# Enhancing Credit Risk Assessment through Transformer-Based Machine Learning Models

Elekanyani Siphuma$^{1[0000-0002-9966-8873]}$ and Terence Van Zyl$^{1[0000-0003-4281-630X]}$

Institute for Intelligent Systems,
University of Johannesburg, Johannesburg, South Africa
elekanyanisiphuma@hotmail.co.za
tvanzyl@uj.ac.za

**Abstract.** This study evaluates the effectiveness of transformer-based deep learning models in improving credit risk assessment for predicting default probabilities among credit card customers. By employing the CNN-SFTransformer and GRU-Transformer models, this research aims to enhance predictive accuracy and robustness compared to traditional machine learning methods. The models were trained and tested on diverse datasets from Taiwan, Germany, and Australia, representing various credit risk scenarios. The experimental setup included rigorous hyperparameter tuning and utilized key evaluation metrics such as ROC AUC, KS statistic, and G-$\tilde{\mu}$ to assess model performance comprehensively. The CNN-SFTransformer model demonstrated superior performance, consistently surpassing baseline models like LSTM, Support Vector Machines (SVM), and Random Forest across all datasets. This performance indicates its enhanced capability in differentiating between defaulters and non-defaulters. The GRU-Transformer model also showed promising results, further validating the effectiveness of transformer architectures in this domain. Statistical significance of the results was confirmed through the McNemar test, ensuring the robustness and reliability of the proposed models. This research introduces a novel approach to credit risk management by providing scalable and adaptable models that improve the precision of default predictions, thereby aiding financial institutions in making more informed lending decisions with greater confidence.

**Keywords:** Credit Scoring · CNN-SFTransformer · GRU-Transformer.

## 1 Introduction

Credit risk, the risk associated with lending to borrowers who may default on their credit obligations, is a significant challenge for lending institutions [1, 2]. Basel regulations specify that customers default if they fail to make payments for 90 consecutive days [3]. Traditionally, statistical models like logistic regression and linear discriminants have been used to model this risk due to their ease of implementation and interpretability [4]. However, these models often fall short when dealing with large, complex, and non-linear datasets [5].

The rapid increase in financial data has opened opportunities to enhance credit risk modelling using machine learning techniques. Machine learning, which builds algorithms trained on data to recognize patterns and support decision-making, can be divided into traditional machine learning and deep learning [6–8]. Traditional models, such as logistic regression, support vector machines, decision trees, and random forests, handle well-structured datasets effectively. In contrast, deep learning models, including recurrent neural networks (RNN), long short-term memory (LSTM), convolutional neural networks (CNN), and transformers, excel with complex, non-linear, and unstructured data [9].

Many researchers have applied traditional machine learning models to credit risk modelling, achieving reasonable prediction accuracy with simple datasets [10–12]. However, these models perform poorly with large, high-dimensional, sequential datasets like those from credit card and P2P lending platforms [13]. To overcome these limitations, scholars have turned to deep learning methods. For instance, Wang et al. [9] used LSTM models to analyze customer behaviour on a P2P lending platform, demonstrating better performance than traditional models. Similarly, Ala'raj et al. [14] applied LSTM to evaluate credit card spending behaviour, achieving higher prediction accuracy. In contrast, Zhang et al. [15] showed that combining textual and hard features with a transformer encoder improved default prediction.

More recently, transformer models have gained attention in the credit risk domain due to their success in natural language processing (NLP) tasks [16]. Wang and Xiao [13] introduced a feature-embedded transformer model for predicting customer defaults, showing that it outperformed LSTM, logistic regression, and gradient-boosted decision trees in terms of ROC-AUC and KS statistics. These findings indicate the potential of transformer models to address some of the limitations inherent in LSTM and traditional approaches, particularly when dealing with high-dimensional and sequential financial data.

The Taiwan credit default crisis underscored the necessity for better lending risk management strategies [17, 18]. As financial data volumes and complexity have grown, researchers have turned to machine learning models to classify borrowers. Yet, traditional approaches still fall short with high-dimensional and sequential datasets [13]. While LSTM has mitigated some of these limitations, it remains susceptible to noise and requires longer training times [13]. With their ability to handle long-term dependencies and perform parallel processing, transformer models provide a promising alternative.

This research aims to model credit card customer behaviour using the CNN-SFTransformer and GRU-Transformer models, compare their effectiveness with LSTM and traditional classifiers, and demonstrate the statistical significance of their performance improvements using the McNemar test.

Given this background, the following research questions guide this study:

- To what extent do the CNN-SFTransformer and GRU-Transformer models outperform LSTM and traditional models in predicting default probability, as measured by AUC, KS statistic, and G-$\tilde{\mu}$?
- Is the performance improvement of the CNN-SFTransformer and GRU-Transformer models over traditional models statistically significant according to the McNemar test?

By addressing these questions, this study aims to contribute to more effective credit risk management strategies. The ultimate goal is to provide lending institutions with enhanced tools for categorizing customers based on their risk profiles, enabling more informed decision-making in the context of credit risk.

## 2 Proposed Architectures for Credit Risk Prediction

This study proposes two architectures for credit risk prediction: the CNN-SFTransformer and the GRU-Transformer. The CNN-SFTransformer, adopted from Wang et al. [19], demonstrated good performance on German and Australian datasets. However, the results presented by Wang et al. [19] cannot be fully relied upon as they did not perform statistical significance tests to validate the performance of their models. This study extends their work by incorporating the McNemar test to ensure the robustness and reliability of the CNN-SFTransformer results.

Wang et al. [19] initially applied the CNN-SFTransformer to the German and Australian datasets. In contrast, this research extends its application to the Taiwan credit card dataset, adapting and fine-tuning the model to suit the unique characteristics of this dataset. By addressing the limitations of the original study and enhancing the model with statistical significance validation, this study aims to provide a more robust and reliable credit risk prediction framework.

In addition to the CNN-SFTransformer, the newly proposed GRU-Transformer model is introduced. This model integrates Gated Recurrent Units (GRUs) with multi-head attention mechanisms to capture sequential and global dependencies in credit risk prediction.

