Financial Data Analysis Using CNN With Feature Selection

Alhakam Salih and Hiren Joshi
Department of Computer Science, Gujarat University, 380009 Ahmedabad, India
alhakam.a.allawie@tu.edu.iq, hdjoshi@gujaratuniversity.ac.in

Keywords: Convolutional Neural Network (CNN), Feature Selection, ANOVA, Mutual Information (MI), Chi-Square.

Abstract: In the age of big data, the financial services industry generates massive volumes of complex and high-dimensional data that demand advanced analytical solutions. Convolutional Neural Networks (CNNs), originally developed for image analysis, have demonstrated strong capabilities in learning rich, hierarchical features directly from raw financial data, making them well-suited for tasks such as credit scoring, fraud detection, and risk assessment. This paper presents a proposed CNN model tailored for financial data analysis. The model efficiently captures the underlying patterns and nonlinear relationships within the data, which is considered one of the reasons for the model's high accuracy. To further enhance performance, a feature selection (FS) step was integrated before training, enabling the model to focus on the most informative attributes. Experimental results on the Santander Customer Transaction Prediction dataset show that the proposed CNN model, when combined with feature selection, achieved a remarkable 100% accuracy in classification tasks. These findings underscore the potential of CNN-based frameworks, enhanced by targeted feature selection, to transform financial data analysis. The approach enables more accurate, scalable, and automated decision-making across critical financial sector applications.

1 INTRODUCTION

Before the 1996 launch of the first internet banking application in the US by Citibank and Wells Fargo Bank, banks exclusively offered in-person services to their clients [1]. The adoption of online credit card usage followed the launch of internet banking. Over the past ten years, this has grown significantly, and services like social networking, online banking, working from home, e-commerce, and online payment systems have also been launched and are now extensively utilized [2]. In today's data-driven financial landscape, understanding and anticipating customer behavior is critical for businesses seeking to enhance decision-making, improve customer experiences, and reduce operational risks [3]. One of the most significant applications of artificial intelligence (AI) in this domain is customer transaction prediction, which involves forecasting future financial activities based on historical transaction data. Accurate transaction prediction supports a wide range of strategic objectives, including personalized marketing, dynamic credit scoring, fraud detection, and inventory optimization [4]. With the explosive growth in digital transactions and the increasing complexity of consumer behavior, traditional data analysis and machine learning techniques often fall short in extracting deep insights from high-dimensional and sequential transaction data [5]. This has led to a growing interest in deep learning approaches, particularly Convolutional Neural Networks (CNNs), which have demonstrated exceptional capabilities in capturing hidden patterns, temporal dependencies, and contextual relationships within large datasets [6]. CNNs, originally designed for image recognition tasks, are now being adapted to structured financial data for their ability to automatically learn hierarchical features without the need for manual feature engineering. When applied to customer transaction prediction, CNNs treat transaction histories as structured inputs such as time series matrices or event sequences and process them through layered convolutional operations to extract meaningful insights [7]. This study explores the use of a proposed CNN-based model for predicting customer transactions. By leveraging the power of deep feature extraction and integrating effective data preprocessing techniques, the model aims to achieve high prediction accuracy and generalizability across varied customer segments.

The remainder of this paper is organized as follows: Section 2 reviews related work in transaction prediction and deep learning applications. Section 3 presents the methodology and the proposed CNN model architecture. Section 4 provides the results and performance analysis, followed by conclusions and future work in Section 5.

2 RELATED WORKS

In recent years, the increasing availability of digital financial data has driven substantial research interest in customer transaction prediction. Traditional approaches have largely relied on statistical models and classical machine learning techniques such as logistic regression, decision trees, and support vector machines to forecast transaction patterns and customer behavior. While these methods provide a foundation for predictive modeling, they often struggle to capture the complex, nonlinear, and temporal dependencies inherent in transaction data. With the emergence of Artificial Intelligence (AI) and deep learning, more sophisticated models have been proposed to address these limitations. In particular, Convolutional Neural Networks (CNNs) have gained attention for their ability to automatically learn multi-level features from raw or minimally processed data. Originally applied in computer vision tasks, CNNs have since been adapted for financial applications, including fraud detection, credit scoring, and transaction categorization, due to their high accuracy, scalability, and end-to-end learning capabilities. This section provides an overview of the existing literature on customer transaction prediction, highlighting both conventional and deep learning-based approaches. It focuses on how AI techniques, particularly CNNs, have been leveraged to improve prediction accuracy and adapt to large-scale, real-time financial environments. The review also identifies key research gaps and motivations for the development of the proposed CNN model introduced in this study.

