1. Introduction

This project focuses on designing and implementing an Artificial Neural Network (ANN) to predict whether a bank customer will subscribe to a term deposit. The work follows a structured engineering approach and is mapped to **Complex Engineering Problems (CEPs)** through WP1, WP2, and WP3. The project is divided into four phases: dataset preprocessing, neural network design, model training and evaluation, and optimization.

2. Phase 1: Dataset Selection and Preprocessing

Dataset Selection

The **Bank Marketing Dataset** was used, containing customer demographic, contact, and campaign-related attributes. The target variable is deposit, indicating whether the customer subscribed to a term deposit.

Data Cleaning and Analysis

- The dataset was examined for missing values and inconsistencies. No missing values were found.
- Class distribution analysis revealed an imbalance between deposit and no-deposit classes, introducing a conflict between predictive accuracy and fair class representation.

Feature Engineering and Encoding

- Categorical variables such as job, marital status, education, and contact type were converted using **one-hot encoding**.
- The target variable was encoded into binary form using label encoding.

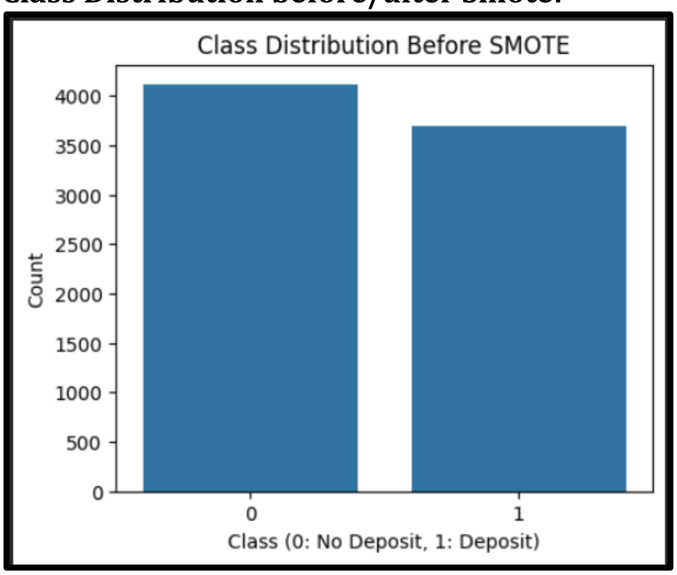
Normalization and Splitting

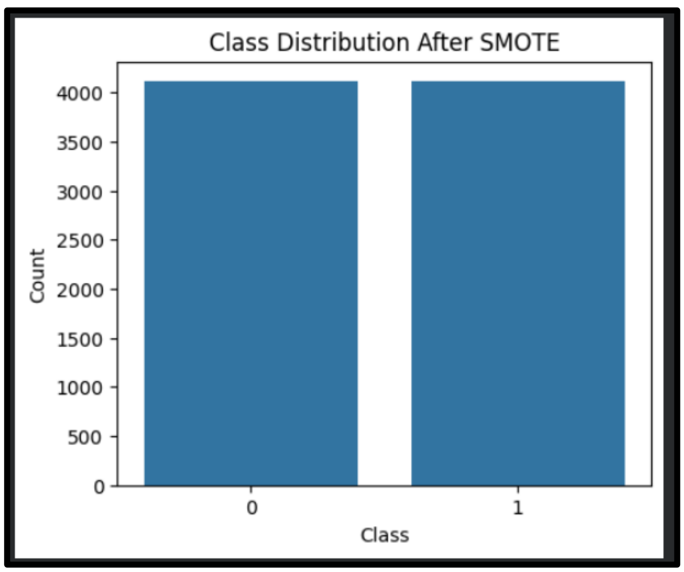
- Features were standardized using StandardScaler to ensure uniform contribution during training.
- The dataset was split into:
    - 70% Training
    - 20% Validation
    - 10% Testing

Handling Class Imbalance

To address class imbalance while maintaining predictive performance, SMOTE (Synthetic Minority Over-sampling Technique) was applied only to the training data. This demonstrates handling conflicting requirements between model bias and data realism.

Class Distribution before/after Smote:





3. Phase 2: Neural Network Design

Architecture Design

A fully connected feedforward ANN was designed with the following structure:

- Input Layer matching the feature dimension
- Hidden Layers: 256 → 128 → 64 neurons

- Output Layer: 1 neuron with sigmoid activation

Activation Functions

- ReLU was used in hidden layers for faster convergence and mitigation of vanishing gradients.
- Sigmoid was used in the output layer for binary classification.

Regularization and Stabilization

- L2 regularization to penalize large weights
- Dropout layers to reduce overfitting
- Batch Normalization to stabilize learning and improve gradient flow

These design choices reflect advanced ANN knowledge and analytical reasoning tailored to tabular data.

