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-ARTIFICIAL INTELLIGENCE

A Mini-Project Report On

"SENTIMENT ANALYSIS ON MOVIE REVIEWS"

A report submitted in partial fulfillment of the requirements for the

NEURAL NETWORK AND DEEP LEARNING

<u>Submitted By</u>

P N SHREELEKHA USN: 3BR22CA040

<u>Under the Guidance of</u>
Prof. Pavan Kumar

Mr. Vijay Kumar

Dept of CSE (ARTIFICIAL INTELLIGENCE),
BITM, Ballari

Visvesvaraya Technological University

Belagavi, Karnataka 2025-2026

BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

Autonomous Institute under VTU, Belagavi | Approved by AICTE,New Delhi Recognized by Govt. of Karnataka

NACC Accredited Institution*
(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belgavi)
"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballar1-583 104 (Karnataka) (India)
Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE

CERTIFICATE

This is to certify that the project work entitled "Sentiment Analysis on Movie Reviews" is a bonafide work carried out by P N Shreelekha in partial fulfillment for the award of degree of **Bachelor Degree in CSE (Artificial Intelligence)** in the BALLARI INSTITUTE OF TECHNOLOGY AND 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 project has been approved as it satisfies the academic requirements in respect of mini project work prescribed for a Bachelor of Engineering Degree.

Signature of project guide
Prof. Pavan Kumar
Mr. Vijay Kumar

11/12/25
Signature of HoD
Dr. Yeresime Suresh

11/12/2025 14:18

ABSTRACT

Sentiment analysis has become a crucial area of research in natural language processing, enabling machines to understand human opinions, emotions, and attitudes expressed in text. With the rapid growth of online platforms such as movie review websites, social media, and digital forums, the ability to automatically classify user-generated content has significant applications in recommendation systems, audience analysis, and decision-making. This project focuses on developing an Artificial Neural Network (ANN)–based sentiment classification system that identifies movie reviews as positive, negative, or neutral. Using a labeled movie review dataset, the system preprocesses textual inputs through cleaning, tokenization, and TF-IDF vectorization to convert natural language into meaningful numerical representations suitable for machine learning models.

The neural network architecture is designed with multiple dense layers, ReLU activations, dropout regularization, and a softmax output layer for three-class sentiment prediction. These components enable the network to learn complex linguistic patterns, distinguish emotional tones, and handle noisy real-world text. Once the model is constructed, it is trained using optimized parameters and evaluated using metrics such as accuracy, precision, recall, F1-score, and a detailed classification report. Additionally, training curves for accuracy and loss are visualized to analyze model performance, detect overfitting, and assess generalization ability.

Experimental results demonstrate that the ANN model effectively captures key sentiment features within movie reviews and provides reliable classification performance across all three sentiment categories. This study highlights the potential of neural networks in understanding human expression and showcases their capability to support automated text analysis systems. With further enhancements, including larger datasets and advanced deep learning models such as LSTMs or transformers, the system can be extended to real-world applications like review monitoring, content filtering, and intelligent recommendation engines.

ACKNOWLEDGEMENT

The satisfactions that accompany the successful completion of our mini project on SENTIMENT ANALYSIS ON MOVIE REVIEWS would be incomplete without the mention of people who made it possible, whose noble gesture, affection, guidance, encouragement and support crowned my efforts with success. It is our privilege to express our gratitude and respect to all those who inspired us in the completion of our mini-project.

I am extremely grateful to my Guide **Prof. Pavan Kumar and Mr. Vijay Kumar** for their noble gesture, support co- ordination and valuable suggestions given in completing the mini-project. I also thank **Dr. Yeresime Suresh,** H.O.D. Department of CSE(AI), for his co-ordination and valuable suggestions given in completing the mini-project. We also thank Principal, Management and non-teaching staff for their co-ordination and valuable suggestions given to us in completing the Mini project.

