JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

# TRANSFORMER-BASED DEEP NEURAL NETWORK WITH ATTENTION MECHANISM FOR CUSTOMER CHURN PREDICTION IN THE BANKING SECTOR

**[1]R Madhavi, [2]Mr B Siva Kumar Reddy**

[1]P G Student, [2] Associate Professor
[1] Dept.of CSE, MJR College of Engineering and Technology, Piler, India
[2] Dept.of CSE, MJR College of Engineering and Technology, Piler, India

***Abstract:*** Customer churn prediction is vital in the banking sector, as retaining existing clients is more cost-effective than acquiring new ones. Traditional models like Logistic Regression and Random Forest often fail to capture complex nonlinear relationships in customer data. To address this, a **Transformer-Based Deep Neural Network (TDNN)** with a multi-head self-attention mechanism is proposed for efficient churn prediction. The model learns global feature interactions across attributes such as demographics, account balance, credit score, and engagement metrics. **SMOTE** is applied to handle class imbalance, and data preprocessing includes standardization and label encoding. Experimental results show that the proposed TDNN outperforms conventional models in accuracy, precision, recall, F1-score, and AUC. The attention mechanism enhances both interpretability and predictive power, making this approach an effective and scalable solution for proactive customer retention in banking.

## I. INTRODUCTION

In the competitive banking sector, customer retention has become essential for maintaining profitability and growth, as customer churn—the discontinuation of a customer's relationship with a bank—leads to significant revenue loss and increased acquisition costs. With the growing volume of digital banking data generated from customer transactions and interactions, Artificial Intelligence (AI) and Machine Learning (ML) techniques offer valuable tools for predicting potential churners. However, traditional models such as Logistic Regression, Decision Trees, and Random Forests often fail to capture the complex nonlinear dependencies and high-dimensional feature interactions present in customer data. To overcome these challenges, recent advancements in deep learning, particularly transformer architectures equipped with attention mechanisms, have enabled more accurate and interpretable predictive modeling. The proposed Transformer-Based Deep Neural Network (TDNN) leverages multi-head attention to identify and prioritize influential features contributing to churn while modeling global dependencies efficiently. Additionally, the Synthetic Minority Oversampling Technique (SMOTE) is employed to address class imbalance, ensuring fair representation between churned and non-churned customers. This work aims to develop an effective and interpretable TDNN framework that enhances prediction accuracy and supports data-driven decision-making for customer retention in the banking industry.

## II. LITERATURE SURVEY

Several studies have focused on customer churn prediction in the banking sector using various data mining and machine learning approaches. Karvana et al. implemented Decision Trees, Random Forests, and Logistic Regression models to analyze customer behavior and found that ensemble methods offered better predictive accuracy. Kaur and Kaur compared multiple classifiers, including SVM and Random Forest, and concluded that Random Forest achieved the best performance for identifying potential churners. Muneer et al. applied Logistic Regression, Decision Trees, and Artificial Neural Networks, reporting that deep learning models outperformed traditional techniques in accuracy and reliability. Guliyev and Tatoğlu proposed an explainable machine learning framework using Gradient Boosting and SHAP analysis to interpret key churn factors such as account balance and tenure, emphasizing transparency in AI-driven decisions. Rahman and Kumar developed a predictive model using Decision Trees, Gradient Boosting, and Neural Networks, achieving high precision and recall, and highlighted the integration of churn prediction into CRM systems for improved customer retention and profitability.

## III. PROPOSED METHODOLOGY

### 3.1 Problem Statement

Traditional customer churn prediction systems in banking rely on machine learning models such as Logistic Regression, Decision Trees, Random Forests, and SVMs. These models analyze structured data like balance, tenure, credit score, and transaction history but fail to handle complex, nonlinear relationships in large datasets. They also depend heavily on manual feature engineering and are sensitive to data imbalance, leading to biased predictions. Although ensemble methods like Gradient Boosting, XGBoost, and CatBoost improved accuracy, they still struggle to capture hierarchical feature dependencies. Deep learning models such as ANNs and CNNs offer better nonlinear modeling but lack interpretability and the ability to learn global dependencies effectively. Hence, a more advanced and explainable approach is required to improve predictive accuracy and interpretability.

### 3.2 Proposed System

The proposed **Transformer-Based Deep Neural Network (TDNN)** integrates a multi-head attention mechanism to enhance prediction accuracy and interpretability. The TDNN automatically learns complex feature relationships from multidimensional customer data, allowing it to focus on the most influential attributes. The transformer's attention layers capture both local and global dependencies across features, unlike traditional models that treat inputs independently. To handle class imbalance, **SMOTE** is applied to generate synthetic samples for churned customers, ensuring balanced learning. The data undergoes cleaning, label encoding, and feature scaling before training. The TDNN architecture comprises stacked multi-head attention and dense layers with dropout regularization to prevent overfitting. The model is optimized using **Adam** and trained with a binary cross-entropy loss function. Performance is evaluated using accuracy, precision, recall, F1-score, and AUC, supported by ROC and confusion matrix analysis. This architecture provides a scalable, interpretable, and high-performing solution for banking churn prediction.

