International Journal of Computing and Digital Systems
2025, VOL. 17, NO. 1, 1–12

http://dx.doi.org/10.12785/ijcds/1571039983

Comparative Analysis of TCN & DeepTCN Models for Indonesian Stock Price Prediction

Felix[1], Evandiaz Fedora[1] and Alexander Agung Santoso Gunawan[2]

[1] *Computer Science Department, BINUS Graduate Program – Master of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia*
[2] *Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia*

Received 17 June 2024, Revised 18 January 2025, Accepted 31 January 2025

Abstract: Accurately forecasting stock prices movements can lead to financial gains, making it a highly sought-after area of study. In recent studies, Temporal Convolutional Network (TCN) has risen in popularity due to its use of dilated convolutions, which are adept at capturing temporal dependencies within time series data. DeepTCN, a variation of TCN designed specifically for probabilistic forecasting, is said to outperform other models in time series forecasting. As far as we know, no extensive research has been conducted to evaluate the performance of DeepTCN compared to TCN. This study conducted a comparative analysis to assess the performance of both TCN and DeepTCN in Indonesian stock price prediction. Both models will be evaluated using Mean Squared Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) scores. The result from this comparative analysis shows that DeepTCN is superior to TCN in predicting stock prices. DeepTCN consistently outperforms TCN, with lower values of MSE, RMSE, and MAPE. This improved performance lies in the parametric approach used in DeepTCN, which allows it to better capture and adapt to fluctuations in stock trends. The findings from this comparative analysis emphasize the need to assess forecast objectives and dataset requirements when choosing between TCN and DeepTCN.

Keywords: : Deep Learning, TCN, DeepTCN, Stock Prediction, Time Series Forecasting, Indonesian Stock Market

1. Introduction

The stock exchange, commonly known as the stock market, functions as a dynamic marketplace where the trading of shares takes place. It has been proven that stock market capitalization holds a crucial role in propelling a global economic development [1]. Shares of publicly traded companies, having undergone the listing process on the stock market, represent tradable ownership units in these companies. Owning company shares serves as evidence of ownership, granting shareholders access to associated benefits and privileges. The movement of stock prices is primarily influenced by the interaction between demand and supply forces, further shaped by the actions of traders engaged in buying and selling shares. Share transactions aim for financial gains, like conventional transactions involving goods and services. Accurately predicting a stock's trajectory can lead to substantial financial gains, making it a highly sought-after area of study. Forecasting the stock market presents a challenge due to the market's susceptibility to national policies, global and regional economic factors, as well as psychological, human, and an excessive focus on univariate data [2].

In the era of artificial intelligence, machine learning has become crucial for time series forecasting. Deep learning algorithms are well-regarded for their advancements in stock price prediction [3]. Since the trends of the stock market are constantly changing, the amount of data generated in the stock market is huge and has significant nonlinearity. To effectively handle such dynamic data, a model that can identify hidden patterns and provide reliable, scalable forecasting solutions is needed [4].

Recently, deep learning models have been increasingly used as their performance surpasses statistical and traditional models. The nonlinear dynamics within deep learning enables a comprehensive understanding of the complex patterns and temporal dependencies in the stock market [5]. Among these models, a specialized Convolutional Neural Network (CNN) architecture, namely Temporal Convolutional Network (TCN), has gained popularity due to its dilated convolutions, which can effectively capture long-range dependencies in time series, making them well-suited for modeling temporal

E-mail address: felix017@binus.ac.id, evandiaz.fedora@binus.ac.id, aagung@binus.edu

relationships in sequential data. In addition to TCN, a variant known as Deep Temporal Convolutional Network (DeepTCN) has been developed by [6 developer] to forecast numerous correlated time series data through an encoder-decoder architecture. DeepTCN builds upon TCN foundation by introducing additional elements like residual blocks to tackle more challenging task of probabilistic forecasting, which stated by [6] that their model demonstrates a better result when compared to state-of-the-art models in both forecasting and probabilistic forecasting tasks. However, to the best of our knowledge, the capabilities of the DeepTCN model have not been thoroughly evaluated in comparison to its baseline TCN model.

To address the issue, this study conducted a comparative analysis between TCN and DeepTCN to evaluate their performance in the sector of forecasting Indonesian stock prices. Utilizing DeepTCN's strength in probabilistic forecasting, this study also presents a novel method for Indonesian stock price prediction by integrating probabilistic forecasting framework with parametric approach to forecast time series data. This method leverages the capability of DeepTCN to deliver not only point predictions but also a measure of uncertainty, allowing for more informed decision-making in stock market investments. Various evaluation metrics, including Mean Squared Error (MSE), Root Mean Squared Error (MSE), and Mean Absolute Percentage Error (MAPE) were employed to evaluate the performance of these models. The key contributions of this study are as follows:

- Providing a comparative analysis of TCN and DeepTCN for Indonesian stock price prediction. This would involve training and testing the models on Indonesian stock data and assessing their effectiveness in forecasting future prices.
- Identifying strengths and weaknesses of TCN and DeepTCN models to analyze which model performs better for Indonesian stock price prediction. This study delves into factors that influence the performance of TCN and DeepTCN, potentially attributing it to factors like the data's characteristics or the models' suitability for the financial sector.
- Given DeepTCN's specialization in probabilistic forecasting, this study explores its ability to predict not just a single future price but also the probability distribution of possible prices. By integrating probabilistic forecasting with parametric approach into Indonesian stock price prediction, this study also explores new possibilities for enhancing forecasting accuracy and managing risks in financial contexts. This could provide valuable insights into potential risks and uncertainties associated with stock price movements in the Indonesian stock market.

The outcomes of this study not only contribute to the advancement of stock price prediction by offering a robust comparison between TCN and DeepTCN in the context of Indonesian stock market, but also illuminate the strengths and limitations of each method in this specific application. The results serve as valuable guidance for financial analysts and researchers in selecting the most appropriate model for similar tasks in the realm of stock price forecasting, promoting more accurate and reliable predictive models.