Name	USN
P N SHREELEKHA	3BR22CA040

TABLE OF CONTENTS

1. INTRODUCTION

Sentiment analysis, also known as opinion mining, has become an essential area of research within natural language processing (NLP) as digital communication continues to grow rapidly across social media, review platforms, and online forums. Movie review sentiment analysis focuses on identifying whether a user's review expresses a positive, negative, or neutral opinion about a film. Understanding audience sentiment helps industries such as filmmaking, marketing, and entertainment platforms gain valuable insights into viewer preferences, reactions, and satisfaction levels. With millions of reviews posted online every day, manually analyzing sentiment is time-consuming, subjective, and inefficient, highlighting the need for automated and intelligent systems.

Traditional methods of sentiment classification often rely on manual reading, rule-based approaches, or lexicon-based techniques, which may fail to capture complex linguistic patterns such as sarcasm, implicit emotions, context-dependent expressions, and mixed sentiments. As the volume of textual data increases, traditional techniques become inadequate for achieving high accuracy and scalability. To overcome these limitations, machine learning and artificial intelligence techniques have become powerful tools for sentiment prediction.

Among various approaches, Artificial Neural Networks (ANNs) have shown strong performance due to their ability to learn non-linear relationships, extract hidden patterns, and generalize well across diverse text inputs. ANNs mimic the functioning of the human brain, enabling them to process high-dimensional data such as TF-IDF vectors generated from review text. When combined with appropriate preprocessing techniques—such as cleaning, tokenization, and vectorization—ANNs can efficiently identify emotional tone and classify sentiment with improved accuracy.

The main objective of this work is to design, implement, and evaluate a reliable ANN-based sentiment classifier capable of understanding emotional patterns in movie reviews. The methodology demonstrates how deep learning techniques can support automated text analysis and highlights their potential applications in recommendation systems, review filtering, entertainment analytics, and digital feedback monitoring.

1.1 Problem Statement

This project addresses the need for an automated and accurate system to classify movie reviews into positive, negative, or neutral sentiments, as manual analysis is slow, subjective, and inefficient. Traditional methods struggle to capture complex emotional patterns and contextual meanings in natural language. Therefore, an ANN-based model using TF-IDF features is developed to accurately predict sentiment and support platforms and users in understanding public opinion more effectively.

1.2 Scope of the project

The scope of this project includes designing, implementing, and analyzing an Artificial Neural Network (ANN) model for sentiment classification on movie review text. It covers essential stages such as data preprocessing, text cleaning, TF-IDF vectorization, model construction using multiple dense layers, training and validation, and evaluation using metrics like accuracy, precision, recall, F1-score, and classification reports. The project also includes graphical visualization of training and validation accuracy and loss, helping interpret model behavior and generalization capability. Although this work is limited to a specific movie review dataset, the methodology can be extended to larger datasets, multi-language sentiment analysis, and real-world applications such as automated review moderation, recommendation systems, audience analytics, and social media monitoring.

1.3 Objectives

❖ To build an ANN-based model capable of accurately classifying movie reviews into positive, negative, or neutral sentiment categories.

❖ To preprocess and vectorize the text dataset using TF-IDF for enhanced numerical representation.

❖ To train and evaluate the ANN model using performance metrics such as accuracy, precision, recall, and F1-score.

❖ To visualize learning behaviour using accuracy and loss graphs for analyzing model performance and stability.

2. LITERATURE SURVEY

[1] *Gupta et al. (2023)* explored sentiment analysis using TF-IDF with machine learning algorithms such as SVM, Logistic Regression, and Random Forest. Their study highlighted that proper text preprocessing and feature extraction significantly improve classification accuracy on movie review datasets.

[2] *Hassan & Qamar (2023)* investigated deep learning techniques for sentiment analysis and demonstrated that ANN and LSTM architectures outperform traditional models when handling complex sentence structures and contextual expressions, particularly in movie review data.

[3] *Singh et al. (2022)* conducted a comparative study on multiple sentiment classification methods and emphasized the importance of vectorization techniques. Their results showed that TF-IDF combined with neural networks produces stable and interpretable performance suitable for real-time review analysis.