### 2.1 CNN-SFTransformer Architecture

The CNN-SFTransformer integrates Convolutional Neural Network (CNN) and Semantic Feature Transformer (SFTransformer) components in a parallel structure for comprehensive feature extraction, as shown in Figure 1. The model processes the feature data in two parallel networks:

**SFTransformer:** Feature data is fed into the SFTransformer to extract feature information between different feature parameters. Initially, a Gaussian distribution-weighted tokenization module converts the feature data into tokenized semantic features, which are then fed into the SFTransformer to construct semantic queries ($Q$) and keys ($K$). The SFTransformer's output is normalized, capturing global dependencies and interactions among features. The multi-head attention mechanism is defined as:

$$\text{Multi-Head}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O \tag{1}$$

where $Q$ represents the queries derived from the tokenized semantic features, $K$ denotes the keys also derived from the tokenized semantic features, $V$ signifies the values extracted from the tokenized semantic features, and $W^O$ is the learnable weight matrix. The attention for each head ($\text{head}_i$) is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2}$$

where $W_i^Q$, $W_i^K$, and $W_i^V$ are the learnable weight matrices specific to each attention head $\text{head}_i$. The attention function is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

where $d_k$ denotes the dimensionality of the keys $K$.

**CNN:** Feature data is processed by a two-layer 1D CNN. The first convolutional layer contains 16 kernels with size three and stride 1usesocal correlations. The operation of the convolutional layer is expressed as:

$$\text{Conv}(x) = (x * w) + b \tag{4}$$

where $x$ is the input, $w$ is the filter, and $b$ is the bias. The output undergoes ReLU activation:

$$\text{ReLU}(x) = \max(0, x) \tag{5}$$

which is then followed by another convolutional layer and batch normalization. Dropout layers prevent overfitting. The flattened output from the 1D CNN is combined with the SFTransformer output in the Features Fusion layer. The final prediction uses a dense layer with a sigmoid activation function, averaging losses from the CNN and SFTransformer components to guide training.

### 2.2 GRU-Transformer Architecture

The GRU-Transformer combines Gated Recurrent Units (GRU) with Multi-Head Attention blocks to capture sequential and global dependencies, as shown in Figure 2. The architecture includes:

**Input Layer:** Ensures data is correctly structured for subsequent layers.

**Bidirectional GRU Layer:** Contains 64 GRU units capturing information from past and future states. The GRU operations include:

- **Update Gate:** $z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$
- **Candidate Hidden State:** $\tilde{h}_t = \tanh(W_h \cdot (r_t \odot h_{t-1}, x_t))$

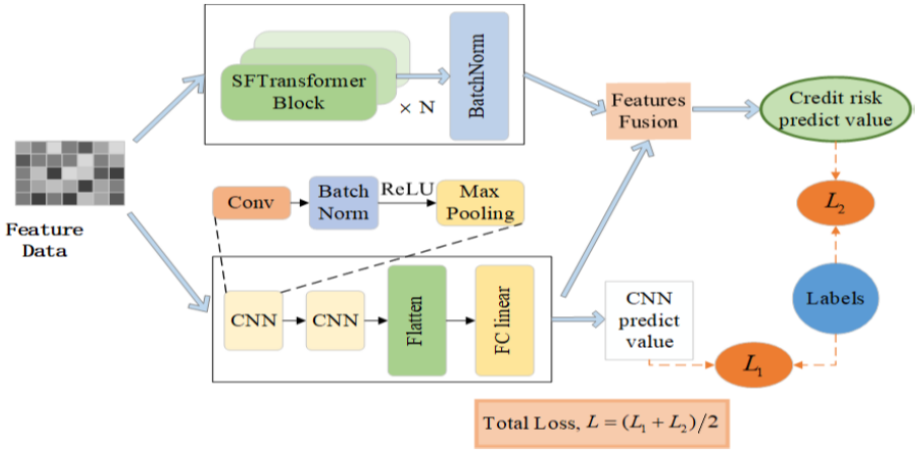

**Fig. 1.** CNN-SFTransformer [19]

- **Hidden State:** $h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$

where $z_t$ is the update gate vector at time step $t$, $\sigma$ is the sigmoid activation function, $W_z$ and $W_h$ are the weight matrices for the update gate and hidden state respectively, $h_{t-1}$ is the hidden state vector from the previous time step $t-1$, $x_t$ is the input vector at the current time step $t$, tanh is the hyperbolic tangent activation function, and $r_t \odot h_{t-1}$ is the element-wise multiplication of the reset gate and the previous hidden state.

**Transformer Blocks:** Include multi-head attention, dropout, add layers (skip connections), and layer normalization.

**Flatten Layer:** Converts the 3D tensor into a 2D tensor, preparing it for the final dense layer.

**Dense Layer:** With a sigmoid activation function, outputs the probability of default.

The GRU-Transformer effectively captures sequential patterns and long-term dependencies, making it suitable for credit risk prediction.

### 2.3 Comparative Analysis and Justification

The CNN-SFTransformer is effective for data with strong local patterns, while the GRU-Transformer is tailored for time-series data with significant temporal patterns. Both architectures were empirically evaluated against baseline models (LSTM, Logistic Regression, Random Forest, SVM), ensuring robust and reliable credit risk predictions.

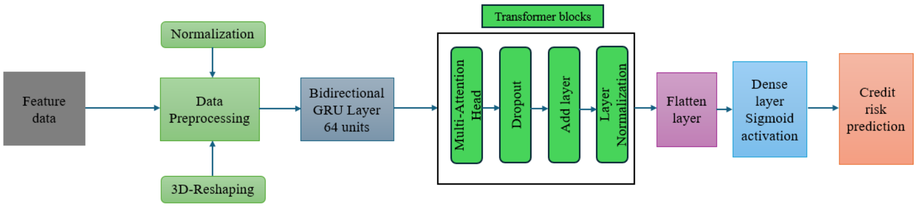

**Fig. 2.** The GRU-Transformer Architecture

## 3 Methods

This study employs a confirmatory experimental design methodology, focusing on the hypothesis-driven evaluation of transformer-based models to enhance credit risk prediction. Confirmatory experimental design emphasizes testing hypotheses through structured experiments [20], ensuring that the proposed models are rigorously evaluated for their effectiveness in improving credit risk assessment.The methodology encompasses dataset descriptions, experimental setup, hyperparameter selection, and evaluation metrics. Additionally, the methodology includes a detailed description of the comparison models used for benchmarking.