This study also introduces a novel method to Indonesian stock price prediction by integrating probabilistic forecasting with a parametric approach. This method has the potential to improve prediction reliability while offering a better way to manage uncertainty, leading to a deeper understanding of stock market trends. By uniting these techniques, the research aims to inspire new ways to address the inherent volatility in financial markets, while fostering innovative approaches to investment planning and risk management. Overall, the insights presented in this paper contribute to the growing field of AI in financial analysis, with potential implications for improving market insights, risk assessment, and investment strategies in Indonesian stock market.

The paper is structured as follows. Section 2 provides a comprehensive review of related research, focusing on the methodologies used in previous studies for stock prediction. This section outlines the various approaches taken by researchers, detailing how they have addressed stock forecasting, and it summarizes the outcomes and insights derived from these studies. It includes a discussion of the evolution of predictive models, leading up to TCN, which have shown promise in handling time series data for stock prediction. Section 3 describes the proposed methodology, outlining the research process with a flowchart, providing information on the dataset and pre-processing steps, and examining the architecture of TCN and DeepTCN in detail. Section 4 presents a detailed description of our experiments and tasks, providing a comprehensive overview of the approaches we use to adjust parameters and examine the model's characteristics, while Section 5 showcases the experimental results and evaluates them using specific metrics. Finally, Section 6 summarizes the conclusions, addresses any limitations, and proposes directions for future research.

2. Literature Review

Numerous research has been conducted to forecast the stock market for financial gains, employing a variety of techniques with diverse outcomes. Historically, conventional statistical techniques that involve different types of moving averages and simple forecasting strategies were often employed to predict stock prices [7]. However, Statistical methods are inherently linear, which restricts their effectiveness in predicting stock prices [8]. Since stock data is nonstationary, chaotic, random, and influenced by various technical factors, traditional statistical methods lack the accuracy needed for reliable forecasts [9].

As machine learning evolves, deep learning is becoming more prominent in stock price prediction, providing advanced models that can identify complex patterns in financial data, which traditional machine learning models often struggle with. These deep learning models are increasingly being used to enhance the accuracy and reliability of stock forecasts. Singh and Srivastava [10] introduced a market forecasting model that utilizes principal component analysis for feature extraction, which serves as an input for a Deep Neural Network (DNN). Althelaya assesses the capabilities of various LSTM-based deep learning architectures in predicting financial time series, focusing on both short-term and long-term forecasts. The study compares bidirectional and stacked LSTM models with simpler neural networks and standard LSTM setups to evaluate their relative performance [11]. In a separate study, Recurrent Neural Network (RNN) with attention mechanisms was introduced by [12] to predict multivariate time series, demonstrating its effectiveness in stock price forecasting.

Besides RNN, Convolutional Neural Network (CNN) are also recognized as powerful models for forecasting stock prices. Although CNN was originally designed for computer vision tasks, they can be applied to time series data due to their capability to capture relevant patterns and features in sequential data. CNN were utilized to predict stock prices using historical data and found that the convolutional sliding window technique can effectively capture stock movements [13]. As the study progresses, it has been observed that traditional RNN are prone to gradient explosion, while CNN are often deemed unsuitable for time series analysis [14].

In more recent studies, a form of specialized CNN architecture, namely temporal convolutional networks (TCN), has gained popularity due to its dilated convolutions, which can effectively capture long-range dependencies in time series, making them well-suited for modeling temporal relationships in sequential data. Deng et al. [15] conducted a study on stock trend prediction using TCN model. They discovered that TCN outperforms ARIMA and deep neural networks such as LSTM and CNN in sequence modeling and classification tasks. Another study [14] introduces a feature attention mechanism into the feature extraction process of TCN. Their results show that this approach enhances TCN performance across various error metrics, suggesting that the features processed by TCN with attention lead to improved predictive outcomes. In another experiment, Zhang and Wang demonstrated that combining wavelet transform with TCN outperformed LSTM [16]. This improvement is attributed to TCN's distinctive convolutional network design, which excels at capturing and analyzing trends in time series data. By combining TCN and BERT, the model was able to capture the contextual nuances of financial news more effectively than traditional methods relying on sentiment scores. Zhang et al. propose that the TCN-at-BERT model offers a more detailed and accurate framework for stock market prediction, showing performance improvements of 5.8% to 19.9% over the LSTM-BERT model [17].

New models are regularly developed to improve the accuracy and flexibility of TCN in the field of time series forecasting [6], [18], [19]. DeepTCN is one of these innovations, designed specifically for probabilistic forecasting [6]. Probabilistic forecasting provides a distribution of possible future outcomes in specific time period that allows for a more detailed understanding of risks and opportunities associated with future events [20]. Baba observed that probabilistic forecasting can improve the prediction accuracy by offering probability distributions within the state space, enabling a more thorough fit to the data than the single curve used in conventional statistical methods [21]. Due to the challenge of predicting uncertainty, probabilistic models often struggle to generate precise probability distributions, leading to either overconfidence in predictions [22] or overly broad uncertainty ranges [23]. Jensen, Bianchi, and Anfinsen [24] introduced an innovative approach to probabilistic time series forecasting using DeepTCN and LSTM as their deep learning models. Their method combines Conformal Prediction (CP) to generate Prediction Intervals (PIs) with reliable coverage and ensemble learners that use Quantile Regression (QR) to address heteroscedastic data. However, a notable limitation of their approach is its reliance on computationally intensive ensemble methods, which may hinder scalability and efficiency in real-time applications. Another study addressing uncertainty issues within the framework of volatility modeling using Efficient Market Hypothesis (EMH) [25]. Although the findings offer valuable insights, the study's assumption of EMH being valid across all market conditions may restrict the model's applicability in markets influenced by behavioral biases or other inefficiencies.

To our knowledge, no comparative study has assessed the effectiveness of DeepTCN against TCN. This gap in research creates an opportunity to investigate the potential strengths and limitations of both models, offering insights into which might be more suitable for specific applications or datasets. Moreover, while probabilistic forecasting holds considerable promise, its broader applicability is hindered by practical challenges, including computational complexity and the need for robust assumptions. By addressing these research gaps, this research not only emphasizes the practical applications of DeepTCN and TCN models but also lays the groundwork for future studies in probabilistic forecasting and financial modeling.