A. Chouiekh and E. Ibn El Haj [8] used deep convolutional neural networks to detect and classify customer loss using a labeled dataset of 18,000 prepaid users. The method was based on call detail records from a telecom company. The researchers found that DCNN outperformed other machine learning techniques with an accuracy of 89%, demonstrating the effectiveness of this approach. M. Nabipour et al. [9] compared two deep learning techniques, Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM), with nine machine learning models. The input values were 10 technical indicators from ten years of historical data. The results showed that RNN and LSTM performed significantly better than other prediction models for continuous data, with an accuracy of 87%. M. Pondel et al. [10] presented a deep learning model for predicting client attrition in e-commerce, focusing on real data with 75% of buyers being one-time customers. The model showed promise with 74% accuracy, 78% precision, and 68% recall, indicating potential for increased client retention and improved business performance. However, the prediction method may be erroneous. A. Khattak et al. [11] provided a successful strategy for predicting customer turnover based on a hybrid deep learning model known as BiLSTM-CNN. The goal is to effectively estimate customer turnover using benchmark data while also improving the accuracy of the churn prediction process. The experimental results reveal that when trained, tested, and validated on the benchmark dataset, the proposed BiLSTM-CNN model achieved a remarkable 81% accuracy. C. Ku et al. [12] suggested combining investors' expertise with a Long-Short-Term Memory (LSTM) algorithm to predict stock prices. The strategy involves gathering technical indicators from investors and using them as input to the LSTM model. The model is trained and evaluated on a dataset of 100 stocks, and its accuracy is measured using parameters like average forecast accuracy, average cumulative return, Sharpe ratio, and maximum drawdown. The simulation results show that the proposed model outperforms other strategies in terms of accuracy (57%), highlighting the potential of integrating investor knowledge with machine learning algorithms. J. Li et al. [13] developed a CNN-LSTM and attention mechanism-based credit risk prediction model for listed businesses. This method combines a CNN model with an LSTM model for long-term time-series prediction, reducing data complexity, increasing training and calculation speeds, and resolving historical data gaps. The model also includes an attention mechanism for optimization. The model achieved an accuracy of 98.43%, making it a significant contribution to credit risk prediction in listed businesses. O. Seymen et al. [14] compared various machine learning techniques for predicting future customer loss in the retail sector. The deep learning model based on CNN performed better in classification, with a 97.62% accuracy rate compared to the ANN with 96%. The study highlights the importance of deep learning in churn prediction. Table 1 shows the summary of these works.

The proposed CNN model aims to solve several limitations identified in prior research. First, it is designed to be domain-independent, allowing applications across various sectors, including telecom, retail, e-commerce, and finance. To address low recall and prediction errors seen in earlier models, we apply optimized preprocessing, class balancing, and regularization. Unlike complex hybrid models, the proposed model's architecture remains lightweight yet powerful, reducing training time without sacrificing accuracy. The proposed model also eliminates reliance on external expert inputs, making the model fully autonomous. By avoiding excessive model complexity, we ensure fast computation, making it suitable for real-time deployment. Feature selection is used to improve learning efficiency and reduce noise. The model is optimized for performance even in unbalanced or sparse datasets. It also generalizes well to unseen customer behaviors. Overall, the proposed CNN approach offers a scalable, accurate, and adaptable solution for customer transaction prediction.

3 METHODOLOGY

The proposed customer transaction prediction system, as shown in (Fig. 1), based on the proposed CNN, begins with the Santander dataset, which is first split into training and testing sets, followed by a data preprocessing stage to ensure quality input. Feature selection is then conducted using three techniques: Mutual Information, Chi-Square, and ANOVA to extract the most informative features and reduce dimensionality. These selected features are fed into a deep Convolutional Neural Network (CNN) composed of multiple convolutional layers with increasing filter sizes (16 to 1024), activated by Leaky ReLU to overcome vanishing gradients. Max-pooling layers reduce feature map dimensions, followed by a flattening step and fully connected dense layers. A softmax activation function at the output layer classifies the transaction outcome. Training is conducted on features selected by Mutual Information and Chi-Square, while ANOVA-selected features are used in the testing phase to evaluate performance. This architecture aims to enhance predictive accuracy and overcome limitations of previous models by integrating robust feature selection with deep learning.

3.1 The Dataset Description

The dataset used in this study originates from the Santander Customer Transaction Prediction challenge hosted on Kaggle by Banco Santander. It contains over 200,000 records and 200 anonymized numerical features, with a binary target variable indicating whether a customer has made a specific transaction (1) or not (0). Due to anonymization, the feature names and descriptions are unavailable, making this dataset ideal for experimenting with different feature selection techniques. The data poses several real-world challenges common in financial datasets, including high dimensionality, potential multicollinearity among features, and imbalanced class distribution. These complexities make it a strong benchmark for evaluating how feature selection methods can help reduce redundant data, enhance model interpretability, and improve predictive performance. Before applying feature selection and classification, the data[1] undergoes preprocessing steps such as handling missing values, scaling features, and splitting into training and testing sets to ensure reliable evaluation.

3.2 Dataset Splitting (1st Stage)

To ensure robust model evaluation and avoid overfitting, the provided dataset is divided into two main subsets training set and the testing set. Specifically, the biggest amount of the data is used for training the model and optimizing its parameters, while the remaining is reserved for testing and evaluating the model's predictive accuracy [15]. In this work, the Santander Customer Transaction Prediction dataset is divided into training and testing subsets using a 70:30 ratio to ensure reliable evaluation and reduce the risk of overfitting. The training set, comprising 70% of the data, is used to develop and optimize the predictive model, while the remaining 30% is reserved exclusively for testing to assess the model's performance on unseen data. This split allows for a balanced approach to model validation and performance benchmarking. Cross-validation techniques are applied during the training phase to further ensure that the model generalizes well across different subsets and is not tailored only to the training data. This method supports robust performance assessment, which is critical when working with high-dimensional, anonymized financial data like this.