### 3.1 Dataset Description

The study utilized three datasets from the UCI Machine Learning Repository: the Taiwan Credit Card dataset, the German Credit dataset, and the Australian Credit Approval dataset. The Taiwan Credit Card dataset contains 30000, the German Credit dataset includes 1000, and the Australian Credit Approval dataset has 690 records. Table 1 shows detailed summaries of these datasets.

**Table 1.** Credit Card Dataset Description

| Dataset | Default | Non-Default | Features |
|---|---|---|---|
| Taiwan Credit Card | 6000 | 24000 | 24 |
| German Credit | 300 | 700 | 21 |
| Australian Credit | 307 | 383 | 15 |

### 3.2 Experimental Setup

The study uses Python 3.10 in Jupyter Notebook with the TensorFlow and Scikit-learn libraries to conduct the experiments. Datasets were split into 80 % training and 20 % testing data to ensure robust model evaluation. Data preprocessing involved transforming temporal features into three-dimensional sequences for the Taiwan dataset using a sliding window of size three with a step size of one, which was crucial for capturing temporal dependencies effectively. The study uses a fixed sliding window of size one for the German and Australian datasets. The Standard-Scaler was used for normalization, ensuring that all features contributed equally to the model's learning process. The proposed models, CNN-SFTransformer and GRU-Transformer, were trained with dropout layers, layer normalization, skip connection and the Adam optimizer with a learning rate 0.001. The study uses binary cross-entropy as the loss function.

### 3.3 Hyperparameter Selection

Due to the computational expense of hyperparameter tuning techniques, the study employed a less computationally expensive approach. This approach involved systematically adjusting three key hyperparameters—number of transformer blocks, attention heads, and batch size—to enhance the performance of both the CNN-SFTransformer and GRU-Transformer models. The ranges for these hyperparameters are summarized in Table 2. To systematically explore the impact of each hyperparameter, one hyperparameter was varied at a time while keeping others constant, with the validation set used to evaluate the effect on model performance. These hyperparameters were selected by values commonly used in the literature.

**Table 2.** Hyperparameters Ranges

| Hyperparameter | Range |
|---|---|
| Transformer blocks | 1-3 |
| Attention heads | 6, 8, 10, 12 |
| Batch size | 32, 64, 128 |

### 3.4 Evaluation Metrics

The study employs ROC-AUC, KS Statistic, and G-$\tilde{\mu}$ to comprehensively assess the models' performance, predictive power and robustness.These metrics were chosen because they are widely used in credit risk modeling due to their ability to handle imbalanced datasets and assess the predictive power of models in distinguishing defaulters from non-defaulters.

**ROC-AUC:** This metric measures the area under the receiver operating characteristic curve, which plots the true positive rate (TPR) against the false positive rate (FPR) across different threshold settings. A higher ROC-AUC indicates better performance in distinguishing between positive and negative classes.

**KS Statistic:** This statistic measures the maximum difference between the cumulative distribution functions of the true and false positive rates. A higher KS value reflects the greater discriminatory power of the model, showing its ability to separate positive and negative cases effectively.

**G-$\tilde{\mu}$:** This metric assesses the model's accuracy in classifying both majority and minority classes by balancing the true positive and negative rates. A higher G-$\tilde{\mu}$ indicates the model performs well in detecting positive cases and avoiding false positives.

**McNemar test:** To assess the statistical significance of the results, the McNemar test was utilized to compare the CNN-SFTransformer model against each baseline algorithm. This test examines discordant pairs of predictions to evaluate performance differences. The McNemar test statistic follows a chi-square distribution with 1 degree of freedom. A threshold value of 3.841 was used to reject the null hypothesis, as it corresponds to the 0.05 significance level in the chi-square distribution. A McNemar test statistic exceeding 3.841 indicates a significant difference in performance between the models, confirming meaningful discrepancies in classification results. The higher the McNemar test statistic value, the greater the difference in performance between the models.

### 3.5 Comparison Models

The machine-learning models used for comparison in this study include Long Short-Term Memory (LSTM) networks, Support Vector Machines (SVM), Random Forests (RF), and Logistic Regression (LR). LSTM networks are recurrent neural networks (RNN) designed for processing sequential data, making them particularly effective for tasks involving temporal dependencies and historical patterns, such as credit risk modelling [21, 22]. SVMs are powerful supervised learning models that find the optimal hyperplane to separate data points of different classes. They enhance generalization to unseen data and make them suitable for binary classification tasks like predicting credit defaults [23, 24]. Random Forest is an ensemble learning method that constructs multiple decision trees and aggregates their predictions. It provides robust performance by addressing overfitting and high variance associated with individual decision trees [25, 26]. Logistic regression is a statistical method for binary classification that models the probability of a given input belonging to a specific class, offering transparent probabilistic estimates of default risk and insights into the significance of different features, making it effective for many fields of data analysis, including credit risk modelling [27, 28].

## 4 Results and Discussion

The results analysis and discussion cover the performance evaluation of the Transformer-based models (CNN-SFTransformer and GRU-Transformer) on the Taiwan, Australian, and German credit datasets. These models are compared against baseline models like LSTM, Logistic Regression, Random Forest, and SVM, using performance metrics including ROC AUC, KS Statistic, and G-$\tilde{\mu}$. The statistical significance of the observed results is validated through the McNemar test.

The performance of both the CNN-SFTransformer and GRU-Transformer models is influenced by several hyperparameters. Detailed experimentation was conducted for both models, focusing on three critical hyperparameters: the number of transformer blocks, attention heads, and batch size. The study uses a systematic approach to identify the optimal configuration for each dataset (Taiwan, German, and Australian). One hyperparameter was varied at a time while keeping the others fixed, allowing for a systematic evaluation of their impact on model performance.