3. Methodology

In this section, we provide a comprehensive overview of the research methodology employed in this study. The model used in this study will also be described in detail, providing a structured way to analyze the collected data. Fig. 1 illustrates the research workflow, outlining the sequential steps taken from data collection to model evaluation.

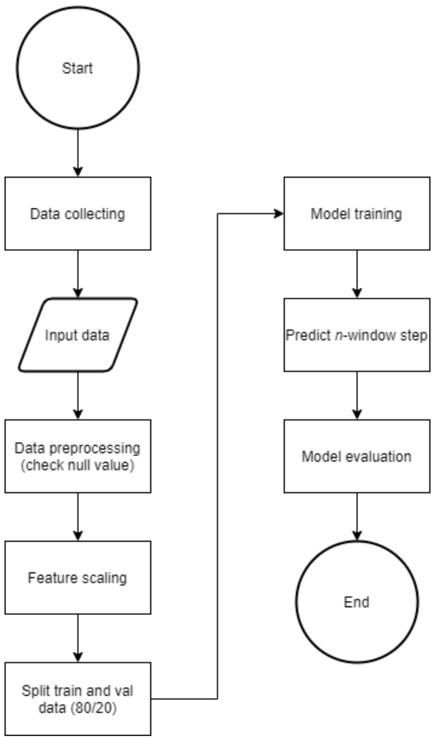


Figure 1. Main research framework

A. TCN

Temporal Convolutional Network (TCN) is a category of convolutional neural network (CNN) uniquely designed to effectively manage time series data [26]. Originally proposed for action segmentation and detection [27], TCN consists of a series of cascaded 1D convolutional layers, enabling the mapping of inputs of varying lengths to output sequences of same length [28]. The network structure of a TCN expands upon that of a 1D CNN, where multiple layers of 1D convolutions are layered consecutively. The fundamentals of 1D convolution layer are depicted in (1) [29].

$$F(x_t) = (x * f)(t) = \sum_{j=0}^{k-1} f_J^T x_{t-j}, t \geq k$$
$$u = (F(x_k), F(x_(k+1)), \ldots, F(x_n)) \quad (1)$$

Both TCN and DeepTCN utilize encoder-decoder architecture to process time series data as shown in Fig. 2. The encoder extracts features, and the decoder uses those features to generate predictions. Both models use causal convolutions to ensure that they only rely on previous information when making predictions, which aligns with the causality principles in time series data [6]. Dilated convolutions might be employed to expand the receptive field and capture long-range dependencies within the sequence.

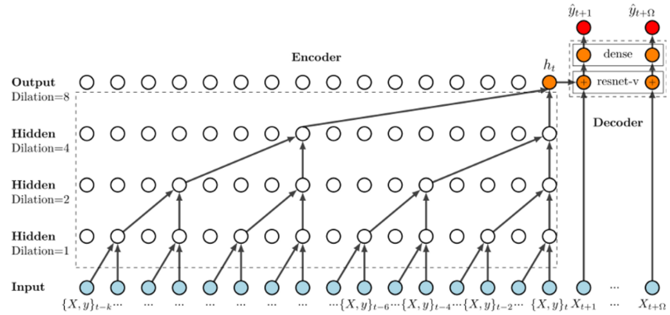


Figure 2. TCN Architecture [?]

Algorithm 1 outlines the TCN algorithm for stock price prediction, offering a step-by-step guide on the architecture and key components. It covers the specific configurations employed to handle time-series data

Algorithm 1 TCN

INPUT
data
arch
OUTPUT
p(val)
eval
check_null(data)
scaler(data)
data(train), data(val)
Model ← *build_model(arch)*
Model ← *train(data)*
p(val) ← predict(Model, window_step, data(val))
MSE, RMSE, MAPE ← (data(val), p(val))
Return MSE, RMSE, MAPE

Unlike conventional CNNs, TCN employs causal and dilated convolutions. In causal convolutions, the output at a given time point t is convolved solely with elements from time t and earlier in the preceding layer. This ensures that there is no information leakage from future time points to past ones [30]. A causal convolutional layer addresses the issue by appending zero padding of length k-1 at the start of the input sequence which is represented in (2) [29].

$$F(x_t) = (x * f)(t) = \sum_{j=0}^{k-1} f_J^T x_{t-j} \qquad x_{\leq 0} := 0$$
$$u = (F(x_1), F(x_2)), \ldots, F(x_n)) \quad (2)$$

An additional concern with the basic 1D CNN is its receptive field, which scales linearly with the number of layers. This is undesirable for our purpose as we intend to capture long-term dependencies. To address this, dilated convolution is employed as a technique that enables receptive fields to grow exponentially with the number of layers. More precisely, when integrated with causal convolution, the dilated convolutional layer at the r-th level can be expressed using (3) [29].

$$F(x_t) = (x *_{l_r} f)(t) = \sum_{j=0}^{k-1} f_J^T x_{t-l_r \cdot j} \qquad x_{\leq 0} := 0$$
$$u = (F(x_1), F(x_2)), \ldots, F(x_n)) \tag{3}$$

B. DeepTCN

Deep Temporal Convolutional Network (DeepTCN) is a forecasting model that builds upon the TCN architecture, but the key difference lies in its use of stacked residual blocks as shown in Fig. 3. Residual blocks allow the network to learn even more intricate temporal relationships by creating a "shortcut" path for the information to flow [6]. These blocks help address the vanishing gradient problem, a common challenge in training deep neural networks on long sequences. The architecture also includes two distinct probabilistic forecasting frameworks. The first framework utilizes a parametric approach, enabling the generation of probabilistic forecasts for future observations by directly predicting the parameters of the hypothetical distribution through maximum likelihood estimation. The second framework, on the other hand, is nonparametric and creates a set of forecasts based on specific quantile values of interest.