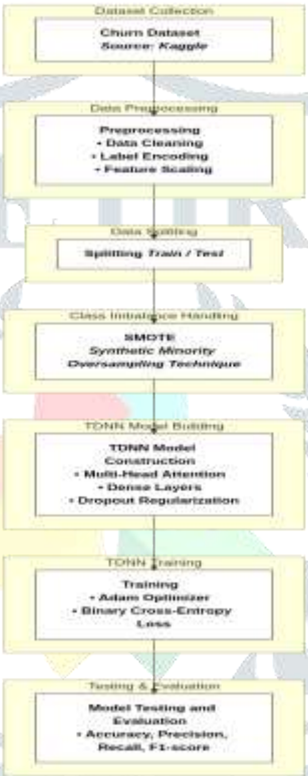



**FIG 1: SYSTEM ARCHITECTURE**

The system consists of data acquisition, preprocessing, model training, testing, and evaluation phases that collectively form an end-to-end predictive pipeline for churn identification. The proposed framework includes eight key modules—**Data Collection**, **Data Preprocessing**, **Data Splitting**, **Class Imbalance Handling**, **TDNN Model Building**, **Model Training**, **Model Testing**, and **Performance Evaluation**. Customer data collected from sources like Kaggle includes demographic, financial, and behavioral attributes. Preprocessing involves cleaning, label encoding, and standardization. Data is split into training and testing sets using an 80–20 ratio with stratified sampling. SMOTE balances churn and non-churn classes. The TDNN model integrates multi-head attention layers with dense layers and dropout for robust feature learning. Training employs the Adam optimizer and binary cross-entropy loss, followed by testing on unseen data. Finally, model performance is validated through metrics such as accuracy, recall, F1-score, and AUC, ensuring reliable churn detection. The TDNN combines transformer attention mechanisms with deep neural layers for superior feature representation and prediction accuracy. Input features are projected into a high-dimensional embedding space, followed by multi-head self-attention layers that learn dependencies among attributes. Each attention block includes residual connections, dropout, and normalization to stabilize training. Feed-forward layers with ReLU activation refine these representations, while dense layers and batch normalization further enhance model generalization. A sigmoid output layer produces churn probability scores. The model is trained using the Adam optimizer (learning rate = 0.001) and binary cross-entropy loss across 50 epochs. The attention mechanism enables the model to focus on key customer attributes, improving interpretability and efficiency. Overall, the TDNN offers a robust and explainable approach for accurate churn prediction, supporting proactive decision-making and customer retention in the banking sector.

## IV. SOFTWARE AND DOMAIN DESCRIPTION

Python is one of the most widely used programming languages for machine learning and deep learning due to its simplicity and extensive library support. Popular libraries include **NumPy**, **SciPy**, **Scikit-learn**, **Theano**, **TensorFlow**, **Keras**, **PyTorch**, **Pandas**, **Matplotlib**, and **Seaborn**. NumPy and SciPy support complex mathematical operations, while Scikit-learn provides a wide range of machine learning algorithms. TensorFlow and PyTorch are powerful frameworks for developing deep neural networks, and Keras offers an easy-to-use interface for rapid model building. Pandas is used for data preprocessing and manipulation, and Matplotlib with Seaborn enables effective data visualization. Additional libraries such as **OpenCV**, **Pillow**, and **ImageAI** support image processing and computer vision tasks.

Data mining, also referred to as data science, is the process of extracting meaningful, previously unknown, and useful information from large datasets. It integrates multiple disciplines, including statistics, machine learning, databases, and high-performance computing, to uncover patterns, correlations, and insights. The primary goals of data mining include **classification**, **association**, and **sequence analysis**, which help in making data-driven predictions and decisions. It involves various stages such as data preprocessing, model creation, testing, and evaluation. Related technologies like **data warehousing** and **OLAP (Online Analytical Processing)** enhance the process by providing centralized storage and multi-dimensional data analysis. In essence, the combination of Python-based tools and data mining techniques enables efficient data analysis, predictive modeling, and intelligent decision-making in various real-world applications such as banking and finance.

***SYSTEM DESIGN***

System design defines the architecture, components, and data flow of a system to ensure efficient processing and accurate functionality. A Data Flow Diagram (DFD) illustrates how input data moves through processes to produce outputs, helping visualize system operations. Unified Modeling Language (UML) provides standardized graphical representations like use case, class, sequence, and activity diagrams to depict interactions, structures, and workflows of the system. Input design focuses on developing accurate, user-friendly data entry methods with proper validation to prevent errors, while output design ensures that processed data is presented clearly and meaningfully to support user decisions. System testing validates the complete functionality of the system through different levels—unit testing checks individual modules, integration testing verifies their interaction, functional and system testing ensure overall performance, and acceptance testing confirms user satisfaction. Both white-box and black-box testing methods are used to ensure correctness, reliability, and efficiency. The testing phase confirmed that all modules worked as expected without major defects, ensuring that the developed system meets its intended objectives effectively.