### 4.1 CNN-SFTransformer Hyperparameter Selection

**Transformer Blocks:** The number of transformer blocks varied from 1 to 3 as shown in Table 4 in the appendix. For the Taiwan dataset, the best performance was observed with three transformer blocks, achieving an AUC of **77.46%**, a KS of **42.45%**, and a G-$\tilde{\mu}$ of **59.28%**. One transformer block yielded the highest values for the German and Australian datasets, indicating fewer blocks might be more effective for these datasets.

**Attention Heads:** The number of attention heads varied from 6 to 12, as shown in Table 5 in the appendix. Performance improved consistently by increasing attention heads up until 10, which provided the best results across all datasets. However, increasing the attention heads to 12 led to a slight decline in performance. This decrease may be attributed to overfitting or the increased complexity of the model, which can lead to diminished returns in performance when the attention mechanisms become too dense. The optimal number of attention heads balances capturing sufficient context and maintaining model generalization.

**Batch Size:** Batch size was varied from 32 to 128 as shown in Table 6 in the appendix. The results show that the best performance for the model for the Taiwan dataset was achieved with a batch size of 64. For the German and Australian datasets, a batch size of 32 yielded the highest values. Increasing the batch size beyond 32 for the German and Australian datasets and beyond 64 for the Taiwan dataset did not improve further performance.

The study then used the optimal hyperparameters to compare the performance with baseline algorithms. Overall, the results indicate that the CNN-SFTransformer model, with the best hyperparameters, demonstrates superior performance across different datasets.

### 4.2 GRU-Transformer Hyperparameter Selection

**Transformer Blocks:** The number of transformer blocks varied from 1 to 3, as shown in Table 7 in the appendix. Similar to the observations with the CNN-SFTransformer model, the optimal number of transformer blocks varied across datasets. Three blocks provided the best performance for the Taiwan dataset, capturing more complex patterns effectively. However, one transformer block yielded the best results across all metrics for the German and Australian datasets, indicating that simpler configurations were more effective.

**Attention Heads:** The number of attention heads was varied from 6 to 12 as shown in Table 8 in the appendix. The evaluation showed that the best performance was observed with ten attention heads for all datasets.

**Batch Size:** Batch size was varied from 32 to 128 as shown in Table 9 in the appendix. The batch size of 64 was optimal for the Taiwan dataset, whereas a batch size of 32 was ideal for the German and Australian datasets.

Both the CNN-SFTransformer and GRU-Transformer models were retrained using the identified optimal hyperparameters. Although the CNN-SFTransformer generally outperformed the GRU-Transformer, the final evaluations of both models using the test datasets are detailed in the following subsection. The results provide a comprehensive comparison of their performance with the optimized settings.

### 4.3 Model Analysis

Table 3 presents a detailed analysis of the experimental results obtained from Taiwan, Australian, and German credit datasets. The models' performance is evaluated based on ROC AUC, KS Statistic, and G-$\tilde{\mu}$ metrics, comprehensively comparing their predictive capabilities.

**Taiwan Dataset:** The CNN-SFTransformer achieved the highest ROC-AUC (77.69%), KS (42.60%), and G-$\tilde{\mu}$ (60.09%) as shown in Table 3. These results indicate balanced performance and strong discriminatory power. The GRU-Transformer also demonstrated robust performance, though slightly lower than the CNN-SFTransformer. The LSTM model follows closely, with a ROC-AUC of 77.64%, KS of 42.25% and G-$\tilde{\mu}$ of 58.88%. SVM and Logistic Regression (LR) demonstrated limited effectiveness, with LR being the lowest performer.

**Australian Dataset:** Similar to the observations from the Taiwan dataset, the CNN-SFTransformer outperformed other models with the highest AUC (94.34%), KS (78.92%), and G-$\tilde{\mu}$ (87.77%) as shown in Table 3, demonstrating superior classification and balance. The GRU-Transformer followed closely with an AUC of 93.65%, KS of 78.87%, and G-$\tilde{\mu}$ of 86.50%. The SVM also performed strongly with an AUC of 93.27%, KS of 78.69%, and G-$\tilde{\mu}$ of 86.64%. LSTM, RF, and LR also yielded good results but outperformed the transformer-based models.

**Table 3.** Model Performance on Different Datasets

| | **Taiwan Dataset** | | |
|---|---|---|---|
| **Classifier** | **AUC (%)** | **KS (%)** | **G-$\tilde{\mu}$ (%)** |
| GRU-Transformer | 77.23 | 42.59 | 59.61 |
| CNN-SFT | **77.69** | **42.60** | **60.09** |
| LSTM | 77.64 | 42.25 | 58.88 |
| SVM | 71.86 | 34.83 | 56.57 |
| LR | 70.79 | 36.39 | 48.48 |
| RF | 74.54 | 38.14 | 58.10 |
| | **Australian Dataset** | | |
| **Classifier** | **AUC (%)** | **KS (%)** | **G-$\tilde{\mu}$ (%)** |
| GRU-Transformer | 93.65 | 78.87 | 86.87 |
| CNN-SFT | **94.34** | **78.92** | **87.77** |
| LSTM | 92.57 | 76.58 | 86.69 |
| SVM | 93.27 | 78.69 | 86.64 |
| LR | 92.22 | 75.11 | 86.02 |
| RF | 92.81 | 75.97 | 86.24 |
| | **German Dataset** | | |
| **Classifier** | **AUC (%)** | **KS (%)** | **G-$\tilde{\mu}$ (%)** |
| GRU-Transformer | 80.63 | 55.13 | 69.73 |
| CNN-SFT | **81.25** | **56.24** | **69.98** |
| LSTM | 80.05 | 54.84 | 68.37 |
| SVM | 78.85 | 50.94 | 66.49 |
| LR | 79.92 | 50.51 | 69.61 |
| RF | 79.00 | 49.39 | 65.67 |

**German Dataset:** The CNN-SFTransformer continued to show its robust performance, achieving an AUC of 81.25%, KS of 56.24%, and G-$\tilde{\mu}$ of 69.98% (Table 3), highlighting its superior classification capability and well-balanced performance. It outperformed the GRU-Transformer and LSTM models, with the LSTM achieving an AUC of 80.05%, KS of 54.84%, and G-$\tilde{\mu}$ of 68.37%. The SVM and Logistic Regression (LR) models demonstrated moderate performance, while Random Forest (RF) was the lowest performer.