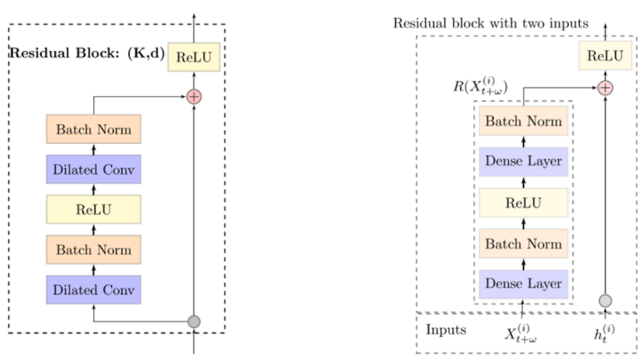


Figure 3. Stacked Residual Blocks
[6]

Algorithm 2 provides an overview of the DeepTCN method for forecasting stock prices, maintaining the same input and output structure as TCN algorithm. The primary variation lies in line 10, where the model incorporates a parametric approach into the algorithm.

Algorithm 2 DeepTCN

1 **INPUT**
2 data
3 arch
4 **OUTPUT**
5 p(val)
6 eval
7 *check_null(data)*
8 *scaler(data)*
9 *data(train), data(val)*
10 Model ← *build_model(arch,* **parameteric_approach***)*
11 Model ← *train(data)*
12 p(val) ← predict(Model, window_step, data(val))
13 MSE, RMSE, MAPE ← (data(val), p(val))
14 Return MSE, RMSE, MAPE

The parametric approach assumes that the data follows a specific distribution defined by parameters such as the mean and standard deviation. In this study, the half-normal distribution is used as the parametric approach. The half-normal distribution is a variant of both the folded normal and truncated normal distributions [31]. Unlike a symmetric normal distribution, the half-normal distribution extends from zero to positive infinity, representing only positive values. It can be visualized as a standard normal distribution folded at its mean [32], resulting in a distribution where all negative values are eliminated. This creates a shape resembling the right half of a typical normal distribution, characteristic of the half-normal distribution. The probability density function of half-normal distribution is defined in (4).

$$y = f(x|\mu, \sigma) = \sqrt{\frac{2}{\pi}} \frac{1}{\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}; x \geq \mu \tag{4}$$

where μ defines the location parameter and σ defines the scale parameter. The probability density function (pdf) is undefined if $x \leq \mu$.

C. Evaluation Metrics

This study uses mostly error assessment metric to see how good the model is to predict stock prices. The metrics that are used in this study are as follows.

1) *Mean Squared Error* (MSE): The Mean Squared Error quantifies the average squared deviation between the actual values and the predicted values [33]. It assigns equal importance to both large and small errors, which makes it particularly sensitive to outliers. MSE is calculated as:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (A_i - P_i)^2 \tag{5}$$

where n represents the number of data points, A_i is the actual value, and P_i is the predicted value.

2) *Root Mean Squared Error* (RMSE): The Root Mean-Squared Error (RMSE) is calculated by taking the square root of the MSE and serves‘ as an indicator for typical size of errors in the original data, it prioritizes significant errors over minor ones. In contrast to MSE, RMSE offers an error metric that uses the same units as the target variable [34]. RMSE is calculated as:

$$RMSE = \sqrt{MSE} \tag{6}$$

3) *Mean Average Percentage Error* (MAPE): The Mean Average Percentage Error calculates the average percentage difference between the actual and predicted values [35]. MAPE is expressed as a percentage, making it easier to interpret. MAPE is calculated as:

$$MAPE = \frac{\sum \frac{|A-P|}{A} \times 100}{N} \tag{7}$$

4. Experimental Procedure

The experiments were conducted on a high-performance workstation featuring an AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx CPU @ 2.1GHz, 20GB of RAM, and an AMD Radeon (TM) Vega 8 GPU. This hardware configuration provided the computational power required for training and evaluating machine learning models efficiently. The machine learning models were implemented using Python programming language (version 3.9.7). We utilized popular libraries such as NumPy, Pandas, and Scikit-learn for data preprocessing, and model evaluation. As for the model used (TCN), the library used is Darts. All experiments were conducted using Python within a Jupyter Notebook environment.

A. Datasets

The dataset used in this study includes historical stock data from the Kaggle [36] , covering the period from April 16, 2001, to January 6, 2023, with intervals ranging from minutes to daily. The characteristics and origins of the stock data are summarized in Table I below, offering a strong foundation for analyzing stock price trends and movements in the following sections.

B. Preprocessing

This study utilized historical stock data from the Indonesian stock market, consisting of 1,570 stock records. The dataset was split into training (1,256 records) and validation sets (314 records) with an 80:20 ratio. When it comes to stock price prediction, cross-validation is not performed because historical data from previous records is required to observe the patterns in a sequential manner. MinMax scaling was applied to normalize features such as opening price, closing price, high price, low price, and volume, ensuring all feature values were scaled between 0 and 1. Feature selection was organized into two approaches: univariate predictions, which focused on predicting the closing price, and multivariate predictions, which aimed to predict both the opening and high prices at the same time.

C. Hyperparameters

In this study, the hyperparameters are derived from the Darts library’s implementation of a TCN model. Both TCN and DeepTCN models share the same basic structural configuration. The parameters for both models are as follows:

1) *Batch_size* : processes input data sequences at a time during training.
2) *Epoch* : the amount of loop that the model will be trained.
3) *Input_chunk_length* : this parameter determines how much historical data the model considers at each step.
4) *Output_chunk_length* : this parameter determines the length of the output sequences it generates.
5) *Dropout* : Regularization technique used to prevent overfitting.
6) *Kernel_size* : this parameter defines the size of the filter that moves across the input data during convolution.
7) *Num_filters* : filters are the building blocks of convolutional neural networks and are responsible for detecting features in the input data.
8) *Optimizers* : used for optimizing the model’s parameters during training.

Despite the structural similarities, DeepTCN in this study adopts a parametric approach using the half-normal distribution. The half-normal distribution is a continuous probability distribution that includes only positive values (where $x \geq 0$). It is derived by taking the absolute values of a standard normal distribution’s random variable, removing any negative values, which results in a distribution focused solely on positive outcomes.

During the training process, we incorporate a parameter known as past covariate. Past covariates provide historical context for the time series data being analyzed. They serve as features that describe past conditions or events that may influence the target variable [36]. In this study, the volume of the stocks will be used as the past covariate.