[4] *Rahman et al. (2022)* evaluated several machine learning approaches and found that handling class imbalance and using optimized preprocessing strategies significantly increase sentiment classification accuracy. Their work supports the need for balanced datasets in text analytics projects.

[5] *Adebayo & Chen (2023)* investigated hybrid deep learning frameworks for text sentiment prediction. They found that neural networks with multiple hidden layers can effectively capture non-linear emotional patterns and generalize better across noisy text data.

[6] *Li & Zhou (2024)* implemented a CNN–BiLSTM hybrid model for multi-class sentiment analysis and demonstrated high accuracy by combining spatial and sequential feature learning. Although computationally heavier, their study confirms the effectiveness of deep neural architectures.

[7] *S. Kumar & R. Sharma (2022)* compared Naïve Bayes, SVM, Decision Trees, and ANN using movie review datasets and concluded that ANN models provide better performance when combined with rich feature extraction methods like TF-IDF or word embeddings.

3. SYSTEM REQUIREMENTS

The system requirements for developing the sentiment analysis model include both software and hardware components necessary for processing text data, training neural networks, and evaluating model performance. The software environment is built using Python along with essential libraries such as TensorFlow/Keras for the ANN model, Pandas and NumPy for dataset handling, Scikit-learn for preprocessing and TF-IDF vectorization, and Matplotlib for visualization. Development platforms like Jupyter Notebook, Google Colab, or VS Code are used for coding and execution.

On the hardware side, the project runs efficiently on standard personal computers with at least 4 GB RAM, although 8 GB RAM provides smoother processing during TF-IDF vectorization and ANN training. A multi-core processor speeds up computations, while a GPU—though optional—can further accelerate neural network training. Overall, the requirements are minimal, making the project compatible with most modern systems.

To implement the sentiment analysis system effectively, the project relies on a stable computing environment capable of handling machine learning workflows. Python serves as the primary programming language due to its extensive support for NLP libraries. Tools such as TensorFlow/Keras are used for building the ANN model, Scikit-learn for preprocessing, TF-IDF transformation, and evaluation, and Pandas for managing textual datasets. Platforms like Google Colab or Jupyter Notebook provide an interactive development interface. Even though the dataset size is moderate, sufficient RAM and processing power ensure efficient preprocessing and training. GPU support, while optional, can improve performance during repeated experiments and hyperparameter tuning.

3.1 Software Requirements

- Python 3.8 or above
- TensorFlow / Keras
- NumPy
- Pandas
- Scikit-learn

- Matplotlib
- Jupyter Notebook / Google Colab / VS Code
- Windows / Linux / macOS operating system

3.2 Hardware Requirements

- Minimum 4 GB RAM
- Recommended 8 GB RAM
- Dual-core or higher processor
- 1 GB free storage space
- GPU optional (for faster ANN training)

3.3 Functional Requirements

- The system must load and preprocess the movie review dataset.
- It must clean text using NLP preprocessing (stopword removal, tokenization, etc.).
- The system must convert text into numerical vectors using TF-IDF.
- It must build an ANN model for sentiment classification.
- The system must train the ANN on the processed data.
- It must evaluate model performance using accuracy, precision, recall, and F1-score.
- It must generate accuracy/loss graphs for interpretation.
- The system must predict sentiment for new review text.

3.4 Non-Functional Requirements

- The system should provide accurate and consistent sentiment predictions.
- It should offer clear, readable outputs.
- The system must run efficiently on basic hardware.
- It should handle noisy or informal text without failing.

4. DESCRIPTION OF MODULES

The ANN-based sentiment classification system is divided into multiple modules, each responsible for a specific stage of the text processing and model training pipeline. These modules work together to enable smooth text handling, feature extraction, training, evaluation, and prediction.

4.1 Data Preprocessing Module

This module loads the movie reviews dataset and prepares the text for analysis. It includes removing unwanted symbols, converting text to lowercase, removing stopwords, and normalizing the review text. This ensures the textual data is clean, consistent, and suitable for TF-IDF conversion.