## V. EXPERIMENTAL SYSTEMS

**MODULE 1-** Data Loading and Exploring

Dataset

| index | customerid | country | gender | creditscore | age | tenure | balance | numofproducts | hascrcard | isactivemember | estimatedsalary | satisfaction_index | churn |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2011499 | Germany | Female | 728 | 46 | 6 | 57066.21 | 2 | 1 | 0 | 144359.64 | 0.6377 | 0 |
| 1 | 2006475 | Germany | Female | 659 | 37 | 0 | 56452.76 | 2 | 1 | 0 | 110292.75 | 0.5045 | 1 |
| 2 | 2013167 | France | Female | 642 | 32 | 2 | 171536.42 | 1 | 1 | 1 | 145600.85 | 0.6986 | 0 |
| 3 | 2000862 | Spain | Female | 651 | 39 | 0 | 138583.98 | 2 | 1 | 0 | 59617.2 | 0.4973 | 1 |
| 4 | 2005970 | Germany | Female | 651 | 49 | 10 | 46573.02 | 1 | 1 | 0 | 67464.23 | 0.5856 | 1 |
| 5 | 2006706 | Germany | Female | 725 | 34 | 1 | 47409.45 | 3 | 0 | 0 | 128697.36 | 0.5516 | 1 |
| 6 | 2003017 | France | Female | 777 | 41 | 1 | 18401.25 | 3 | 1 | 1 | 142132.84 | 0.7005 | 0 |
| 7 | 2003781 | France | Male | 694 | 34 | 0 | 0.0 | 3 | 0 | 1 | 187879.43 | 0.6845 | 0 |
| 8 | 2003898 | Spain | Male | 591 | 22 | 0 | 80593.36 | 1 | 0 | 1 | 44874.65 | 0.6801 | 0 |
| 9 | 2002250 | France | Male | 680 | 50 | 3 | 25636.61 | 3 | 1 | 0 | 68772.65 | 0.5167 | 1 |