The TCN model is designed with a batch size of 32 to strike a balance between computational efficiency and the stability of gradient updates. It is trained in over 50 epochs, offering enough iterations for the model to learn effectively while minimizing the risk of overfitting. An input chunk length of 300 allows the model to identify medium to long-term patterns in time-series data, such as cycles and trends, while an output chunk length of 30 targets short-term predictions, which are particularly relevant for stock market analysis. To reduce the likelihood of overfitting, a dropout rate of 0.2 is applied, randomly deactivating 20% of the model’s units during training. The kernel size of 3 is selected to capture localized temporal patterns, and 4 filters are used to maintain a balance between extracting key features and keeping the model architecture simple. Lastly, the Adam optimizer is chosen

TABLE I. Dataset Used

No	Ticker	Stock Name	Sector	Features	No Data	Source
1	INDF	PT Indofood Indonesia Tbk.	Consumer Goods	5	1564	Kaggle [36]
2	ACES	PT Ace Hardware Indonesia Tbk.	Cyclical			
3	BBNI	PT Bank Negara Indonesia Tbk.	Finance			

TABLE II. Hyperparameter Settings

Method	Parameter
TCN	batch_size = 32, epoch = 50, input_chunk_length = 300, output_chunk_length = 30, dropout = 0.2, kernel_size = 3, num_filters = 4, optimize = Adam
DeepTCN	batch_size = 32, epoch = 50, input_chunk_length = 300, output_chunk_length = 30, dropout = 0.2, kernel_size = 3, num_filters = 4, optimize = Adam, likelihood = HalfNormalDistribution()

TABLE III. Metrics for univariate predictions

Method	Ticker	Period(Days)	MSE	RMSE	MAPE
Naive	INDF	1	6675.3193	81.7026	0.9078
	ACES		561.3642	23.6931	1.8991
	BBNI		19394.969	139.2658	1.1697
TCN	INDF	1	0.0063	0.0797	0.1352
	ACES		0.0055	0.0745	0.1094
	BBNI		0.0764	0.2764	0.5683
	INDF	5	0.0045	0.0671	0.1014
	ACES		0.0131	0.1152	0.1643
	BBNI		0.0662	0.2573	0.5308
	INDF	20	0.0025	0.0782	0.0503
	ACES		0.0311	0.1766	0.2522
	BBNI		0.0282	0.1681	0.1191
	INDF	30	0.0018	0.0425	0.0613
	ACES		0.0285	0.1691	0.2421
	BBNI		0.0254	0.1591	0.6338
DeepTCN	INDF	1	0.0011	0.0334	0.0703
	ACES		0.0003	0.0015	0.0021
	BBNI		0.0185	0.1361	0.3889
	INDF	5	0.0035	0.0596	0.0905
	ACES		0.0089	0.0948	0.1126
	BBNI		0.0267	0.1635	0.2573
	INDF	20	0.0014	0.0119	0.0209
	ACES		0.0099	0.0997	0.1151
	BBNI		0.0141	0.1191	0.3041
	INDF	30	0.0017	0.0135	0.0239
	ACES		0.0103	0.1017	0.1064
	BBNI		0.0193	0.1389	0.2272

TABLE IV. Metrics for multivariate predictions

Method	Ticker	Period(Days)	MSE	RMSE	MAPE
TCN	INDF	1	0.0097	0.0986	0.1622
	ACES		0.0057	0.0756	0.1113
	BBNI		0.1093	0.3307	0.6073
	INDF	5	0.0292	0.1711	0.2654
	ACES		0.0173	0.1315	0.1925
	BBNI		0.0922	0.3036	0.5711
	INDF	20	0.0198	0.1409	0.2638
	ACES		0.0312	0.1766	0.2547
	BBNI		0.0441	0.2101	0.3726
	INDF	30	0,0197	0,1406	0,2975
	ACES		0,0288	0,1699	0,2451
	BBNI		0,0314	0,1773	0,2931
DeepTCN	INDF	1	0.0003	0.0173	0.0352
	ACES		0.0001	0.0121	0.0158
	BBNI		0.0051	0.0715	0.2507
	INDF	5	0.0063	0.0796	0.1182
	ACES		0.0081	0.0901	0.1024
	BBNI		0.0427	0.2067	0.3769
	INDF	20	0.0047	0.0689	0.1224
	ACES		0.0119	0.1091	0.1233
	BBNI		0.0314	0.1773	0.2931
	INDF	30	0.0033	0.0578	0.0944
	ACES		0.0178	0.1334	0.1541
	BBNI		0.0207	0.4298	0.1441

for its ability to adapt learning rates dynamically and its effectiveness in handling large datasets, resulting in faster and more reliable convergence.

DeepTCN will be provided with the same hyperparameter as TCN. However, DeepTCN incorporates an additional parameter, likelihood = HalfNormalDistribution(), to enable probabilistic modeling. This likelihood function is well-suited for stock price predictions, as it captures the uncertainty and skewness often observed in such data. By integrating this probabilistic approach, DeepTCN enhances its ability to provide robust predictions even in the presence of highly volatile market conditions, making it more adaptable to real-world scenarios.

5. Result And Discussion

In this study, we evaluate both models into two categories for each stock: univariate prediction and multivariate prediction. Univariate evaluation examines single variables independently, whereas multivariate evaluation explores the connections between several variables to offer a more profound understanding of the phenomenon studied.

The evaluation metrics for univariate and multivariate predictions are provided in Table V and Table VI, respectively. These tables display error metrics (MSE, RMSE, and MAPE) for predictions made across various companies (INDF, ACES, and BBNI) and over different window steps (1, 5, 20, and 30 days).

A. *Performance Evaluation - Univariate Prediction*

In univariate prediction, we established a naive model as a baseline to provide a point of comparison for TCN and DeepTCN in 1 day prediction. The naive model, often representing a simple yet effective prediction method, helps to contextualize the improvements brought by more complex architectures. By comparing the evaluation metrics of TCN and DeepTCN against the naive model, we can assess the capability of these advanced models.