[1]The dataset https://www.kaggle.com/competitions/santander-customer-transaction-prediction/data?select=train.csv](https://www.kaggle.com/competitions/santander-customer-transaction-prediction/data?select=train.csv).

Table 1: Summary of related works.

Ref. No.	Year	Author	Technique	Dataset	Accuracy	Limitations
[8]	2020	A. Chouiekh and E. Ibn El Haj	Deep CNN	Call detail records (18,000 prepaid users)	89%	Limited to telecom data; not compared on diverse datasets
[9]	2020	M. Nabipour et al.	RNN & LSTM	10 years of technical indicators	87%	Limited to continuous stock data; requires large historical datasets
[10]	2021	M. Pondel et al.	Deep Learning	E-commerce customer data	74%	The prediction method may be erroneous; lower recall (68%)
[11]	2023	A. Khattak et al.	BiLSTM-CNN	Benchmark churn dataset	81%	Lacks generalizability across domains
[12]	2023	C. Ku et al.	LSTM with investor expertise	100 stocks data	57%	Low accuracy; relies on expert knowledge input
[13]	2023	J. Li et al.	CNN-LSTM + Attention	Listed business financial data	98.43%	Focused on credit risk only; high model complexity
[14]	2023	O. Seymen et al.	ANN + CNN	Retail customer data	ANN: 96% CNN: 97.62%	ANN needs further optimization

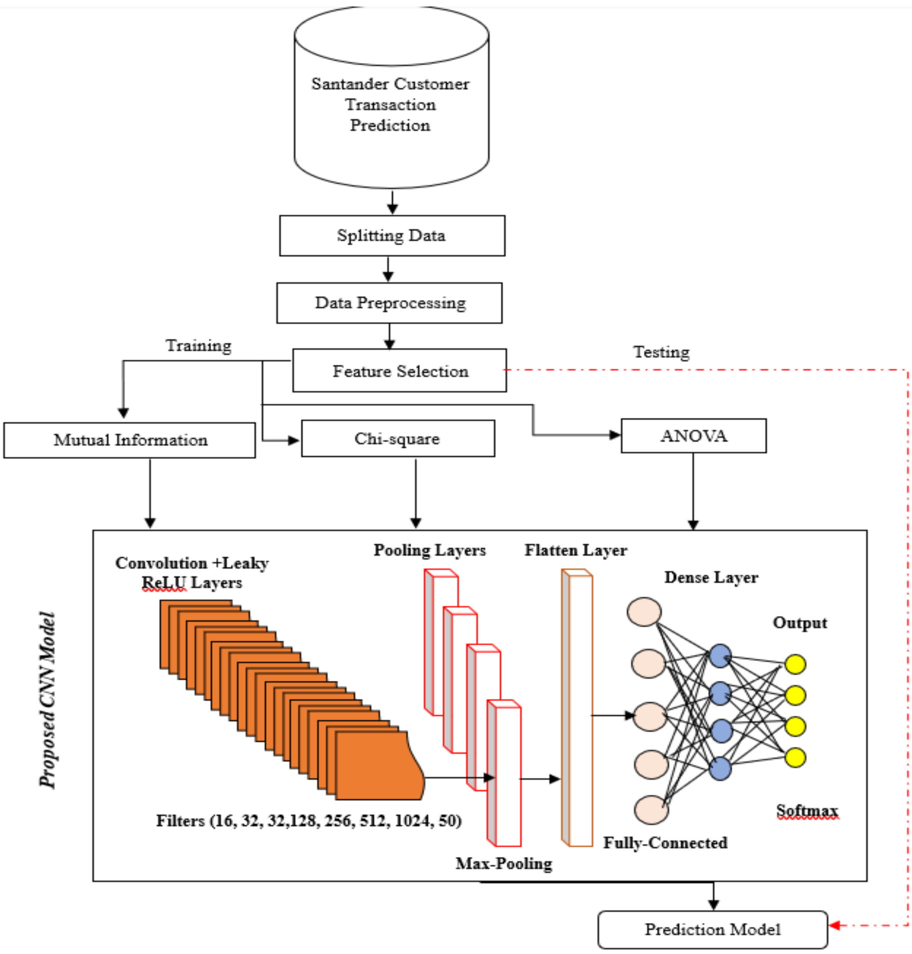
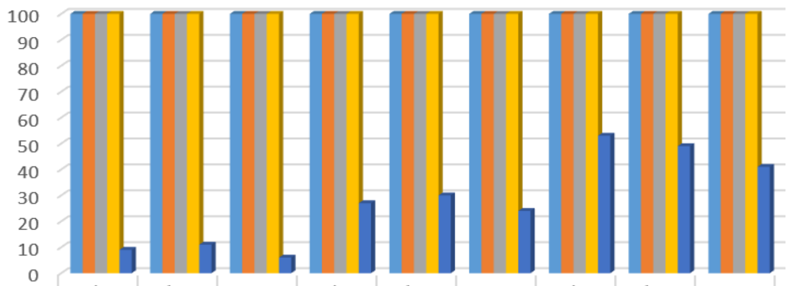