4.2 ANN Model Building Module

This module constructs the Artificial Neural Network. It defines the input layer based on TF-IDF features, hidden dense layers with ReLU activation, dropout layers to prevent overfitting, and a softmax output layer for multi-class (positive, negative, neutral) classification. The model is compiled using the Adam optimizer and categorical cross-entropy loss.

4.3 Model Training Module

This module trains the ANN model on the processed review data. Training parameters such as batch size, number of epochs, and validation split are defined here. It tracks training and validation accuracy and loss throughout the process.

4.4 Model Evaluation Module

This module evaluates the performance of the trained ANN. It computes accuracy, precision, recall, F1-score, and generates the classification report. It also analyzes how well the model identifies each sentiment class and examines misclassifications.

4.5 Visualization Module

This module generates visual outputs such as training vs. validation accuracy graphs and loss curves. These visualizations help in understanding model learning behavior and detecting overfitting or underfitting.

4.6 Prediction Module

This module applies the trained ANN to new review text. It preprocesses the input, converts it into TF-IDF vectors, and predicts whether the sentiment is positive, negative, or neutral. This enables real-time sentiment prediction for user-entered text.

4.7 Data Splitting Module

This module divides the dataset into training and testing sets (typically 80:20). It ensures data used for training is separate from data used for evaluation. Stratified sampling is used to maintain sentiment class balance. This ensures a fair and unbiased model evaluation.

4.8 TF-IDF Vectorization (Feature Extraction) Module

This module converts textual movie reviews into numerical feature vectors using TF-IDF. Since text cannot be directly fed into neural networks, this vectorization captures important words and weights them based on relevance. It enhances model performance by providing meaningful numerical inputs.

4.9 Output Interpretation Module

This module interprets the model's softmax probabilities and converts them into readable sentiment labels: Positive, Negative, or Neutral. It produces user-friendly outputs, confidence scores, and the final predicted sentiment for easy understanding.

5. IMPLEMENTATION

The implementation of the sentiment analysis system is carried out using Python and an ANN model. First, the movie review dataset is loaded into a Pandas DataFrame, and the review_text and sentiment labels are extracted. The dataset is then split into training and testing sets using stratified sampling to preserve the distribution of sentiment classes.

Next, the text data undergoes preprocessing, including noise removal and normalization. The processed text is transformed into numerical vectors using the TF-IDF technique to convert words into weighted features suitable for neural network input. After vectorization, an ANN model is constructed using TensorFlow/Keras. The architecture includes multiple dense layers activated by ReLU, dropout layers to minimize overfitting, and a final softmax layer for multi-class classification. The model is compiled with the Adam optimizer and trained for several epochs with a validation split to monitor performance. During training, the model learns to differentiate between positive, negative, and neutral sentiments based on patterns in the text data.

After training, the model is evaluated on the test set, where accuracy and a detailed classification report are generated. Visualization tools such as accuracy vs. validation accuracy and loss vs. validation loss graphs help analyze the model's learning behavior. These outcomes highlight how effectively the ANN captures emotional patterns in movie reviews.
Finally, the system includes a prediction module that accepts new review text, preprocesses it, transforms it using the trained TF-IDF vectorizer, and outputs the predicted sentiment. The complete implementation—from preprocessing to visualization—ensures a robust sentiment analysis system capable of supporting automated opinion mining and text analytics.