The results in Table III reveal that DeepTCN consistently outperforms TCN across all window steps in terms of MSE, MAPE, and RMSE, as evidenced by the data. For a 1-day window step, DeepTCN reduces MSE by 18%, 75.2%, and 82.54% for different companies, while for a 5-day window, the reduction in MSE ranges from 22.3% to 59.3%, demonstrating its effectiveness over short-term predictions. In longer window steps, such as 20 and 30 days, the reductions are even more significant, with MSE dropping by up to 68% in some instances, reflecting the model's ability to capture long-term stock price fluctuations. Similarly, improvements in MAPE and RMSE are also notable, with DeepTCN consistently delivering better performance across all metrics. For example, in specific cases, MAPE sees reductions of up to 98%, and RMSE decreases by as much as 76.32%, further illustrating the superiority of DeepTCN in handling both short- and long-term stock movements.

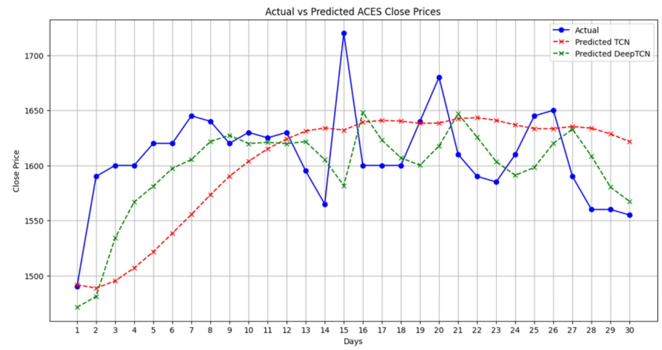
B. Performance Evaluation - Multivariate Prediction

In this study, the results in Table IV reveal that DeepTCN consistently outperforms TCN across all window steps in terms of MSE, MAPE, and RMSE, as evidenced by the data. For a 1-day window step, DeepTCN reduces MSE by 18%, 75.2%, and 82.54% for different companies, while for a 5-day window, the reduction in MSE ranges from 22.3% to 59.3%, demonstrating its effectiveness over short-term predictions. In longer window steps, such as 20 and 30 days, the reductions are even more significant, with MSE dropping by up to 68% in some instances, reflecting the model's ability to capture long-term stock price fluctuations. Similarly, improvements in MAPE and RMSE are also notable, with DeepTCN consistently delivering better performance across all metrics. For example, in specific cases, MAPE sees reductions of up to 98%, and RMSE decreases by as much as 76.32%, further illustrating the superiority of DeepTCN in handling both short- and long-term stock movements.

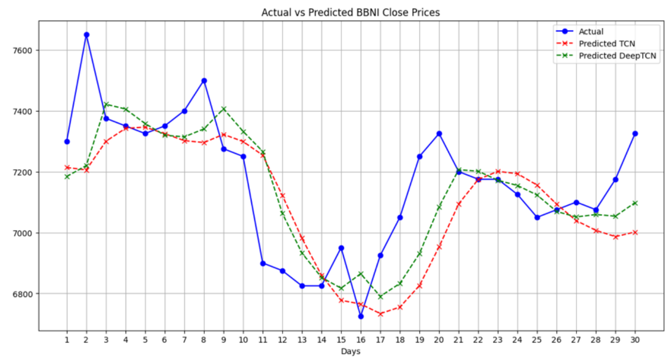
C. Forecast Analysis

In this study, we found that using a parametric approach in DeepTCN significantly improved prediction outcomes, particularly for stock prices. The DeepTCN model with this parametric approach outperformed traditional TCN models without parametric features. Even when tested across three different stocks, the DeepTCN model showed consistent and stable performance. Both models perform much better than a basic, simple naïve model when it comes to predicting stock data. We used a parametric approach based on the half-normal distribution which ensures that the dataset's variance remains positive, a critical aspect when dealing with stock prices where negative values are not possible.

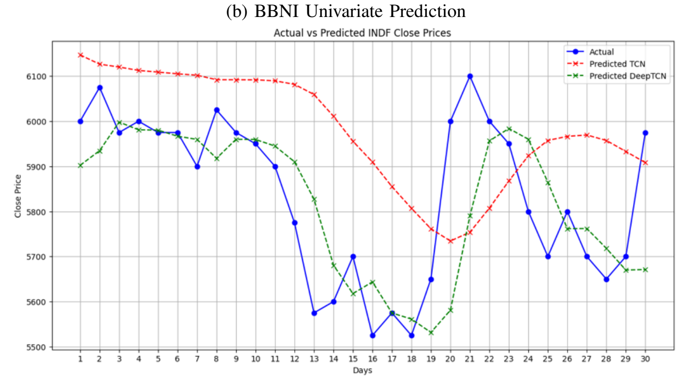
The forecasts shown in Fig. 4 highlight that DeepTCN outperforms TCN across all observed stock emitters. The parametric approach employed by DeepTCN allows the model to recognize and adapt to highly volatile stock prices, which are common in dynamic markets. While both models can capture the general direction of stock price movements, DeepTCN excels at identifying intricate patterns and trends in the data. In contrast, TCN's predictions tend to be more linear, lacking the detailed recognition that DeepTCN offers. This distinction in pattern recognition underscores the more advanced capabilities of DeepTCN in providing accurate and robust stock price predictions across different companies and time windows.