Layer (type)	Output Shape	Param #
dense_25 (Dense)	(None, 256)	11,008
batch_normalization_10 (BatchNormalization)	(None, 256)	1,024
dropout_10 (Dropout)	(None, 256)	0
dense_26 (Dense)	(None, 128)	32,896
batch_normalization_11 (BatchNormalization)	(None, 128)	512
dropout_11 (Dropout)	(None, 128)	0
dense_27 (Dense)	(None, 64)	8,256
dropout_12 (Dropout)	(None, 64)	0
dense_28 (Dense)	(None, 1)	65

4. Phase 3: Model Training and Evaluation

Training Strategy

- Optimizer: Adam with a reduced learning rate
- Loss Function: Binary Cross-Entropy
- Batch Size: 32
- Maximum Epochs: 150

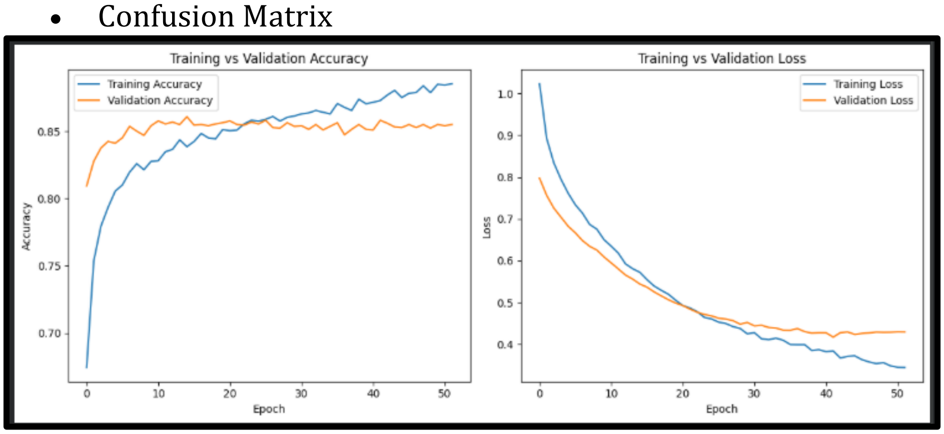
Callbacks

- Early Stopping to prevent overfitting
- ReduceLROnPlateau to dynamically adjust learning rate

Performance Metrics

The model was evaluated using:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix



Interpretation:
Compared to earlier results, this model achieves a better bias–variance trade-off. Regularization, dropout, batch normalization, learning rate scheduling, and early stopping have successfully stabilized training. The model learns meaningful patterns from the data while maintaining strong generalization performance, making it suitable for deployment or final evaluation.

5. Phase 4: Optimization and Problem Solving

Overfitting Mitigation

- Reduced network depth
- Increased dropout rates
- Stronger regularization
- Early stopping

Gradient Stability

- Batch normalization and He initialization were used to prevent vanishing and exploding gradients.

Threshold Optimization

Instead of using a fixed probability threshold of 0.5, the optimal threshold was selected using F1-score maximization, improving the balance between precision and recall.

```
Optimal Threshold: 0.32033336
```

6. Results and Discussion

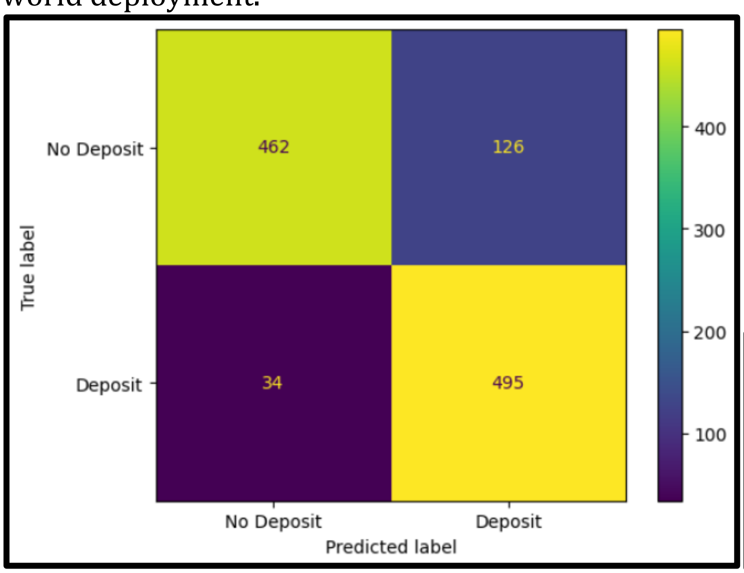
Confusion Matrix Interpretation

- True No Deposit: 462
- False Positives: 126
- False Negatives: 34
- True Deposit: 495

The model demonstrates strong predictive capability, particularly in identifying deposit subscribers, which is critical for banking decision-making. The low false-negative rate indicates effective detection of potential customers.

Overall Performance

The final model achieves a robust balance between accuracy and generalization, making it suitable for real-world deployment.



	precision	recall	f1-score	support
0	0.93	0.79	0.85	588
1	0.80	0.94	0.86	529
accuracy			0.86	1117
macro avg	0.86	0.86	0.86	1117
weighted avg	0.87	0.86	0.86	1117

Interpretation:
The confusion matrix shows strong overall classification performance. The model correctly predicts 462 "No Deposit" and 495 "Deposit" cases, indicating high accuracy for both classes. There are 126 false positives (No Deposit predicted as Deposit) and 34 false negatives (Deposit predicted as No Deposit).

Importantly, the low number of false negatives suggests the model is effective at identifying customers who are likely to make a deposit, which is often a priority in banking applications.

Frontend Testing:

7. Conclusion

This project successfully demonstrates the application of Artificial Neural Networks to a real-world classification problem while addressing complex engineering challenges. Through careful preprocessing, informed architectural design, and systematic optimization, the final model achieves strong performance and reliable generalization, fully meeting the project requirements.