6. SYSTEM ARCHITECTURE

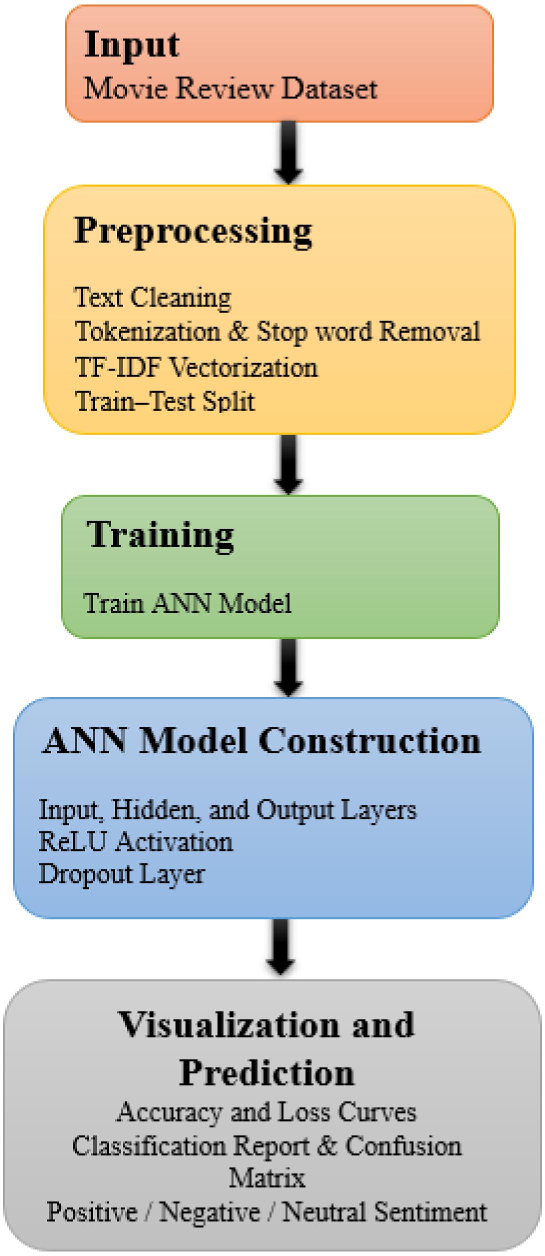

Fig 1: System architecture of sentiment analysis

1. Input – Movie Review Dataset

This is the starting point of the system. The dataset contains text reviews written by users about movies. Each review is labeled with its sentiment (positive, negative, or neutral). These raw text reviews form the foundation on which the model learns emotional patterns and sentence structures.

2. Preprocessing

Preprocessing prepares unstructured text for machine learning.

It includes:

a. Text Cleaning

Removes unnecessary elements such as:

- punctuation
- numbers
- HTML tags
- special characters
- extra spaces

This makes the text cleaner and easier for the model to process.

b. Tokenization & Stopword Removal

Tokenization splits sentences into individual words (tokens).

Stopword removal eliminates commonly used words like *the, is, are,* which don't contribute to sentiment.

c. TF-IDF Vectorization

Since ANN models cannot read text directly, TF-IDF converts words into numerical feature vectors based on:

- how often a word appears in a document, and
- how rare it is across all documents.

This helps the model focus on meaningful words that influence sentiment.

d. Train–Test Split

The dataset is divided into:

- **Training set (80%)** – used to train the ANN
- **Testing set (20%)** – used to evaluate performance

This ensures unbiased testing on unseen data.

3. ANN Model Construction

This step involves designing the structure of the neural network.

a. Input Layer

Receives TF-IDF numeric vectors representing each review.

b. Hidden Layers

Multiple dense layers with **ReLU activation** learn patterns in word usage and sentiment context.

c. Dropout Layer

Randomly drops neurons during training to prevent overfitting and improve generalization.

d. Output Layer

Uses **softmax or sigmoid activation** (depending on binary or multi-class sentiment) to classify the review as:

- Positive
- Negative
- Neutral

4. Training

The ANN model is trained using the training dataset.

During training:

- the model adjusts weights
- reduces loss
- improves accuracy

It learns how different word patterns correspond to different sentiments.

The training process includes:

- batch size
- number of epochs
- validation split

These control how well the model learns.

5. Visualization and Prediction

a. Accuracy and Loss Curves

Graphs showing:

- how accuracy improves over epochs
- how the loss decreases

These help evaluate training stability and detect overfitting.

b. Classification Report & Confusion Matrix

These measures show:

- precision, recall, F1-score
- how many reviews were correctly/incorrectly classified
- performance for each sentiment class

c. Sentiment Prediction (Positive/Negative/Neutral)

Finally, the trained ANN predicts the sentiment of new, unseen movie reviews.