(a) ACES Univariate Prediction



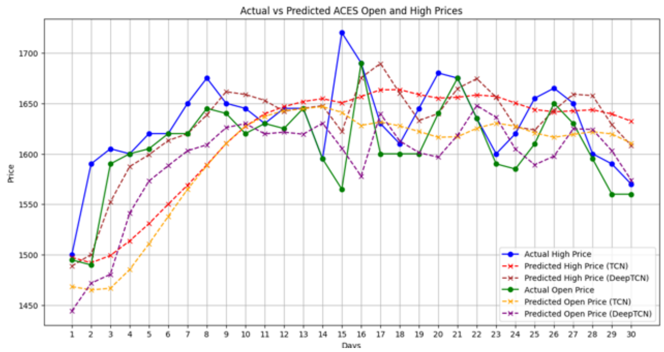
(b) BBNI Univariate Prediction



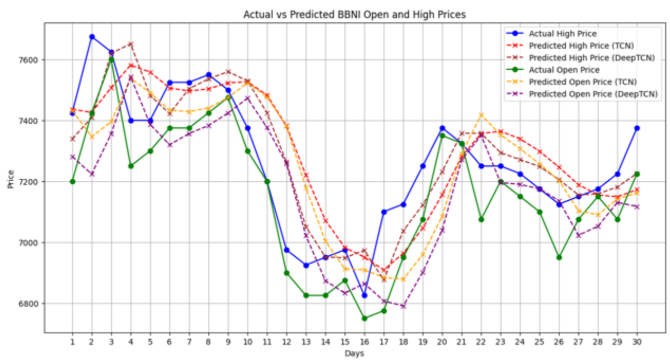
(c) INDF Univariate Prediction

Figure 4. Forecast comparison between TCN and DeepTCN for univariate predictions

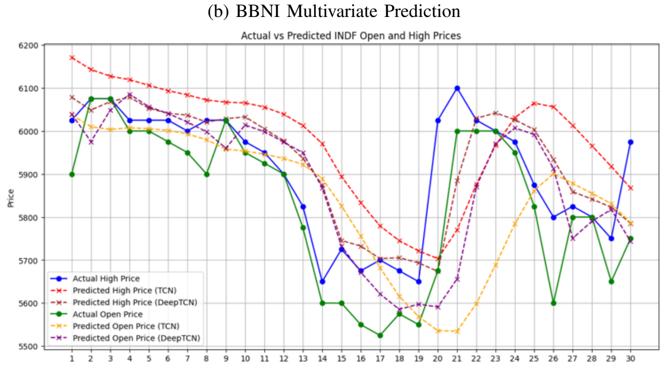
Additionally, the predictions in Figure 5 demonstrate that DeepTCN consistently outperforms TCN across all observed stock emitters, with its parametric approach allowing it to adapt to the highly volatile and dynamic nature of stock prices. While both models can capture general trends, DeepTCN excels in identifying complex patterns and nuances, whereas TCN's predictions are more linear and lack detailed pattern recognition. Notably, the multivariate prediction results closely mirror those from the univariate approach, further affirming DeepTCN's robustness in delivering accurate forecasts regardless of input dimensions. This underscores its superiority in navigating the challenges of the Indonesian stock market, an emerging market characterized by high volatility and sensitivity to both local and global events.



(a) ACES Multivariate Prediction



(b) BBNI Multivariate Prediction



(c) INDF Multivariate Prediction

Figure 5. Forecast comparison between TCN and DeepTCN for multivariate predictions

D. Statistical Validation

To strengthen the claim that DeepTCN outperforms TCN, statistical validation was conducted using a paired t-test. This method aims to assess the significance of the performance difference between the two models based on RMSE, ensuring that DeepTCN superiority is not merely coincidental but is supported by statistical evidence. A paired t-test was conducted to assess whether the differences in RMSE between the two models were statistically significant.

$$t - value = \frac{\bar{d}}{s_d / \sqrt{n}} \tag{8}$$

The hypotheses for the test were as follows: the null hypothesis H_0 stated that the mean difference in RMSE was zero, indicating no significant performance difference, while the alternative hypothesis H_a proposed that the mean difference was greater than zero, implying DeepTCN's superior performance. The results showed a t-statistic of 3.36 and a p-value of 0.0027. Since the p-value is significantly lower than the standard threshold ($\alpha = 0.05$), the null hypothesis was rejected, providing strong statistical evidence that DeepTCN significantly outperforms TCN in predicting stock prices.

6. Conclusion

In this study, a comparison analysis was conducted between TCN and DeepTCN models in forecasting stock prices using 3 Indonesian stock historical price data. In summary, DeepTCN demonstrates its superiority in stock price prediction compared to TCN. DeepTCN is capable of outperforming TCN by achieving lower values of MSE, MAPE, and RMSE. The half-normal distribution parametric approach used in this study has proven to make DeepTCN better at capturing fluctuating stock trends. Datasets limited to 2023 may not fully capture emerging patterns or significant events occurring in subsequent years, which are critical for accurate stock price forecasting. For example, new geopolitical events, changes in monetary or fiscal policies, and breakthroughs in technology can significantly impact stock market dynamics.

The future work recommended in this study involves evaluating the performance of TCN and DeepTCN on a varied set of datasets. This evaluation aims to understand the capabilities of DeepTCN across different types of data characteristics and tasks. By assessing these models on diverse datasets, researchers can gain insights into how well DeepTCN generalizes and performs in various scenarios. When considering parametric approaches, it is important to choose a method that aligns with the characteristics of the dataset and the specific goals of the analysis. This evaluation can provide valuable insight for understanding the strengths and limitations of DeepTCN and guide its application in real-world datasets across different domains.

Author Contribution

This research was planned by Felix and Evandiaz Fedora. Experiments and paper writing were conducted by Felix and Evandiaz Fedora with the guidance of Alexander Agung Santoso Gunawan. All authors have read and approved the manuscript.

Availability Data and Materials

This research utilizes data from Indonesia Stock Dataset, which is available at https://www.kaggle.com/datasets/muamkh/ihsgstockdata.

References

[1] H. N. Dao, W. ChuanYuan, A. Suzuki, H. Sudo, L. Ye, and D. Roy, "AI in stock market forecasting: A bibliometric analysis," *SHS Web of Conferences*, vol. 194, 2024.

[2] C. Fan and X. Zhang, "Stock price nowcasting and forecasting with deep learning," *Journal of Intelligent Information Systems*, pp. 1–18, 2024.

[3] O. Samuel and A. Kadar, "Time-series forecasting with deep learning: A new approach to stock market predictions," 2024.

[4] V. Katragadda and O. Richard, "Harnessing deep learning for stock market forecasting: Insights from time-series data," 2024.

[5] N. Kahan and O. Thomas, "Deep learning in financial forecasting: Unlocking the potential of stock market prediction," 2024.