The CNN-SFTransformer consistently demonstrated superior performance across all three datasets, outperforming state-of-the-art models like LSTM. This performance underscores its ability to effectively utilize attention mechanisms, capturing complex patterns and enhancing model robustness.

### 4.4 Statistical Significance Analysis

To validate whether the performance improvements of the CNN-SFTransformer model over baseline algorithms are statistically significant, the McNemar test was conducted five times for each baseline algorithm across the Taiwan, German, and Australian datasets. A p-value of 0.05 and a critical value of 3.841 were used to assess statistical significance. This analysis ensures that the observed performance improvement of the CNN-SFTransformer was not due to random chance.

**Taiwan Dataset:** Table 10 in the appendix shows the McNemar test results for the Taiwan dataset. The results indicate that the CNN-SFTransformer model consistently outperforms baseline algorithms, with significant differences in most runs. Logistic Regression (LR) showed significant differences in all five runs, with $\chi^2$ values ranging from 30.91 to 44.34 and p-values $< 0.001$. SVM also demonstrated significant differences in four out of five runs. LSTM showed significant differences in all five runs, while Random Forest (RF) exhibited fewer significant differences (three out of five), indicating closer performance to CNN-SFTransformer. GRU-Transformer showed significant differences in two out of five runs.

**German dataset:** Table 11 in the appendix shows that the CNN-SFTransformer model consistently demonstrated robust performance, maintaining statistical significance in several runs against baseline algorithms. The LSTM model showed statistically significant differences in four out of five runs, indicating that CNN-SFTransformer consistently outperforms LSTM. The GRU-Transformer exhibited significant differences in three out of five runs. Random Forest (RF) showed significant differences in all five runs, highlighting that CNN-SFTransformer outperforms RF. Logistic Regression (LR) displayed significant differences in three runs. Support Vector Machine (SVM) showed significant differences in only two runs, suggesting a closer alignment in performance with CNN-SFTransformer in this dataset.

**Australian Dataset:** Table 12 in the appendix shows similar observations to the Taiwan and German datasets. The CNN-SFTransformer model demonstrated robust performance with statistical significance against baseline algorithms in multiple runs. LSTM showed significant differences in all five runs, with extremely high $\chi^2$ values ranging from 97.39 to 122.87 and p-values $< 0.001$. RF also showed significant differences in all runs, with $\chi^2$ values ranging from 42.75 to 43.91 and p-values $< 0.001$. GRU-Transformer showed significant differences in three out of five runs. SVM and LR exhibited significant differences in three runs, further validating the robustness of the CNN-SFTransformer model.

Overall, the McNemar test results confirm the superior performance of the CNN-SFTransformer model across all datasets, with statistically significant improvements over baseline algorithms.

### 4.5 Discussion

The results of this study underscore the CNN-SFTransformer model's effectiveness in predicting credit card default probabilities. Statistical significance testing confirmed the robust performance of the CNN-SFTransformer model, affirming that the observed results are not due to random chance. Figure 3 illustrates the CNN-SFTransformer's performance compared to state-of-the-art approaches reported in the literature for the Taiwan and German credit card datasets.

For the Taiwan dataset, the CNN-SFTransformer achieved an AUC of 77.69%, a KS statistic of 42.66%, and a G-$\tilde{\mu}$ of 60.09%. While Ala'raj et al. [14] reported a slightly higher AUC of 78.00% with their LSTM model, the CNN-SFTransformer exhibited a superior KS statistic, indicating improved discriminatory power. For the German dataset, the CNN-SFTransformer significantly outperformed existing models, achieving an AUC of 81.25%, a KS of 56.34%, and a G-$\tilde{\mu}$ of 69.98%. As shown in Figure 3, these results surpass those reported by Shen et al. [29] for their LSTM model (AUC of 80.32% and KS of 48.40%) and Wu and Shang. [30] with their BPNN model (AUC of 76.10% and KS of 40.71%).

The CNN-SFTransformer's architecture, which integrates Convolutional Neural Networks (CNN) with a Semantic Feature Transformer (SFTransformer), allows for effective local and global feature extraction. This comprehensive approach, capturing both temporal and non-temporal features, enhances the model's ability to understand customer behaviour and improve predictive accuracy.

Implementing the CNN-SFTransformer offers several advantages for lending institutions, including enhanced risk management capabilities and more informed lending decisions, which could reduce default rates and improve credit portfolio quality. The model's interpretability, supported by attention mechanisms, provides valuable insights into credit risk factors, facilitating refined credit scoring policies and targeted customer management strategies. Furthermore, its scalability ensures effective application to large datasets.

Despite these strengths, the study acknowledges limitations such as high computational costs and the requirement for large datasets, which may pose challenges for smaller institutions. Future research should explore additional features and alternative model architectures and test the model's applicability across different financial contexts to further validate its robustness and generalizability.

## 5 Conclusion

This study explored deep learning techniques, focusing on transformer-based models to enhance credit risk management by predicting default probabilities among credit card customers. The CNN-SFTransformer and GRU-Transformer models were developed and validated, with their performance compared to baseline models including LSTM, RF, GB, SVM, and LR. The CNN-SFTransformer demonstrated superior discriminatory power on the Taiwan dataset, achieving an AUC of 77.69%, KS of 42.66%, and G-$\tilde{\mu}$ of 60.09%. Similar superior performance of the CNN-SFTransformer was observed on the German and Australian datasets, confirming its generalizability. The GRU-Transformer also improved over traditional models, indicating its effectiveness in credit risk management. The statistical significance of the CNN-SFTransformer's performance improve-