Users can input any review text and instantly get a sentiment classification.

7. CODE IMPLEMENTATION

Algorithm: Sentiment Analysis on Movie Reviews.
Input: Movie Review Text Dataset
Output: Predicted Sentiment (Positive / Negative / Neutral) and Performance Metrics

1. Start

2. Load Dataset

   2.1 Load the movie review dataset from the CSV/text file.

   2.2 Separate the dataset into:

   - Text data X (all review sentences)
   - Target labels y (positive / negative / neutral)

3. Preprocess Text Data

   3.1 Clean text by removing punctuation, numbers, HTML tags, and special characters.

   3.2 Convert all text to lowercase.

   3.3 Tokenize the text into words.

   3.4 Remove stopwords to eliminate unnecessary common words.

   3.5 Apply TF-IDF Vectorization to convert text into numerical feature vectors.

   3.6 Split data into training and testing sets using train_test_split with:

   - test_size = 0.2
   - stratify = y (to preserve label distribution)

4. Build ANN Model

   4.1 Initialize a Sequential model.

   4.2 Add input layer with input dimension = TF-IDF feature size.

   4.3 Add first hidden layer: Dense(128) with ReLU activation.

   4.4 Add Dropout layer with rate 0.3 to reduce overfitting.

   4.5 Add second hidden layer: Dense(64) with ReLU activation.

   4.6 Add output layer:

   - Dense(3, softmax) for Positive/Negative/Neutral (multi-class classification)

5. Compile Model

5.1 Set optimizer = Adam.

5.2 Set loss function = Categorical Cross-Entropy.

5.3 Set evaluation metric = Accuracy.

6. Train Model

6.1 Train the ANN model on training data with:

- Epochs = 20–30
- Batch size = 32
- Validation split = 0.2

6.2 Store training history (training & validation accuracy and loss).

7. Test Model

7.1 Use the trained model to predict sentiment probabilities for test TF-IDF vectors.

7.2 Convert probabilities to class labels using argmax() to choose:

- 0 → Negative
- 1 → Neutral
- 2 → Positive

8. Evaluate Performance

8.1 Compute test accuracy using accuracy_score(y_test, y_pred).

8.2 Generate classification report (precision, recall, F1-score for each sentiment).

8.3 Compute confusion matrix to compare actual vs predicted labels.

9. Visualize Results

9.1 Plot training vs. validation accuracy across epochs.

9.2 Plot training vs. validation loss across epochs.

9.3 Plot confusion matrix as a heatmap.

10. End

8.RESULT

Classification Report:

	precision	recall	f1-score	support
negative	0.74	1.00	0.85	14
neutral	1.00	0.83	0.90	23
positive	0.89	0.85	0.87	20
accuracy			0.88	57
macro avg	0.88	0.89	0.88	57
weighted avg	0.90	0.88	0.88	57