Dataset shape / Dataset input features

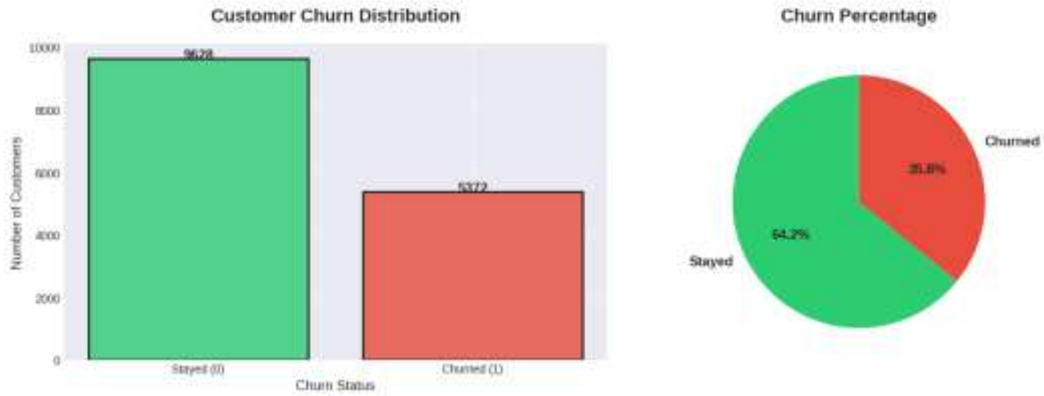
```
df.shape
```

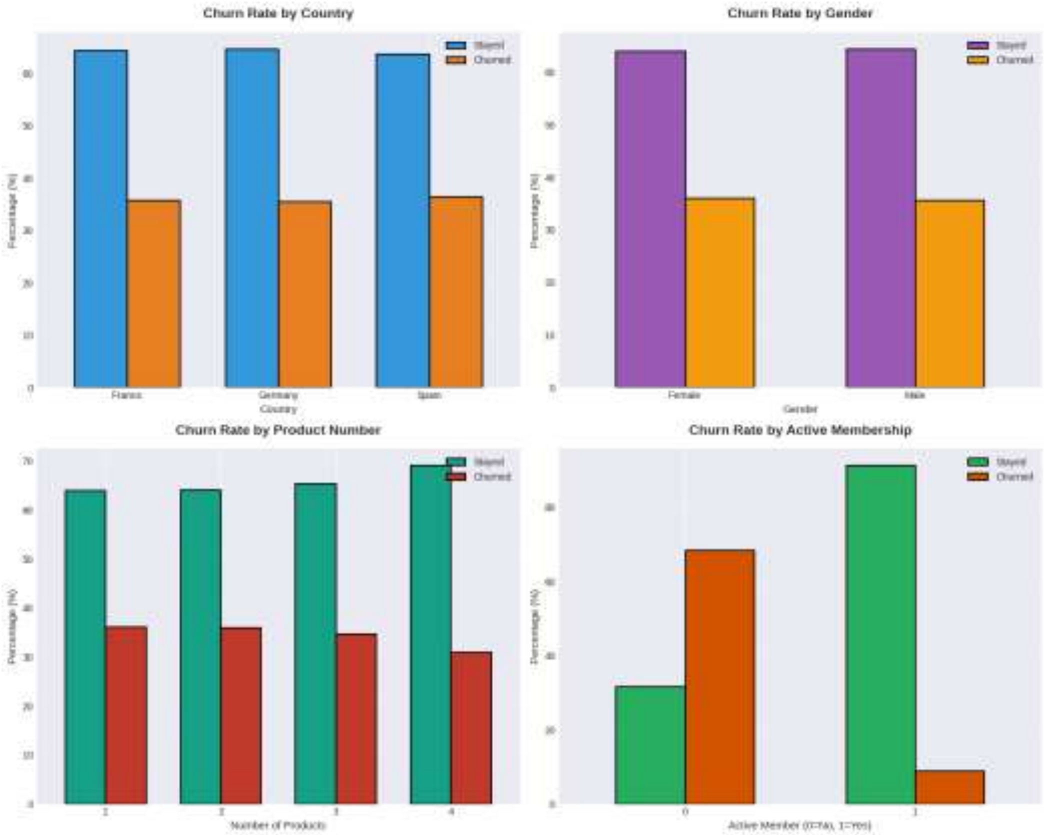
```
(15000, 13)
```

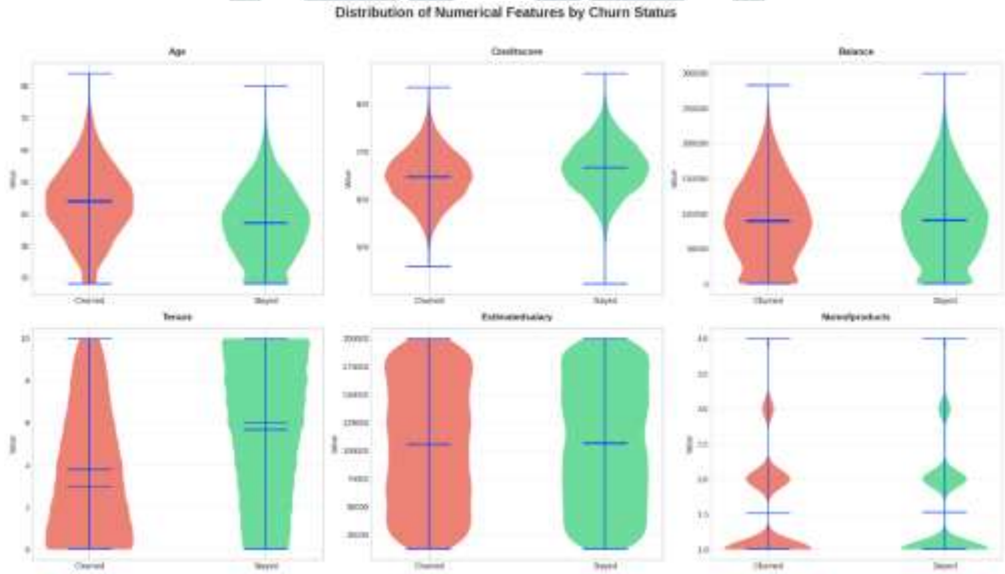
```
df.columns
```

```
Index(['customerid', 'country', 'gender', 'creditscore', 'age', 'tenure',
       'balance', 'numofproducts', 'hascrcard', 'isactivemember',
       'estimatedsalary', 'satisfaction_index', 'churn'],
      dtype='object')
```

Data Visualization




MODULE 2 - Data Preprocessing

Data Cleaning

```
Missing Values Check:
customerid            0
country               0
gender                0
creditscore           0
age                   0
tenure                0
balance               0
numofproducts         0
hascrcard             0
isactivemember        0
estimatedsalary       0
satisfaction_index    0
churn                 0
dtype: int64

No missing values found.

Duplicate Rows: 0
```

