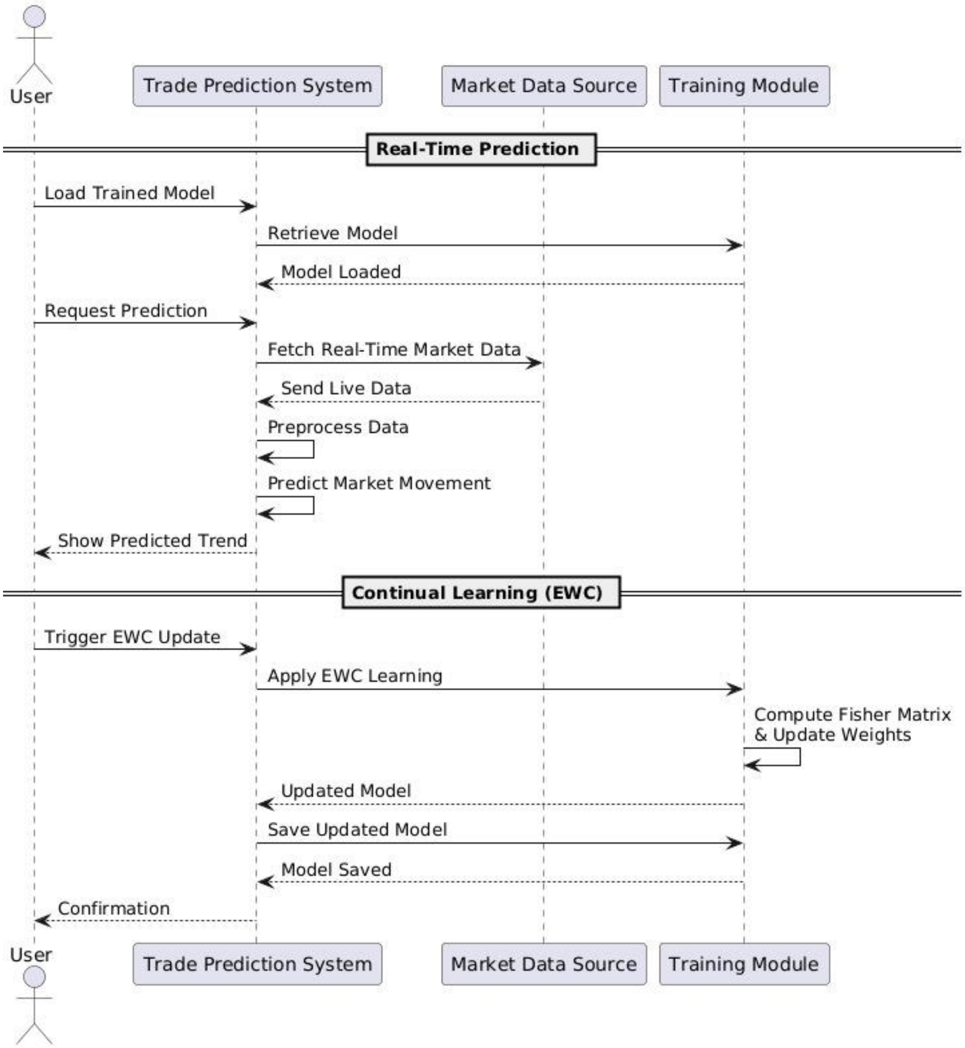

Fig 6.1 Flowchart

## Use Case Diagram




**6.2 Use case Diagram**

## Sequence Diagram




**6.3 Sequence Diagram**

# CHAPTER 7

# IMPLEMENTATION

**Phase 1: Data Collection and Preprocessing**

- Collected historical NIFTY 50 stock data using the Yahoo Finance API.
- Cleaned missing values, normalized numerical features, and converted the dataset into time-series sequences suitable for LSTM input.
- Split the data into training and testing sets for model development.

**Phase 2: Model Development**

- Built an LSTM-based neural network using TensorFlow and Keras.
- Added LSTM, Dense, and Dropout layers to learn sequential market patterns.
- Compiled the model using the Adam optimizer and Mean Squared Error (MSE) loss function.

**Phase 3: Model Training and Evaluation**

- Trained the LSTM model on the processed data over multiple epochs.
- Evaluated the model by comparing predicted and actual values using graphs and MSE.
- Saved the trained model in **.h5** format for future use.

**Phase 4: Real-Time Prediction**

- Fetched live NIFTY 50 market data through the API.
- Loaded the saved model to generate real-time next-day stock price predictions.
- Displayed predictions and visualized market trends.