Fig 2: Classification Report

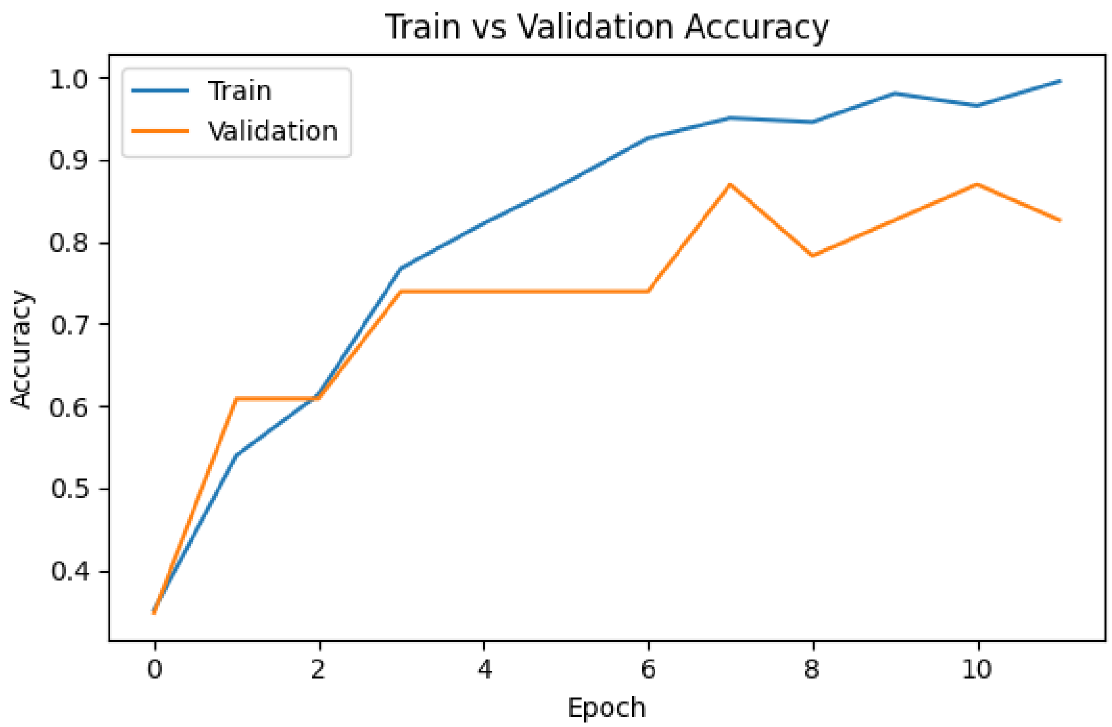

Fig 3: Accuracy Graph

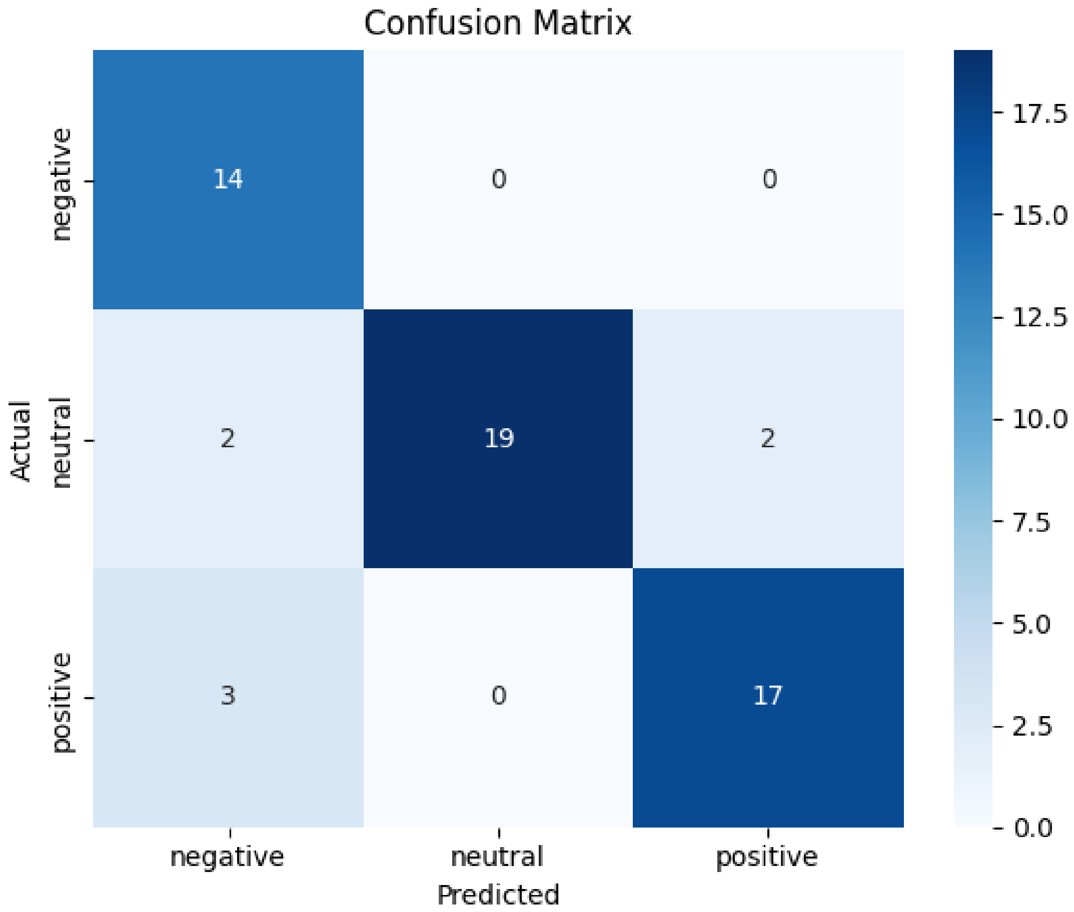

Fig 4: Confusion Matrix

9. CONCLUSION

The Artificial Neural Network–based diabetes prediction system developed in this project successfully demonstrates the capability of deep learning models to analyse medical data and predict health outcomes with good accuracy. By using the Pima Indians Diabetes Dataset and performing proper preprocessing, scaling, model building, and evaluation, the ANN was able to learn meaningful patterns that differentiate diabetic and non-diabetic individuals.

The evaluation metrics, including accuracy, precision, recall, F1-score, and the confusion matrix, confirm that the model performs reliably and generalizes well on unseen data. The training and validation graphs show stable learning behaviour, indicating that the chosen architecture and hyperparameters are effective for this task.

This project also emphasizes the importance of key clinical features such as glucose levels, BMI, age, and diabetes pedigree function in early diabetes detection. Although the model is not a substitute for professional medical diagnosis and is limited by the size and nature of the dataset, it highlights the potential of machine learning to support healthcare professionals in risk assessment and early intervention.

Overall, the project demonstrates that Artificial Neural Networks can play a significant role in building intelligent, data-driven healthcare systems. It provides a strong foundation for future improvements such as using larger and more diverse datasets, incorporating advanced deep learning models, and deploying the system in real-time medical applications.