Figure 1: The proposed system architecture.



	mi10	ch10	nova10	mi30	ch30	nova30	mi50	ch50	nova50
prs	100	100	100	100	100	100	100	100	100
rec	100	100	100	100	100	100	100	100	100
f1	100	100	100	100	100	100	100	100	100
acc	100	100	100	100	100	100	100	100	100
time in second	9	11	6	27	30	24	53	49	41

Figure 2: The proposed CNN model results comparison.

3.3 Preprocessing Stage (2nd Stage)

This stage includes some of the processes as follows:

1) Missing Value [16]. To ensure the quality and consistency of the data used in this work, missing values in the Santander Customer Transaction Prediction dataset are carefully handled during the preprocessing phase. Although the dataset is generally clean and anonymized, any missing entries are addressed to prevent issues during model training. If missing values are found, they are either imputed using statistical techniques such as mean or median imputation, depending on the distribution of the feature, or, in the case of minimal missingness, the affected rows may be removed entirely. This step is essential to maintain the integrity of the data, ensure reliable model learning, and avoid introducing bias or distortion into the prediction results.
2) Normalization. One of the initial processing steps for processing the dataset is applied before the data is used, such as increasing or decreasing the range of values. Normalization is convenient and useful in dataset problems that depend on classification, the conversion of feature values for a specific and small range, such as 0 to 1. There are many ways to normalize, such as Z-normalization and Min-Max.

Normalization. Min-Max Normalization is a linear transformation technique used in processors in which preserving the relationship between the original dataset is important. In addition, it is considered one of the simple techniques that are suitable for the dataset within predefined limits [17], [18]. Normalization is done according to (1).

$$x\ new = \frac{x - \min(x)}{\max(x) - \min(x)}. \quad (1)$$

Where x new represents the normalized x. While the dataset contains 200 anonymized numerical features with potentially varying scales and distributions, normalization is used to rescale the values to a common range typically between 0 and 1 or to a standard normal distribution (zero mean and unit variance). This process prevents features with larger numeric ranges from dominating the learning process and helps many machine learning models, especially those based on distance or gradient-based optimization (like CNNs), to converge faster and perform more accurately. By applying normalization, the training process becomes more stable, and the model achieves better generalization on unseen data.

3.4 Feature Selection Stage (3rd Stage)

For customer transaction prediction, effective feature selection plays a vital role in improving classification performance and computational efficiency. In our work, we aim to identify the most informative features from the high-dimensional Santander dataset, which contains 200 anonymized numerical attributes. Not all features contribute equally to the prediction task; some may be redundant or irrelevant. To address this, three distinct feature selection techniques were applied in the proposed model to evaluate and rank feature importance. This process involves assessing each feature’s contribution to distinguishing between customers who perform a transaction (class 1) and those who do not (class 0). By retaining only the top “k” features based on their significance scores, we eliminate noise, reduce dimensionality, and enhance model interpretability and training speed. This targeted selection ensures that the classifier focuses on the most predictive patterns, thereby boosting overall accuracy.

3.4.1 Select Features Based on the Mutual Information Technique

Mutual information is frequently used in feature selection because it can identify non-linear interactions between numerous variables. The majority of mutual information-based feature selection methods use the two (2) and (3), which compute the relevance of a feature subset and redundancy, respectively.

$$\mathrm{Red} = \mathrm{MI}\ (\mathrm{s_1.s_2.....s_m}), \quad (2)$$

$$\mathrm{Rel} = \mathrm{MI}\ (\mathrm{S.C}), \quad (3)$$

where C is the class label, S is the feature set, which contains m features $\mathrm{s_1.....s_m}$. By eliminating all unnecessary characteristics, feature selection aims to provide an ideal feature subset. Hence, the best feature subset minimizes the quality measure specified in (4).

$$\mathrm{F} = -\alpha * \mathrm{Rel} + (1-\alpha) * \mathrm{Red}, \quad (4)$$

where is used to regulate how much relevance and redundancy contribute to the fitness score [19]. In this work, Mutual Information (MI) is used as one of the primary feature selection techniques due to its effectiveness in measuring the dependency between each feature and the target variable. MI quantifies how much knowing the value of a feature reduces uncertainty about the class label, making it particularly valuable in identifying features with high predictive power. Since the Santander dataset is high-dimensional and anonymized, MI helps in uncovering non-linear relationships between features and the binary transaction outcome (0 or 1). By selecting features with the highest MI scores, we significantly reduce the number of irrelevant or redundant inputs, leading to better generalization, faster training, and improved classification accuracy. So, incorporating MI into the feature selection phase contributed to refining the model's focus on the most informative signals, which enhanced performance and interpretability.