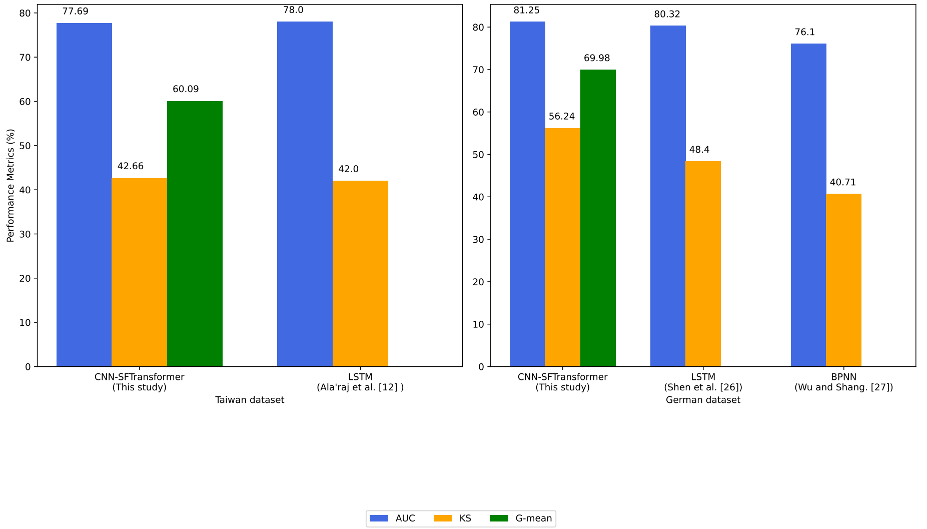

**Fig. 3.** Comparative performance of CNN-SFTransformer and state-of-the-art models

ments was validated using the McNemar test, ensuring these improvements were not due to random chance.

The findings of this study suggest that transformer-based models, especially the CNN-SFTransformer, can significantly enhance credit risk prediction. Financial institutions are encouraged to adopt these advanced models to improve risk management and inform lending decisions. Future research should explore these models across various loan types and economic conditions and optimize hyperparameters for better performance. The scalability and interpretability of these models offer valuable insights into credit risk factors, which can aid in developing targeted credit scoring policies.

Overall, the CNN-SFTransformer and GRU-Transformer models represent significant advancements in credit risk prediction, offering potential benefits for risk management and portfolio quality.

## References

1. Christian Bluhm, Ludger Overbeck, and Christoph Wagner. *Introduction to credit risk modeling*. Chapman and Hall/CRC, 2016.
2. Samuel Hymore Boahene, Julius Dasah, and Samuel Kwaku Agyei. Credit risk and profitability of selected banks in ghana. *Research Journal of finance and accounting*, 3(7):6–14, 2012.
3. Béchir Ben Lahouel, Lotfi Taleb, Younes Ben Zaied, and Shunsuke Managi. Financial stability, liquidity risk and income diversification: evidence from european banks using the camels–dea approach. *Annals of Operations Research*, 334(1):391–422, 2024.

4. Vijay S Desai, Jonathan N Crook, and George A Overstreet Jr. A comparison of neural networks and linear scoring models in the credit union environment. *European journal of operational research*, 95(1):24–37, 1996.
5. Marcos Roberto Machado and Salma Karray. Assessing credit risk of commercial customers using hybrid machine learning algorithms. *Expert Systems with Applications*, 200:116889, 2022.
6. Siddharth Bhatore, Lalit Mohan, and Y Raghu Reddy. Machine learning techniques for credit risk evaluation: a systematic literature review. *Journal of Banking and Financial Technology*, 4(1):111–138, 2020.
7. Thabang Mathonsi and Terence L van Zyl. Multivariate anomaly detection based on prediction intervals constructed using deep learning. *Neural Computing and Applications*, pages 1–15, 2022.
8. Siddeeq Laher, Andrew Paskaramoorthy, and Terence L Van Zyl. Deep learning for financial time series forecast fusion and optimal portfolio rebalancing. In *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2021.
9. Chongren Wang, Dongmei Han, Qigang Liu, and Suyuan Luo. A deep learning approach for credit scoring of peer-to-peer lending using attention mechanism lstm. *Ieee Access*, 7:2161–2168, 2018.
10. Trilok Nath Pandey, Alok Kumar Jagadev, Suman Kumar Mohapatra, and Satchidananda Dehuri. Credit risk analysis using machine learning classifiers. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pages 1850–1854. IEEE, 2017.
11. Artem Bequé and Stefan Lessmann. Extreme learning machines for credit scoring: An empirical evaluation. *Expert Systems with Applications*, 86:42–53, 2017.
12. Si Shi, Rita Tse, Wuman Luo, Stefano D'Addona, and Giovanni Pau. Machine learning-driven credit risk: a systemic review. *Neural Computing and Applications*, 34(17):14327–14339, 2022.
13. Chongren Wang and Zhuoyi Xiao. A deep learning approach for credit scoring using feature embedded transformer. *Applied Sciences*, 12(21):10995, 2022.
14. Maher Ala'raj, Maysam F Abbod, Munir Majdalawieh, and Luay Jum'a. A deep learning model for behavioural credit scoring in banks. *Neural Computing and Applications*, 34(8):5839–5866, 2022.
15. Weiguo Zhang, Chao Wang, Yue Zhang, and Junbo Wang. Credit risk evaluation model with textual features from loan descriptions for p2p lending. *Electronic commerce research and applications*, 42:100989, 2020.
16. Narendra Patwardhan, Stefano Marrone, and Carlo Sansone. Transformers in the real world: A survey on nlp applications. *Information*, 14(4):242, 2023.
17. M. Chou. Cash and credit card crisis in taiwan. *Business Weekly*, page 24–27, 2006.
18. Chih-Hsiung Chang. Information asymmetry and card debt crisis in taiwan. *Bulletin of Applied Economics*, 9(2):123–145, 2022.
19. Mengyuan Wang, Lijian Zhou, Qingyu Meng, Yifan Kong, and Jie Sun. Credit risk prediction network based on semantic feature transformer and cnn. In *2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT)*, pages 723–728. IEEE, 2023.
20. Erlend B Nilsen, Diana E Bowler, and John DC Linnell. Exploratory and confirmatory research in the open science era. *Journal of Applied Ecology*, 57(4):842–847, 2020.
21. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

22. Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.
23. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.
24. Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.
25. Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.
26. Oded Z Maimon and Lior Rokach. *Data mining with decision trees: theory and applications*, volume 81. World scientific, 2014.
27. David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*. John Wiley & Sons, 2013.
28. Scott Menard. *Applied logistic regression analysis*, volume 106. Sage, 2002.
29. Feng Shen, Xingchao Zhao, Gang Kou, and Fawaz E Alsaadi. A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique. *Applied Soft Computing*, 98:106852, 2021.
30. Xinyu Wu, Jielin Shang, et al. Analysis of credit customer delinquency based on bp neural network model. *Financial Engineering and Risk Management*, 7(3):181–186, 2024.