[6] Y. Chen, Y. Kang, Y. Chen, and Z. Wang, "Probabilistic forecasting with temporal convolutional neural network," *Neurocomputing*, vol. 399, pp. 491–501, 2020.

[7] I. Bhattacharjee and P. Bhattacharja, "Stock price prediction: A comparative study between traditional statistical approach and machine learning approach," in *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, 2019.

[8] A. Durgapal and V. Vimal, "Prediction of stock price using statistical and ensemble learning models: A comparative study," in *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, 2021.

[9] H. Grigoryan, "A stock market prediction method based on support vector machines (SVM) and independent component analysis (ICA)," *Database Systems Journal*, vol. 7, no. 1, pp. 12–21, 2016.

[10] R. Singh and S. Srivastava, "Stock prediction using deep learning," *Multimedia Tools and Applications*, vol. 76, no. 18, pp. 18 569–18 584, 2017.

[11] K. A. Althelaya, E.-S. M. El-Alfy, and S. Mohammed, "Evaluation of bidirectional LSTM for short-and long-term stock market prediction," in *2018 9th International Conference on Information and Communication Systems (ICICS)*, 2018.

[12] Y. Liu, C. Gong, L. Yang, and Y. Chen, "DSTP-RNN: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction," *Expert Systems with Applications*, vol. 143, p. 113082, 2020.

[13] Hiransha, Gopalakrishnan, V. K. Menon, and Soman, "NSE stock market prediction using deep-learning models," *Procedia Computer Science*, vol. 132, pp. 1351–1362, 2018.

[14] Y. Fu and H. Xiao, "Stock price prediction model based on dual attention and TCN," *SSRN Electronic Journal*, 2022.

[15] S. Deng, N. Zhang, W. Zhang, J. Chen, J. Z. Pan, and H. Chen, "Knowledge-driven stock trend prediction and explanation via temporal convolutional network," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019.

[16] J. Zhang and W. Wang, "A stock trend prediction model based on wavelet transform and TCN combined with market sentiment," in *Proceedings of the 4th International Conference on Informatization Economic Development and Management (IEDM)*, Kuala Lumpur, Malaysia, February 23–25 2024.

[17] Z. Zhang, Y. Matsuda, and Y. Liu, "TCN BERT attention guided stock market movement prediction," in *Proceedings of the 5th International Conference on Computer Information and Big Data Applications*, 2024.

[18] H. V. Dudukcu, M. Taskiran, Z. G. C. Taskiran, and T. Yildirim, "Temporal Convolutional Networks with RNN approach for chaotic time series prediction," *Applied Soft Computing*, vol. 133, p. 109945, 2023.

[19] H. Liu, T. Zhao, S. Wang, and X. Li, "A stock rank prediction method combining industry attributes and price data of stocks," *Information Processing and Management*, vol. 60, no. 4, p. 103358, 2023.

[20] J. V. Belle, R. Crevits, D. Caljon, and W. Verbeke, "Probabilistic forecasting with modified N-BEATS networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, pp. 1–0, 2024.

[21] A. Baba, "Probabilistic-based forecasting method for time series datasets," *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, vol. 11, no. 2, pp. 563–573, 2023.

[22] K. Maciejowska, T. Serafin, and B. Uniejewski, "Probabilistic forecasting with a hybrid Factor-QRA approach: Application to electricity trading," *Electric Power Systems Research*, vol. 234, p. 110541, 2024.

[23] C. Zhao, S. Yang, C. Qin, J. Zhou, and L. Chen, "A novel smart beta optimization based on probabilistic forecast," *Computers, Materials Continua*, vol. 75, no. 1, pp. 477–491, 2023.

[24] V. Jensen, F. M. Bianchi, and S. N. Anfinsen, "Ensemble conformalized quantile regression for probabilistic time series forecasting," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, 2022.

[25] J. de Vilmarest and N. Werge, "An adaptive volatility method for probabilistic forecasting and its application to the M6 financial forecasting competition," *International Journal of Forecasting*, 2024.

[26] P. Lara-Benítez, M. Carranza-García, J. M. Luna-Romera, and J. C. Riquelme, "Temporal Convolutional Networks applied to energy-related time series forecasting," *Applied Sciences (Basel)*, vol. 10, no. 7, p. 2322, 2020.

[27] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal Convolutional Networks for Action Segmentation and Detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[28] A. Casolaro, V. Capone, G. Iannuzzo, and F. Camastra, "Deep learning for time series forecasting: Advances and open problems," *Information (Basel)*, vol. 14, no. 11, p. 598, 2023.

[29] J. You, Y. Wang, A. Pal, P. Eksombatchai, C. Rosenburg, and J. Leskovec, "Hierarchical temporal convolutional networks for dynamic recommender systems," in *The World Wide Web Conference*, 2019.

[30] R. Wan, S. Mei, J. Wang, M. Liu, and F. Yang, "Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting," *Electronics (Basel)*, vol. 8, no. 8, p. 876, 2019.

[31] K. Cooray and M. M. A. Ananda, "A generalization of the half-normal distribution with applications to lifetime data," *Communications in Statistics - Theory and Methods*, vol. 37, no. 9, pp. 1323–1337, 2008.

[32] M. Tsagris, C. Beneki, and H. Hassani, "On the Folded Normal Distribution," *Mathematics*, vol. 2, no. 1, pp. 12–28, 2014.

[33] J. Palet, V. Manquinho, and R. Henriques, "Multiple-input neural networks for time series forecasting incorporating historical and prospective context," *Data Mining and Knowledge Discovery*, 2023.

[34] V. Plevris, G. Solorzano, N. Bakas, and M. B. Seghier, "Investigation of performance metrics in regression analysis and machine learning-based prediction models," in *8th European Congress on Computational Methods in Applied Sciences and Engineering*, 2022.

[35] M. S. Chowdhury *et al.*, "Deep learning models for stock market forecasting: A comprehensive comparative analysis," *Journal of Business and Management Studies*, vol. 6, no. 2, pp. 95–99, 2024.

[36] M. Khadafi, "IHSG Stock Data," https://www.kaggle.com/datasets/muamkh/ihsgstockdata, 2024, accessed: 2024-06-18.