Data Encoding

| index | country | gender | creditscore | age | tenure | balance | numofproducts | hascrcard | isactivemember | estimatedsalary | satisfaction_index | churn |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 728 | 46 | 6 | 57066.21 | 2 | 1 | 0 | 144359.64 | 0.6377 | 0 |
| 1 | 1 | 0 | 659 | 37 | 0 | 56452.76 | 2 | 1 | 0 | 110292.75 | 0.5045 | 1 |
| 2 | 0 | 0 | 642 | 32 | 2 | 171536.42 | 1 | 1 | 1 | 145600.85 | 0.6986 | 0 |
| 3 | 2 | 0 | 651 | 39 | 0 | 138583.98 | 2 | 1 | 0 | 59617.2 | 0.4973 | 1 |
| 4 | 1 | 0 | 651 | 49 | 10 | 46573.02 | 1 | 1 | 0 | 67484.23 | 0.5856 | 1 |

Data Scaling

```
array([[ 0.31437304, -0.94966511,  1.23352227, ..., -1.0984222 ,
         0.69500966, -0.24733824],
       [ 0.31437304, -0.94966511, -0.02082212, ..., -1.0984222 ,
         0.06876942, -1.54242806],
       [-0.89475404, -0.94966511, -0.32986349, ...,  0.91039675,
         0.71782641,  0.34478616],
       ...,
       [-0.89475404,  1.05300278,  0.76087076, ...,  0.91039675,
        -0.74683746,  1.750717  ],
       [ 0.31437304,  1.05300278,  0.19732473, ..., -1.0984222 ,
         1.56522757, -0.25414426],
       [-0.89475404, -0.94966511,  2.2151831 , ...,  0.91039675,
         1.17392368, -0.13649722]])
```

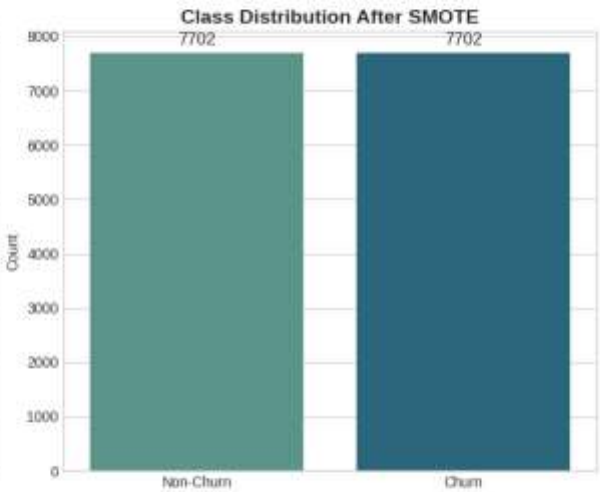
**MODULE 3 –** Data Splitting

Data Splitting (80:20)

```
training set size: (12000, 11), testing set size: (3000, 11)
```

**MODULE 4 –** Class Imbalance Handling using SMOTE

Class imbalance handling to training set

Class Distribution Summary:

| Class | Before SMOTE | After SMOTE |
|---|---|---|
| Non-Churn | 7702 | 7702 |
| Churn | 4298 | 7702 |

**MODULE 5 –** TDNN Model Building

Model: "TDNN_Attention"

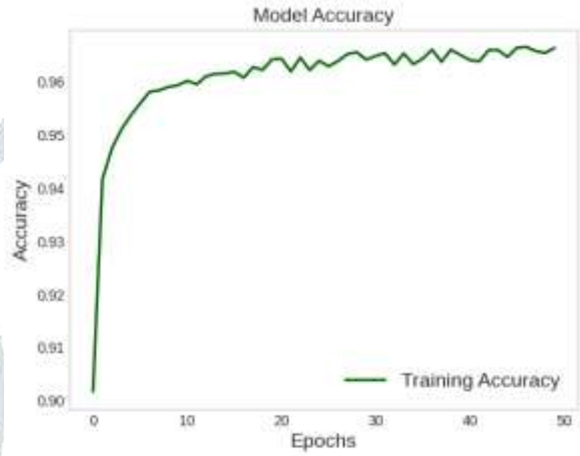
| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| inputs (InputLayer) | (None, 11) | 0 | - |
| dense (Dense) | (None, 128) | 1,536 | inputs[0][0] |
| layer_normalization (LayerNormalizatio… | (None, 128) | 256 | dense[0][0] |
| lambda (Lambda) | (None, 1, 128) | 0 | layer_normalizat… |
| mha_block_0 (MultiHeadAttentio… | (None, 1, 128) | 66,048 | lambda[0][0], lambda[0][0] |
| dropout_1 (Dropout) | (None, 1, 128) | 0 | mha_block_0[0][0] |
| add (Add) | (None, 1, 128) | 0 | lambda[0][0], dropout_1[0][0] |
| layer_normalizatio… (LayerNormalizatio… | (None, 1, 128) | 256 | add[0][0] |
| dense_1 (Dense) | (None, 1, 256) | 33,024 | layer_normalizat… |
| dense_2 (Dense) | (None, 1, 128) | 32,896 | dense_1[0][0] |
| dropout_2 (Dropout) | (None, 1, 128) | 0 | dense_2[0][0] |
| add_1 (Add) | (None, 1, 128) | 0 | layer_normalizat… dropout_2[0][0] |
| layer_normalizatio… (LayerNormalizatio… | (None, 1, 128) | 256 | add_1[0][0] |
| mha_block_1 (MultiHeadAttentio… | (None, 1, 128) | 66,048 | layer_normalizat… layer_normalizat… |
| dropout_4 (Dropout) | (None, 1, 128) | 0 | mha_block_1[0][0] |
| add_2 (Add) | (None, 1, 128) | 0 | layer_normalizat… dropout_4[0][0] |
| layer_normalizatio… (LayerNormalizatio… | (None, 1, 128) | 256 | add_2[0][0] |
| dense_3 (Dense) | (None, 1, 256) | 33,024 | layer_normalizat… |
| dense_4 (Dense) | (None, 1, 128) | 32,896 | dense_3[0][0] |
| dropout_5 (Dropout) | (None, 1, 128) | 0 | dense_4[0][0] |
| add_3 (Add) | (None, 1, 128) | 0 | layer_normalizat… dropout_5[0][0] |
| layer_normalizatio… (LayerNormalizatio… | (None, 1, 128) | 256 | add_3[0][0] |
| flatten (Flatten) | (None, 128) | 0 | layer_normalizat… |
| dense_5 (Dense) | (None, 128) | 16,512 | flatten[0][0] |
| batch_normalization (BatchNormalizatio… | (None, 128) | 512 | dense_5[0][0] |