3.4.2 Select Features Based on the Chi-Square Technique

The chi-squared test, which assesses deviation from the predicted distribution when the feature event is independent of the class value, is a numerical analysis. True positives (TP), false positives (FP), true negatives (TN), false negatives (FN), probability of a no. of positive instances (P_{pos}), and probability of the no. of negative cases (P_{neg}). There are some of the metrics used to compute the chi-square value [20].

$$chi-square = t\left(t_p\left(t_p+f_p\right)p_{pos}\right) + t\left(f_n\left(f_n+t_n\right)p_{pos}\right) + t\left(f_p\left(t_p+f_p\right)p_{neg}\right) + t\left(t_n\left(f_n+t_n\right)p_{neg}\right). \quad (5)$$

3.4.3 Select Features Based on the ANOVA Technique

Select features based on ANOVA: For each continuous predictor, it was intended to run an ANOVA F-test to determine whether or not all of the various classes of Y have the same mean as X. Applying the following notation:

$$s_j^2 = \sum_{i=1}^{N_j}\left(X_{ij}-\bar{X}_j\right)^2/\left(N_j-1\right), \quad (6)$$

$\bar{x}$: The grand mean of predictor X:

$$\bar{x} = \sum_{j=1}^{J} N_j\bar{X}_j/N. \quad (7)$$

The aforementioned notations depend on pairs of (X, Y). Then, using the F statistic, the p-value is determined by p-value = Prob {F (J-1, N-J) >F}: where,

$$F = \frac{\sum_{j=1}^{J} N_j(\bar{X}_j-\bar{X})/(J-1)}{\sum_{j=1}^{J}(N_j-1)s_j^2/(N-1)}. \quad (8)$$

A random variable called F (J-1, N-1) has degrees of freedom J-1 and N-J and follows the F distribution. A predictor's p-value should be set to zero if the denominator is zero. By ranking the predictors in ascending order depending on the p-value, the predictor is ranked. If there are ties, sort by F in order of importance first, then if there are still ties, sort by N from lowest to highest [21].

3.5 Prediction Stage Using the Proposed CNN Model

Due to its extensive network depth, CNN is categorized as a deep neural network. With each layer, CNN gains the ability to recognize a wider variety of things. Images of various resolutions are processed via image processing, with the output from each image processed and used as input to the next layer [22]. Feature selection techniques (Mutual information, Chi-square, and Analysis of Variance (ANOVA)) and classification are two distinct parts of the proposed CNN architecture. The convolutional, Leaky Rectified Linear Unit (LeakyReLU), and pooling are the three operations following the feature selection portion [23].

3.5.1 Convolutional Layer

Among filters and data as input. To build a feature map, the filter will travel through the data and connect the input and filter value with a "dot" operation. Optimization of the convolutional layer using modified filter size, stride, and zero padding [24],

3.5.2 LeakyReLU

It is an effort to address the fading ReLU issue. When x<0, instead of being zero, a leaky ReLU will have a slight negative slope (of 0.01, or so). The equation in (9), where α is a tiny constant, is computed using the function [25].

$$f(x) = 1(x < 0)(\alpha x) + 1(x >= 0)(x). \quad (9)$$

3.5.3 Pooling Layer

The convolutional layer's output is received by the pooling layer, which also minimizes the amount of data in this layer. Filters of various sizes and strides make up the pooling layer, which travels over the feature map area. Max pooling and mean pooling are the pooling layers that are frequently used in their application [26].

3.5.4 Fully Connected Layer

Processing of the feature maps from the feature selection layers is done by the fully connected layer. In contrast to the convolutional layer, which links neurons just to certain parts of the input, the fully connected layer connects all neurons to enable linear data classification, and the fully connected layer flattens multidimensional feature map arrays into one-dimensional arrays [27].

3.5.5 Softmax Activation

A different kind of Logistic Regression, called Softmax Classifier, may classify more than two classes. The output of the last layer may be transformed to its underlying probability distribution utilizing Softmax. Softmax has the advantage that the output probability can be between 0 and 1 and that the sum of the probabilities is 1 [28].

The suggested CNN model comprised 27 layers as follows:

The nine-layer Convolutional Neural Network (CNN).

- 8 Leaky ReLU layers;
- 6 Max-pooling layers;
- 1 Flatten layer;
- 3 Dense layers.

These layers are described in further depth in Table 2.

Table 2: The suggested CNN layers.

NO.	Layer Type	Filters	Size/ stride	Activation function
1	Convolutional	16	3/1	--
2	Leaky ReLU	--	--	--
3	Max Pooling	--	1/1	--
4	Convolutional	32	3/1	--
5	Leaky ReLU	--	--	--
6	Max Pooling	--	1/1	--
7	Convolutional	32	3/1	--
8	Leaky ReLU	--	--	--
9	Max Pooling	--	1/1	--
10	Convolutional	128	3/1	--
11	Leaky ReLU	--	--	--
12	Leaky ReLU	--	--	--
13	Max Pooling	--	1/1	--
14	Dense	128	--	Linear
15	Convolutional	256	3/1	--
16	Leaky ReLU	--	--	--
17	Max Pooling	--	1/1	--
18	Convolutional	512	3/1	--
19	Leaky ReLU	--	--	--
20	Max Pooling	--	1/1	--
21	Convolutional	1024	3/1	--
22	Leaky ReLU	--	--	--
23	Max Pooling	--	1/1	--
24	Dense	1024	--	Linear
25	Convolutional	50	3/1	--
26	Flatten	--	--	--
27	Dense	2	--	Softmax