# 6 Appendix

**Table 4.** CNN-SFTransformer: Transformer Blocks Hyperparameter Selection

| | **Taiwan** | | | **German** | | | **Australian** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Blocks** | **AUC** | **KS** | **G-$\tilde{\mu}$** | **AUC** | **KS** | **G-$\tilde{\mu}$** | **AUC** | **KS** | **G-$\tilde{\mu}$** |
| 1 | 77.25 | 42.38 | 59.17 | **81.08** | **55.65** | **69.42** | **93.30** | **77.54** | **87.04** |
| 2 | 77.33 | 42.39 | 59.22 | 80.54 | 55.41 | 69.39 | 93.15 | 77.51 | 86.98 |
| 3 | **77.46** | **42.45** | **59.28** | 80.38 | 55.27 | 69.31 | 92.99 | 77.49 | 86.97 |

**Table 5.** CNN-SFTransformer: Multi-Attention Head Hyperparameter Selection

| | **Taiwan** | | | **German** | | | **Australian** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Heads** | **AUC** | **KS** | **G-$\tilde{\mu}$** | **AUC** | **KS** | **G-$\tilde{\mu}$** | **AUC** | **KS** | **G-$\tilde{\mu}$** |
| 6 | 77.46 | 42.45 | 59.28 | 81.08 | 55.65 | 69.42 | 93.30 | 77.54 | 87.04 |
| 8 | 77.49 | 42.48 | 59.33 | 81.15 | 55.74 | 69.58 | 93.44 | 77.60 | 87.20 |
| 10 | **77.53** | **42.51** | **59.41** | **81.19** | **55.87** | **69.67** | **93.64** | **77.82** | **87.29** |
| 12 | 77.51 | 42.50 | 59.41 | 81.17 | 55.88 | 69.65 | 93.61 | 77.75 | 87.23 |

**Table 6.** CNN-SFTransformer: Batch Size Hyperparameter Selection

| | **Taiwan** | | | **German** | | | **Australian** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Batch** | **AUC** | **KS** | **G-$\tilde{\mu}$** | **AUC** | **KS** | **G-$\tilde{\mu}$** | **AUC** | **KS** | **G-$\tilde{\mu}$** |
| 32 | 77.53 | 42.51 | 59.41 | **81.19** | **55.87** | **69.67** | **93.64** | **77.82** | **87.29** |
| 64 | **77.56** | **42.55** | **59.47** | 81.11 | 55.82 | 69.62 | 93.61 | 77.79 | 87.24 |
| 128 | 77.52 | 42.53 | 59.44 | 81.08 | 55.78 | 69.59 | 93.59 | 77.74 | 87.20 |

**Table 7.** GRU-Transformer: Transformer Blocks Hyperparameter Selection

| **Transformer** | **Taiwan** | | | **German** | | | **Australian** | | |
|---|---|---|---|---|---|---|---|---|---|
| **block** | **AUC** | **KS** | **G-$\tilde{\mu}$** | **AUC** | **KS** | **G-$\tilde{\mu}$** | **AUC** | **KS** | **G-$\tilde{\mu}$** |
| 1 | 76.74 | 41.66 | 58.87 | **79.43** | **54.67** | **69.32** | **93.27** | **77.42** | **85.04** |
| 2 | 76.77 | 41.69 | 58.89 | 79.40 | 54.60 | 69.11 | 93.24 | 77.39 | 84.99 |
| 3 | **76.79** | **41.74** | **58.91** | 79.37 | 54.57 | 69.10 | 93.24 | 78.35 | 84.96 |

**Table 8.** GRU-Transformer: Multi-Attention Head Hyperparameter Selection

| | **Taiwan** | | | **German** | | | **Australian** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Heads** | **AUC** | **KS** | **G-$\tilde{\mu}$** | **AUC** | **KS** | **G-$\tilde{\mu}$** | **AUC** | **KS** | **G-$\tilde{\mu}$** |
| 6 | 76.79 | 41.74 | 58.91 | 79.43 | 54.67 | 69.32 | 93.27 | 77.42 | 85.04 |
| 8 | 76.82 | 41.77 | 58.94 | 79.51 | 54.74 | 69.56 | 93.31 | 77.68 | 85.08 |
| 10 | **76.86** | **41.84** | **58.98** | **79.67** | **54.87** | **69.70** | **93.36** | **77.74** | **85.12** |
| 12 | 76.85 | 41.81 | 58.96 | 79.63 | 54.84 | 69.68 | 93.34 | 77.71 | 85.10 |

**Table 9.** GRU-Transformer: Batch Size Hyperparameter Selection

| | **Taiwan** | | | **German** | | | **Australian** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Batch** | **AUC** | **KS** | **G-$\tilde{\mu}$** | **AUC** | **KS** | **G-$\tilde{\mu}$** | **AUC** | **KS** | **G-$\tilde{\mu}$** |
| 32 | 76.86 | 41.84 | 58.98 | **79.67** | **54.87** | **69.70** | **93.36** | **77.54** | **85.12** |
| 64 | **76.91** | **41.88** | **59.12** | 79.65 | 54.84 | 69.56 | 93.34 | 77.51 | 85.11 |
| 128 | 76.87 | 41.86 | 59.11 | 79.61 | 54.82 | 69.51 | 93.30 | 77.48 | 84.99 |