| | | | |
|---|---|---|---|
| dropout_6 (Dropout) | (None, 128) | 0 | batch_normalizat… |
| dense_6 (Dense) | (None, 64) | 8,256 | dropout_6[0][0] |
| batch_normalizatio… (BatchNormalizatio… | (None, 64) | 256 | dense_6[0][0] |
| dropout_7 (Dropout) | (None, 64) | 0 | batch_normalizat… |
| dense_7 (Dense) | (None, 32) | 2,080 | dropout_7[0][0] |
| dropout_8 (Dropout) | (None, 32) | 0 | dense_7[0][0] |
| output (Dense) | (None, 1) | 33 | dropout_8[0][0] |

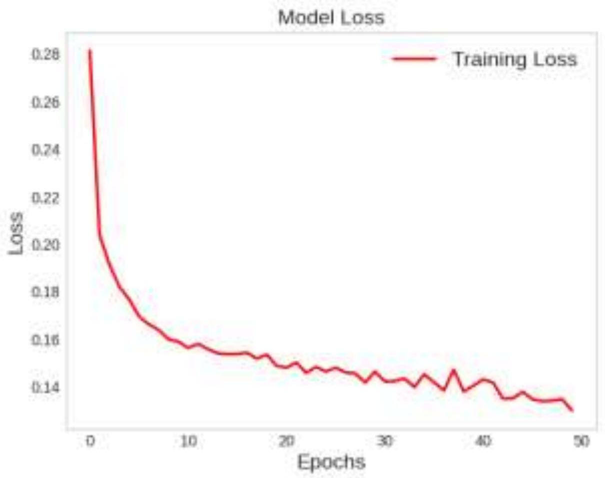
```
Total params: 294,401 (1.12 MB)
Trainable params: 294,017 (1.12 MB)
Non-trainable params: 384 (1.50 KB)
```

MODULE 6 – Train the TDNN Model

Training Accuracy




Training Loss




The proposed Transformer-Based Deep Neural Network (TDNN) model shows excellent learning stability and convergence over 50 training epochs. The training accuracy rapidly rises from about 84% at epoch 1 to over 95% by epoch 10 and stabilizes around 96–97% after epoch 20, indicating strong generalization without overfitting. The loss curve consistently decreases from 0.28 to below 0.14, confirming effective learning with well-tuned parameters such as a learning rate of 1e-3 and dropout regularization. These results demonstrate that the multi-head attention and dense layers in the TDNN efficiently capture feature relationships, achieving high accuracy and low loss. Overall, the model exhibits stable learning, minimal overfitting, and strong predictive capability for customer churn prediction.