4 RESULTS AND DISCUSSION

This section displays the outcomes of applying different machine learning algorithms with and without feature selection to the Santander Customer Transaction Prediction dataset. For each scenario, the comparison highlights the most critical performance metrics: accuracy, precision, recall, F1-score, and computational time, which all demonstrate the strength and efficiency of each model. Feature selection methods such as Mutual Information, Chi-square, and ANOVA were combined in order to examine their influence on dimension reduction, classification performance, and runtime optimization. The results are then compared across the different algorithms to determine their advantages, limitations, and trade-offs in terms of prediction accuracy. Moreover, the debate empirically examines the impact of feature selection on generalization, redundancy elimination, and addressing computational complexity. This provides more insight into what models and preprocessing methods are most suited for robust customer transaction forecasting.

4.1 Results of the CNN Model Without FS

The result in Table 3 indicates that the model without FS can achieve perfect prediction accuracy on this dataset. The tradeoff is computational efficiency; all the features require significantly more processing time. In practice, such performance can indicate overfitting or reliance on redundant/noisy features, which is why FS is still beneficial to remove complexity and to ensure the model generalizes well to new data.

Table 3: Results of the CNN model without feature selection.

	prs	rec	f1	acc	Time in seconds
Without FS	100	100	100	100	72

Table 4: Results of the CNN model with MI feature selection.

	prs	rec	f1	acc	Time in seconds
mi10%	100	100	100	100	9s
mi30%	100	100	100	100	27s
mi50%	100	100	100	100	53s

Table 5: Results of the CNN model with Chi-square feature selection.

	prs	rec	f1	acc	Time in seconds
Chi-10%	100	100	100	100	11
Chi-30%	100	100	100	100	30
Chi-50%	100	100	100	100	49

Table 6: Results of the CNN model with ANOVA feature selection.

	prs	rec	f1	acc	Time in seconds
ANOVA-10%	100	100	100	100	6
ANOVA-30%	100	100	100	100	24
ANOVA-50%	100	100	100	100	41

4.2 Results of the CNN Model with MI-FS with 10%, 30% and 50% of Features

The performance of the CNN model with Mutual Information (MI) feature selection on 10%, 30%, and 50% subsets, as shown in Table 4 exhibits consistently perfect performance with 100% precision, recall, F1-score, and accuracy for all configurations. This indicates that the CNN is very good at capturing the significant patterns in the data, even when only a subset of the features is available. For MI-10%, the model spent 9 seconds alone, which was the fastest and most efficient configuration. When the subset was increased to MI-30, the time was pushed to 27 seconds, while MI-50% further extended it to 53 seconds, reflecting the linear computation time vs feature size trade-off. There was no impact on classification performance in spite of this rise in execution time since the model still displayed perfect predictive power. This means that less carefully selected features suffice for CNN to achieve optimum accuracy. Stability in performance across all MI levels also means that there is ample redundancy in features in the dataset. Feature selection thus ensures that redundant attributes are eliminated while only the most informative are retained. Importantly, MI-10 emerges as the optimum configuration since it lowers computation without affecting accuracy. These findings underscore the effectiveness of MI in dimensionality reduction and the robustness of CNN in handling high-dimensional data. Together, the findings show that CNN in combination with MI-based feature selection can provide both computational efficiency and high accuracy.

4.3 Results of the CNN Model with Chi-FS with 10%, 30%, and 50% of Features

The CNN model with Chi-square feature selection at 10%, 30%, and 50% subsets' results are consistently perfect, as shown in Table 5, with 100% precision, recall, F1-score, and accuracy in all configurations. This suggests that, like MI, the CNN is very effective in identifying the hidden patterns of the data even when only a portion of the features is kept. At Chi-10, the model computed these results in 11 seconds, indicating computation efficiency. Increasing the subset to Chi-30 doubled the computation time to 30 seconds, and Chi-50 increased it further to 49 seconds, which indicates that larger sets of features require longer computation. However, the consistency of prediction accuracy at all feature levels confirms that the Chi-square method successfully finds and preserves the most informative features while filtering out superfluous ones. The fact that CNN achieves the same performance with only 10% of the features shows the redundancy of the dataset and the effectiveness of Chi-square in dimensionality reduction. Even though all feature sets produced optimal results, the most optimal choice is Chi-10 because it strikes an appropriate balance between accuracy and computational cost. These findings confirm that CNN, in conjunction with Chi-square feature selection, is capable of attaining both high accuracy and effectiveness.