10. REFERENCES

[1] G. S. Ganie, M. Asif, and M. Rasool, "Ensemble learning methods for diabetes prediction with emphasis on boosting algorithms and feature selection techniques," *Journal of Healthcare Informatics Research*, 2023.

[2] A. Gündoğdu, "Application of XGBoost and hybrid feature selection methods for early diabetes detection," *Biomed Signal Process Control*, 2023.

[3] C. Chang, H. Lin, and S. Huang, "Comparative study of machine learning models and their integration into IoMT-based healthcare systems for diabetes prediction," *IEEE Access*, 2023.

[4] M. Tasin, R. Paul, and K. Rahman, "Evaluation of classical and ensemble machine learning models for diabetes prediction: A case study on clinical datasets," *Computers in Biology and Medicine*, 2022.

[5] A. Madan, S. Prakash, and N. Singh, "Hybrid deep learning architectures for medical diagnosis: Investigation of neural networks for diabetes prediction," *Artificial Intelligence in Medicine*, 2022.

[6] M. Ayat, "A CNN–LSTM hybrid deep learning model for improved medical data classification," *Expert Systems with Applications*, 2024.

[7] R. Kumar and S. Verma, "Comparative analysis of SVM, Decision Trees, and Random Forest for diabetes prediction using the Pima Indians dataset," *International Journal of Advanced Computer Science and Applications*, 2022.

[8] Kaggle, "Pima Indians Diabetes Dataset," 2024.

[9] TensorFlow Developers, *TensorFlow: Large-Scale Machine Learning Framework*, 2015–2024.

[10] Scikit-Learn Developers, *Scikit-Learn: Machine Learning in Python*, 2011–2024.