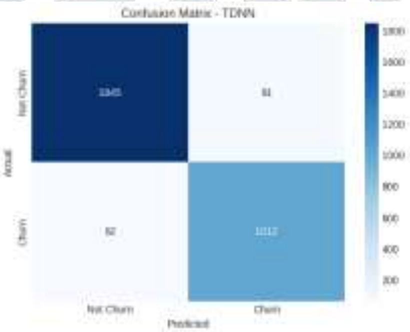
MODULE 7 – Model Testing and Evaluation

Testing set predicted result

| Index | Actual Result | Predicted Result |
|---|---|---|
| 1580 | 0 | 0 |
| 7734 | 1 | 1 |
| 8268 | 0 | 0 |
| 72 | 1 | 1 |
| 5947 | 0 | 0 |
| 3831 | 1 | 1 |
| 10509 | 1 | 1 |
| 10096 | 1 | 1 |
| 3696 | 1 | 1 |
| 5975 | 0 | 0 |
| 2654 | 0 | 0 |
| 1887 | 0 | 0 |
| 10519 | 1 | 1 |
| 7572 | 1 | 1 |
| 13392 | 0 | 0 |
| 10360 | 0 | 0 |
| 3858 | 1 | 0 |
| 12384 | 1 | 1 |
| 9672 | 0 | 0 |
| 5220 | 1 | 1 |
| 8198 | 0 | 0 |
| 2353 | 1 | 1 |
| 5370 | 1 | 1 |
| 10876 | 1 | 1 |
| 402 | 1 | 1 |

After training, the proposed Transformer-Based Deep Neural Network (TDNN) model was tested on unseen data to assess its accuracy and generalization. The predicted results closely matched the actual churn labels, showing that the model effectively distinguishes between churned and non-churned customers. Only a few mismatches were observed, indicating a low error rate and strong reliability. The balanced classification of both classes confirms that SMOTE-based balancing and multi-head attention helped the model learn key patterns from both major and minor classes. Overall, the TDNN model demonstrates high predictive precision, robust generalization, and suitability for real-world banking applications to identify and reduce customer churn.

Confusion Matrix




The confusion matrix highlights the strong classification performance of the proposed Transformer-Based Deep Neural Network (TDNN) model. It correctly identified 1,845 Non-Churn and 1,012 Churn customers, with only 81 false positives and 62 false negatives. This low error rate indicates balanced learning without bias toward either class. The high true positive and true negative counts demonstrate that the model effectively captures key behavioral and transactional patterns influencing churn. The use of multi-head attention and SMOTE balancing further enhances its accuracy and fairness. Overall, the confusion matrix confirms the TDNN model's robustness, reliability, and suitability for real-world banking applications in proactive customer retention.

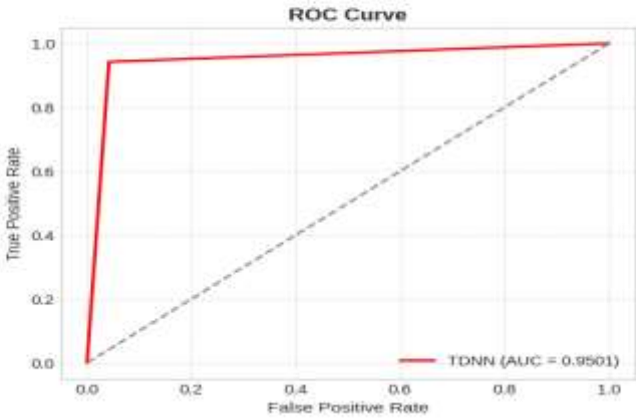
Performance Metrics

Accuracy on test set: 95.23%

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.9675 | 0.9579 | 0.9627 | 1926 |
| 1 | 0.9259 | 0.9423 | 0.9340 | 1074 |
| accuracy | | | 0.9523 | 3000 |
| macro avg | 0.9467 | 0.9501 | 0.9484 | 3000 |
| weighted avg | 0.9526 | 0.9523 | 0.9524 | 3000 |

The proposed Transformer-Based Deep Neural Network (TDNN) model achieved excellent performance in customer churn prediction, with a test accuracy of 95.23%, demonstrating strong learning and generalization. For the non-churn class, it attained a precision of 0.9675, recall of 0.9579, and F1-score of 0.9627, while for the churn class, precision and recall were 0.9259 and 0.9423, yielding an F1-score of 0.9340. These balanced results show that the model effectively detects true churners with minimal false predictions. The macro and weighted average F1-scores of 0.9484 and 0.9524 confirm consistent performance across both classes. The integration of SMOTE balancing, multi-head attention, and dropout regularization enhances accuracy and stability. Overall, the TDNN model outperforms traditional methods, offering a reliable and scalable solution for proactive customer retention in banking.

ROC Curve




The Receiver Operating Characteristic (ROC) curve evaluates the discriminative power of the proposed Transformer-Based Deep Neural Network (TDNN) model in classifying churned and non-churned customers. The curve rises sharply toward the top-left corner, indicating high sensitivity and specificity, with an impressive AUC value of 0.9501. This confirms the model's strong ability to distinguish between the two classes and maintain an excellent balance between recall and false positives. The integration of multi-head attention, dense layers, and dropout regularization enhances feature learning and generalization. Overall, the high AUC score validates the TDNN model's robustness, precision, and suitability for real-world banking churn prediction and proactive customer retention.