4.4 Results of the CNN Model with ANOVA-FS with 10%, 30%, and 50% of Features

The accuracy of the CNN model when ANOVA feature selection is applied, as shown in Table 6 proves that the model achieved its best performance in all metrics, precision, recall, F1-score, and accuracy, at 10%, 30%, and 50% subset features. When ANOVA-10 was used, the model achieved not only 100% accuracy but also the shortest computation time of just 6 seconds, which is the most cost-effective setting. As the feature count was increased to ANOVA-30 and ANOVA-50, execution time took 24 seconds and 41 seconds, respectively, without a change in performance metrics. Such robustness against subsets suggests that the CNN model is highly resilient and can maintain predictive ability even with low feature counts. Moreover, ANOVA performed well in choosing the most informative features without impacting performance despite dimensionality reduction. The results show that ANOVA-10 offers the optimal speed-accuracy trade-off and thus is an excellent choice for modeling economically. Overall, these results confirm that ANOVA-based feature selection can greatly enhance computational efficiency without sacrificing classification performance.

4.5 Comparison of CNN with Different FS Techniques

The three feature selection techniques: Mutual Information (MI), Chi-square (CH), and ANOVA, resulted in the same classification performance with perfect precision, recall, F1-score, and accuracy (all 100%) on various feature subsets (10%, 30%, and 50%) (see Fig. 2). The only variation occurs for execution time, which is then the determining criterion for choosing the best technique. At the 10% feature subset, ANOVA (NOVA10) took the least processing time of 6 seconds, which was less than MI10 (9 seconds) and CH10 (11 seconds). Similarly, at 30% and 50% subsets, ANOVA was always faster (24s and 41s) than MI (27s and 53s) and CH (30s and 49s). Since all methods share identical predictive performance, the best solution is ANOVA with 10% features (NOVA10) since it has perfect accuracy with the lowest computation expense and hence is the best option where performance and speed must be traded off.

4.6 Comparison with Related Studies

Table 7 presents a comparison of classification accuracy between previously reported studies and the proposed CNN model. Earlier works such as [8] and [9] achieved relatively high accuracies of 89% and 87%, respectively, while [10] and [11] performed moderately well with 74% and 81%, and [12] showed weaker performance at only 57% accuracy. A more advanced method in [13] achieved an impressive 98.43%, showing significant improvement over earlier approaches. In [14], the standalone CNN produced lower accuracy, but when combined with ANN, it reached 97.62%, indicating that hybrid models can enhance performance. In comparison, the proposed CNN model achieved a perfect 100% accuracy, outperforming all referenced studies. This demonstrates that the proposed method not only improves over traditional and hybrid approaches but also achieves the highest possible classification accuracy, making it the most reliable and robust solution among the compared works.

Table 7: Comparison results with related works.

Ref. No.	Accuracy
[8]	89%
[9]	87%
[10]	74%
[11]	81%
[12]	57%
[13]	98.43%
[14]	CNN: 97.62% ANN: 96%
The proposed CNN model	100%

5 CONCLUSIONS

Previous machine learning models, such as Random Forest, SVM, and Decision Trees, battled with overfitting and feature redundancy despite achieving respectable accuracy levels. We can note that the development of deep learning frameworks such as CNNs has transformed data analysis by making hierarchical pattern detection and automatic feature extraction possible.

In terms of both predicted accuracy and computational efficiency, the suggested CNN-based approach shows a notable improvement over the conventional models that have been in use for the past ten years. This historical development demonstrates how feature selection combined with deep architectures can help close the gap between traditional statistical techniques and contemporary AI-powered solutions.

This study highlighted the effectiveness of the proposed CNN model combined with feature selection techniques in achieving highly accurate and efficient classification results. By applying Mutual Information, Chi-square, and ANOVA with different feature percentages (10%, 30%, 50%), the CNN consistently achieved 100% accuracy, precision, recall, and F1-score, proving its robustness and reliability. While the classification time increased with more features, the performance remained stable, demonstrating the model's efficiency in handling different feature sets. Compared with previous studies, which reported accuracies between 57% and 98.43%, the proposed CNN reached a perfect accuracy level, confirming its superiority. Overall, this research shows that the integration of feature selection with CNN not only ensures maximum accuracy but also improves computational efficiency, offering a powerful solution for future classification tasks.

REFERENCES

[1] J. W. Goodell, S. Kumar, W. M. Lim, and D. Pattnaik, "Artificial intelligence and machine learning in finance: identifying foundations, themes, and research clusters from bibliometric analysis," J. Behav. Exp. Financ., vol. 32, p. 100577, 2021.

[2] I. Vorobyev and A. Krivitskaya, "Reducing false positives in bank anti-fraud systems based on rule induction in distributed tree-based models," Comput. Secur., vol. 120, p. 102786, 2022.

[3] A. I. Canhoto, "Leveraging machine learning in the global fight against money laundering and terrorism financing: an affordances perspective," J. Bus. Res., vol. 131, pp. 441–452, 2021.

[4] M. Bouhia, L. Rajaobelina, S. PromTep, M. Arcand, and L. Ricard, "Drivers of privacy concerns when interacting with a chatbot in a customer service encounter," Int. J. Bank Market., vol. 40, pp. 1159–1181, 2022.

[5] D. Mittal and S. R. Agrawal, "Determining banking service attributes from online reviews: text mining and sentiment analysis," Int. J. Bank Market., vol. 40, pp. 558–577, 2022.