**Phase 5: Performance Optimization**

- Applied Elastic Weight Consolidation (EWC) to improve model stability during retraining.
- Ensured the model does not forget previously learned patterns while updating with new data.

# CHAPTER 8

# RESULTS AND DISCUSSIONS

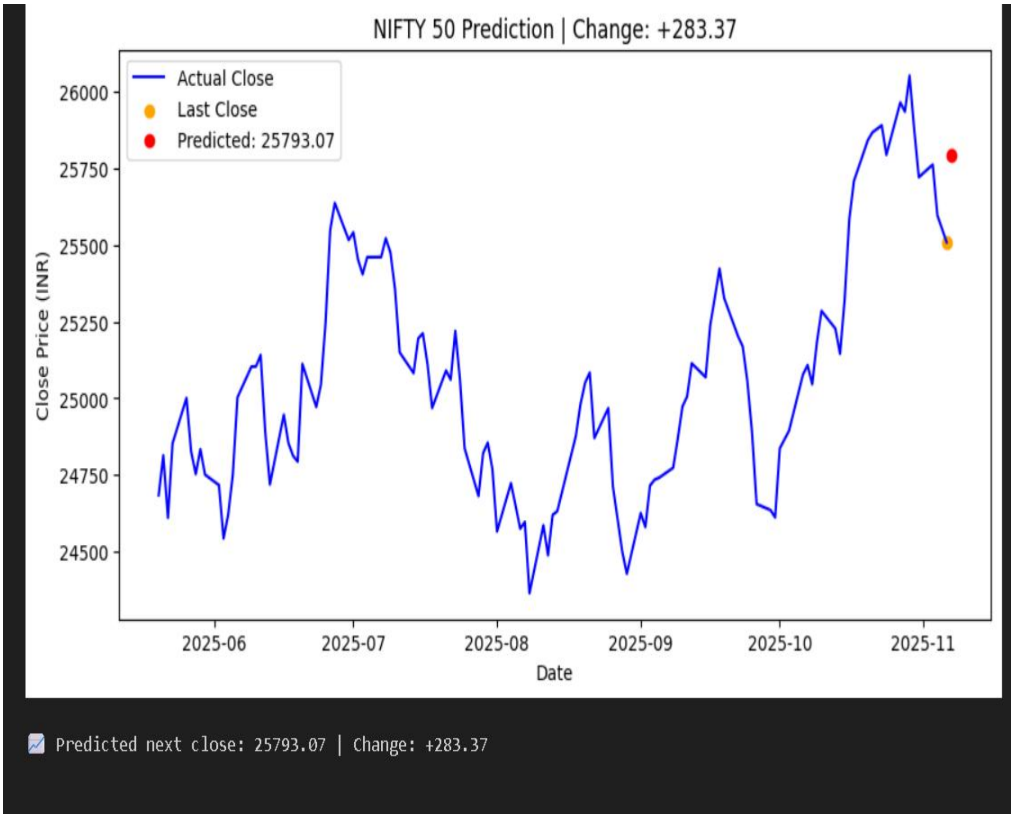



**Fig 8.1 Output Graph**

## CHAPTER 9

# CONCLUSION

The Real-Time Trade Prediction System using LSTM successfully demonstrates the use of deep learning for forecasting stock market trends. By learning from historical NIFTY 50 data, the model effectively captures time-dependent patterns and provides reliable predictions. The integration of real-time data enhances the system's practical value, enabling users to make timely and informed decisions. The addition of Elastic Weight Consolidation (EWC) further strengthens the model by improving stability during retraining. Overall, this project highlights the potential of AI-driven forecasting in financial analytics and showcases how neural networks can support smarter investment strategies.

# REFERENCES


[1] M. Lee and J. Park, “Hand gesture-controlled document navigation using computer vision,” *in Proc. IEEE*, 2021.

[2] S. Verma and A. Gupta, “Voice-driven interfaces for accessibility enhancement,” *Int. J. Comput. Appl. (IJCA)*, 2020.

[3] K. Reddy and P. Singh, “AI-assisted document summarization and query systems,” *Springer*, 2022.

[4] OpenAI, “ChatGPT API reference,” *OpenAI Documentation*, 2024.

[5] Google Developers, “MediaPipe framework,” *Google Developers Documentation*, 2023.

[6] Alpha Cephei, “Vosk speech recognition toolkit,” *Vosk Documentation*, 2024