## V. CONCLUSION

The proposed Transformer-Based Deep Neural Network (TDNN) model for customer churn prediction in the banking sector effectively combines multi-head attention with deep learning for highly accurate classification. Using SMOTE for class balancing and attention layers for feature learning, the model achieved 95.23% accuracy and an AUC of 0.9501, surpassing traditional methods. It demonstrated strong generalization with balanced precision, recall, and F1-scores. Overall, the TDNN framework offers a robust and interpretable solution for proactive churn management, helping banks identify at-risk customers early and enhance customer retention and profitability.

## VI. REFERENCES


[1] Alexandropoulos, S. A. N., Kotsiantis, S. B., & Vrahatis, M. N. (2019). Data preprocessing in predictive data mining. *The Knowledge Engineering Review*, *34*, 1–33. https://doi.org/10.1017/S026988891800036X

[2] Baran, R. J., & Galka, R. J. (2017). *Customer relationship management: The foundation of contemporary marketing strategy*. Routledge.

[3] Biau, G., & Scornet, E. (2016). A random forest guided tour. *TEST, 25*(2), 197–227. https://doi.org/10.1007/s11749-016-0481-7

[4] Cunningham, P., & Delany, S. J. (2021). k-Nearest neighbour classifiers – A tutorial. *ACM Computing Surveys, 54*(6), 1–25. https://doi.org/10.1145/3459665

[5] Dawes, J., & Swailes, S. (1999). Retention sans frontieres: Issues for financial service retailers. *International Journal of Bank Marketing, 17*(1), 36–43. https://doi.org/10.1108/02652329910254037

[6] Dolatabadi, S. H., & Keynia, F. (2017, July). Designing of customer and employee churn prediction model based on data mining method and neural predictor. *Proceedings of the 2nd International Conference on Computer and Communication Systems, Krakow, Poland*, 74–77. https://doi.org/10.1109/CCOMS.2017.8075270

[7] Elyusufi, Y., & Ait Kbir, M. (2022). Churn prediction analysis by combining machine learning algorithms and best features exploration. *International Journal of Advanced Computer Science and Applications, 13*(7), 615–622. https://doi.org/10.14569/IJACSA.2022.0130773

[8] Guliyev, H., & Tatoğlu, F. Y. (2021). Customer churn analysis in banking sector: Evidence from explainable ma-chine learning models. *Journal of Applied Microeconometrics, 1*(2), 85–99. https://doi.org/10.53753/jame.1.2.03

[9] Hameed, M., & Naumann, F. (2020). Data preparation: A survey of commercial tools. *ACM SIGMOD Record, 49*(3), 18–29. https://doi.org/10.1145/3444831.3444835

[10] Jakkula, V. (2006). *Tutorial on support vector machine (SVM).* School of EECS, Washington State University. https://course.ccs.neu.edu/cs5100f11/resources/jakkula.pdf

[11] Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C., Silverman, R., & Wu, A. Y. (2000). The analysis of a simple *k*-means clustering algorithm. *Proceedings of the 16th Annual Symposium on Computational Geometry*, 100–109. https://doi.org/10.1145/336154.336189

[12] Karvana, K. G. M., Yazid, S., Syalim, A., & Mursanto, P. (2019, October). Customer churn analysis and predic-tion using data mining models in banking industry. *Proceedings of the International Workshop on Big Data and Information Security, Bali, Indonesia*, 33–38. https://doi.org/10.1109/IWBIS.2019.8935884

[13] Kaur, I., & Kaur, J. (2020, November). Customer churn analysis and prediction in banking industry using ma-chine learning. *Proceedings of the Sixth International Conference on Parallel, Distributed and Grid Computing, Wak-naghat, India*, 434–437. https://doi.org/10.1109/PDGC50313.2020.9315761

[14] Muneer, A., Ali, R. F., Alghamdi, A., Taib, S. M., Almaghthawi, A., & Abdullah Ghaleb, E. A. (2022). Predicting customers churning in banking industry: A machine learning approach. *Indonesian Journal of Electrical Engi-neering and Computer Science, 26*(1), 539–549. https://doi.org/10.11591/ijeecs.v26.i1.pp539-549

[15] Olaniyi, A. S., Olaolu, A. M., Jimada-Ojuolape, B., & Kayode, S. Y. (2020). Customer churn prediction in bank-ing industry using k-means and support vector machine algorithms. *International Journal of Multidisciplinary Sciences and Advanced Technology, 1*(1), 48–54.

[16] Pradana, M. G., & Ha, H. T. (2021). Maximizing strategy improvement in mall customer segmentation using k-means clustering. *Journal of Applied Data Sciences, 2*(1), 19–25. https://doi.org/10.47738/jads.v2i1.18

[17] Quinlan, J. R. (1990). Decision trees and decision-making. *IEEE Transactions on Systems, Man, and Cybernetics, 20*(2), 339–346. https://doi.org/10.1109/21.52545.