**Table 10.** Taiwan Dataset: McNemar's test results for multiple classifiers over five runs.

| **Classifier** | $\chi^2$ | **p-value** |
|---|---:|---:|
| **LSTM** | | |
| Run 1 | 13.26 | $2.71 \times 10^{-4}$ |
| Run 2 | 6.09 | $1.36 \times 10^{-2}$ |
| Run 3 | 32.49 | $1.20 \times 10^{-8}$ |
| Run 4 | 4.89 | $2.70 \times 10^{-2}$ |
| Run 5 | 23.65 | $1.16 \times 10^{-6}$ |
| **GRU-Transformer** | | |
| Run 1 | 9.27 | $2.56 \times 10^{-3}$ |
| Run 2 | 8.74 | $3.12 \times 10^{-3}$ |
| Run 3 | 0.47 | $4.92 \times 10^{-1}$ |
| Run 4 | 8.81 | $3.00 \times 10^{-3}$ |
| Run 5 | 0.87 | $3.98 \times 10^{-1}$ |
| **RF** | | |
| Run 1 | 6.57 | $2.65 \times 10^{-2}$ |
| Run 2 | 2.97 | $8.50 \times 10^{-2}$ |
| Run 3 | 4.42 | $3.55 \times 10^{-2}$ |
| Run 4 | 5.08 | $2.42 \times 10^{-2}$ |
| Run 5 | 2.97 | $8.50 \times 10^{-2}$ |
| **SVM** | | |
| Run 1 | 22.96 | $1.65 \times 10^{-6}$ |
| Run 2 | 33.36 | $7.67 \times 10^{-9}$ |
| Run 3 | 19.70 | $9.08 \times 10^{-6}$ |
| Run 4 | 19.35 | $1.09 \times 10^{-5}$ |
| Run 5 | 2.82 | $9.31 \times 10^{-2}$ |
| **LR** | | |
| Run 1 | 32.97 | $9.40 \times 10^{-10}$ |
| Run 2 | 41.37 | $1.00 \times 10^{-10}$ |
| Run 3 | 44.34 | $1.03 \times 10^{-10}$ |
| Run 4 | 36.97 | $1.20 \times 10^{-10}$ |
| Run 5 | 30.91 | $1.00 \times 10^{-8}$ |

**Table 11.** German Dataset: McNemar's test results for multiple classifiers over five runs.

| **Classifier** | $\chi^2$ | **p-value** |
|---|---:|---:|
| **LSTM** | | |
| Run 1 | 4.01 | $4.55 \times 10^{-2}$ |
| Run 2 | 6.02 | $1.42 \times 10^{-2}$ |
| Run 3 | 0.47 | $4.94 \times 10^{-1}$ |
| Run 4 | 4.87 | $2.72 \times 10^{-2}$ |
| Run 5 | 7.56 | $5.98 \times 10^{-3}$ |
| **GRU-Transformer** | | |
| Run 1 | 4.86 | $6.25 \times 10^{-2}$ |
| Run 2 | 4.98 | $2.56 \times 10^{-2}$ |
| Run 3 | 1.09 | $2.95 \times 10^{-1}$ |
| Run 4 | 6.63 | $3.47 \times 10^{-2}$ |
| Run 5 | 2.17 | $1.41 \times 10^{-1}$ |
| **RF** | | |
| Run 1 | 5.02 | $2.88 \times 10^{-3}$ |
| Run 2 | 6.78 | $9.22 \times 10^{-3}$ |
| Run 3 | 5.85 | $3.22 \times 10^{-3}$ |
| Run 4 | 3.91 | $4.81 \times 10^{-2}$ |
| Run 5 | 3.91 | $4.81 \times 10^{-2}$ |
| **SVM** | | |
| Run 1 | 5.45 | $3.25 \times 10^{-2}$ |
| Run 2 | 7.11 | $7.65 \times 10^{-2}$ |
| Run 3 | 0.01 | $9.10 \times 10^{-1}$ |
| Run 4 | 3.24 | $7.18 \times 10^{-2}$ |
| Run 5 | 3.26 | $7.25 \times 10^{-2}$ |
| **LR** | | |
| Run 1 | 4.99 | $2.88 \times 10^{-2}$ |
| Run 2 | 8.27 | $4.04 \times 10^{-3}$ |
| Run 3 | 7.85 | $3.98 \times 10^{-3}$ |
| Run 4 | 2.55 | $1.10 \times 10^{-1}$ |
| Run 5 | 2.58 | $1.11 \times 10^{-1}$ |

**Table 12.** Australian Dataset: McNemar's test results for multiple classifiers over five runs.

| **Classifier** | $\chi^2$ | **p-value** |
|---|---:|---:|
| **LSTM** | | |
| Run 1 | 122.87 | $1.70 \times 10^{-9}$ |
| Run 2 | 98.42 | $1.70 \times 10^{-9}$ |
| Run 3 | 105.19 | $1.70 \times 10^{-9}$ |
| Run 4 | 115.78 | $1.72 \times 10^{-9}$ |
| Run 5 | 97.39 | $1.01 \times 10^{-9}$ |
| **GRU-Transformer** | | |
| Run 1 | 1.64 | $2.00 \times 10^{-1}$ |
| Run 2 | 6.89 | $8.66 \times 10^{-3}$ |
| Run 3 | 3.75 | $5.28 \times 10^{-2}$ |
| Run 4 | 5.14 | $2.33 \times 10^{-2}$ |
| Run 5 | 9.14 | $2.50 \times 10^{-3}$ |
| **RF** | | |
| Run 1 | 43.91 | $3.45 \times 10^{-11}$ |
| Run 2 | 42.75 | $6.22 \times 10^{-11}$ |
| Run 3 | 42.78 | $6.22 \times 10^{-11}$ |
| Run 4 | 43.91 | $3.45 \times 10^{-11}$ |
| Run 5 | 42.98 | $3.45 \times 10^{-11}$ |
| **SVM** | | |
| Run 1 | 1.59 | $2.08 \times 10^{-1}$ |
| Run 2 | 4.24 | $3.58 \times 10^{-2}$ |
| Run 3 | 4.07 | $3.88 \times 10^{-2}$ |
| Run 4 | 1.95 | $1.62 \times 10^{-1}$ |
| Run 5 | 4.04 | $4.12 \times 10^{-2}$ |
| **LR** | | |
| Run 1 | 2.82 | $9.33 \times 10^{-2}$ |
| Run 2 | 4.34 | $3.73 \times 10^{-2}$ |
| Run 3 | 4.88 | $4.72 \times 10^{-2}$ |
| Run 4 | 3.75 | $5.28 \times 10^{-2}$ |
| Run 5 | 4.14 | $3.88 \times 10^{-2}$ |