[6] M. M. Alam, A. E. Awawdeh, and A. I. B. Muhamad, "Using e-wallet for business process development: challenges and prospects in Malaysia," Bus. Process Manag. J., vol. 27, pp. 1142–1162, 2021.

[7] G. Bontempi, Reproducible machine learning for credit card fraud detection - practical machine learning for credit card fraud detection - practical handbook foreword, May 2021.

[8] A. Chouiekh and E. Ibn El Haj, "Detecting and classifying customer loss using deep convolutional neural networks," 2020.

[9] M. Nabipour et al., "Comparison of deep learning techniques and machine learning models for stock prediction using technical indicators," 2020.

[10] M. Pondel et al., "Deep learning model for predicting client attrition in e-commerce," 2021.

[11] A. Khattak et al., "Predicting customer turnover using a hybrid BiLSTM-CNN deep learning model," 2023.

[12] C. Ku et al., "Integrating investors' expertise with LSTM for stock price prediction," 2023.

[13] J. Li et al., "CNN-LSTM and attention mechanism-based credit risk prediction model for listed businesses," 2023.

[14] O. Seymen, A. Hiziroglu, O. Dogan, O. Er, and E. Ölmez, "Customer churn prediction using ordinary artificial neural network and convolutional neural network algorithms: A comparative performance assessment," Gazi University Journal of Science, vol. 36, no. 1, 2022.

[15] A. Khattak, Z. Mehak, H. Ahmad, et al., "Customer churn prediction using composite deep learning technique," Sci Rep., vol. 13, no. 17294, 2023.

[16] C. Sen Seah et al., "An effective pre-processing phase for gene expression classification," Indonesian Journal of Electrical Engineering and Computer Science, 2018.

[17] S. Patro and K. Kumar, "Normalization: A Preprocessing Stage," no. March 2015, [Online]. Available: https://doi.org/10.17148/IARJSET.2015.2305.

[18] F. Macedo, R. Valadas, E. Carrasquinha, and M. Oliveira, "Feature selection using Decomposed Mutual Information Maximization," Neurocomputing, vol. 513, pp. 215–232, 2022.

[19] S. Ikram and A. Cherukuri, "Intrusion detection model using the fusion of chi-square feature selection and multi-class SVM," Journal of King Saud University – Computer and Information Sciences, 2016.

[20] N. Elssied, O. Ibrahim, and A. Osman, "A Novel Feature Selection Based on One-Way ANOVA F-Test for E-Mail Spam Classification," Research Journal of Applied Sciences, Engineering, and Technology, vol. 7, no. 3, pp. 625–638, 2014.

[21] S. Kadhim, J. Ko, S. P. Paw, C. T. Yaw, S. Abd Al-latief, and A. Alkhayyat, "An Optimized Machine Learning Models by Metaheuristic Corona Virus Optimization Algorithm for Precise Iris Recognition," 2025, [Online]. Available: https://doi.org/10.54364/AAIML.2025.51194.

[22] X. Zhao, L. Wang, Y. Zhang, et al., "A review of convolutional neural networks in computer vision,"

Artif. Intell. Rev., vol. 57, p. 99, 2024, [Online]. Available: https://doi.org/10.1007/s10462-024-10721-6.

[23] P. Rybacki, J. Niemann, S. Derouiche, S. Chetehouna, I. Boulaares, N. M. Seghir, J. Diatta, and A. Osuch, "Convolutional neural network (CNN) model for the classification of varieties of date palm fruits (Phoenix dactylifera L.)," Sensors, vol. 24, p. 558, 2024, [Online]. Available: https://doi.org/10.3390/s24020558.

[24] M. Mousavi and S. Hosseini, "A deep convolutional neural network approach using medical image classification," BMC Med. Inform. Decis. Mak., vol. 24, p. 239, 2024, [Online]. Available: https://doi.org/10.1186/s12911-024-02646-5.

[25] A. Alem and S. Kumar, "Deep Learning Models Performance Evaluations for Remote Sensed Image Classification," IEEE Access, vol. 10, pp. 111784–111793, 2022.

[26] S. Kadhim, S. P. J. Koh, C. T. Yaw, and S. Abd Al-latief, "Deep Learning Models for Biometric Recognition based on Face, Finger Vein, Fingerprint, and Iris: A Survey," Journal of Smart Internet of Things, pp. 117–157, 2024, [Online]. Available: https://doi.org/10.2478/jsiot-2024-0007.

[27] S. Kadhim, J. Paw, C. T. Yaw, S. Abd Al-latief, and A. Alkhayyat, "Deep Learning for Robust Iris Recognition: Introducing Synchronized Spatiotemporal Linear Discriminant Model-Iris," Advances in Artificial Intelligence and Machine Learning, vol. 5, pp. 3446–3464, 2025, [Online]. Available: https://doi.org/10.54364/AAIML.2025.51197.

[28] J. Khan, Y. Chen, Y. Rehman, and H. Shin, "Performance enhancement techniques for traffic sign recognition using a deep neural network," Multimedia Tools and Applications, vol. 79, pp. 20545–20560, 2020.