Transformer Based Swing Level Detection in Forex

Fatih Erdoğan * [1]

Abstract

This study focuses on the use of transformer-based models for predicting swing levels in the foreign exchange (forex) market. Our method differs from the traditional approach which is price forecasting and targets to identify significant reaction points in price action (PA). Therefore, the objective is both interesting and important as it aligns more closely with the strategic decision-making processes of traders, who often base their actions on key market turning points rather than on continuous price movements. Our main contributions can be summarized as the novel prediction target (swing level), the introduction of 2 transformer based models (TFT and TOTFM), a new approach in data preprocessing, and a task specific evaluation metric measuring the trading effectiveness of the models in a real-world context. The project code is available on GitHub: https://github.com/Fatih-Erdogan/swing-project.git

1. Introduction

The price movements in the modern stock market are influenced by a variety of factors including the news, human psychology etc. Apart from the fundamental analysis including financial statements evaluation, economic indicators, company management etc., the technical analysis which makes use of the historical movements, trading volumes, etc. is considered as a valuable tool to predict the future movements. The method of extracting meaningful information with this tool and using it to inform trading decisions in the stock market is called price action (PA). The purpose of this method is not to predict the price of the interested stock in the upcoming time interval but it aims to determine the important areas where the price is likely to receive an important reaction and change the direction of its movement. With this idea, this project differs from the others in the prediction target, which we set as the next swing level. We built 2 transformer based models which are fed with data that are considered useful in PA. The attention mechanism allowing to focus on the relevant parts of the data is expected to be useful as the swing prices are mostly affected by the price at these important areas. Therefore, the underlying task of the model is to detect these important areas in the historical data. The usefulness of such a model in trading decisions, the remarkable results of the transformer architecture in other fields and the convenience of the determined task to the attention mechanism make this research topic worth to be explored.

In the rest of the paper, we will first present the related work targeting similar objectives. Then, we will introduce our baseline and dive into our models' design with explanations on the components. Finally, we will share our experimental setup and experimental results based on the evaluation metrics that we found useful to measure the model performances.

2. Related Work

Various research was conducted to create a model estimating the price movements for the financial stock markets. Some of them compare several machine learning algorithms with deep learning algorithms (Sakhare & Imambi, 2022). Some research came up with a custom model based on LSTM architecture combined with a custom attention mechanism (Liou & Huang, 2023). Moreover, there exist research that make use of some textual data which might benefit to the price prediction (Prachyachuwong & Vateekul, 2021). We also observed that the most popular algorithm for price prediction is LSTM. (Yan & Ouyang, 2018) proposed the reconstruction of data points with wavelet analysis and provided the results of their experiments showing LSTM outperforming MLP, SVM, and KNN. (Jeevan et al., 2018) and (Selvin et al., 2017) compared the performances of LSTM, RNN; and LSTM, RNN, and CNN's respectively in price prediction. (Totakura et al., 2020) focuses on the trend prediction based on the RSI indicator using LSTM. There exist numerous other research using LSTM's for price prediction such as (Prachyachuwong & Vateekul, 2021; Banik et al., 2022; Juairiah et al., 2022; Liou & Huang, 2023). However, there exist very limited number of experiments made on transformer architecture which is the case even in

[1]Department of Computer Engineering. Correspondence to: Fatih Erdoğan <ferdogan20@ku.edu.tr>.

COMP441Deep Learning, Fall 2023.

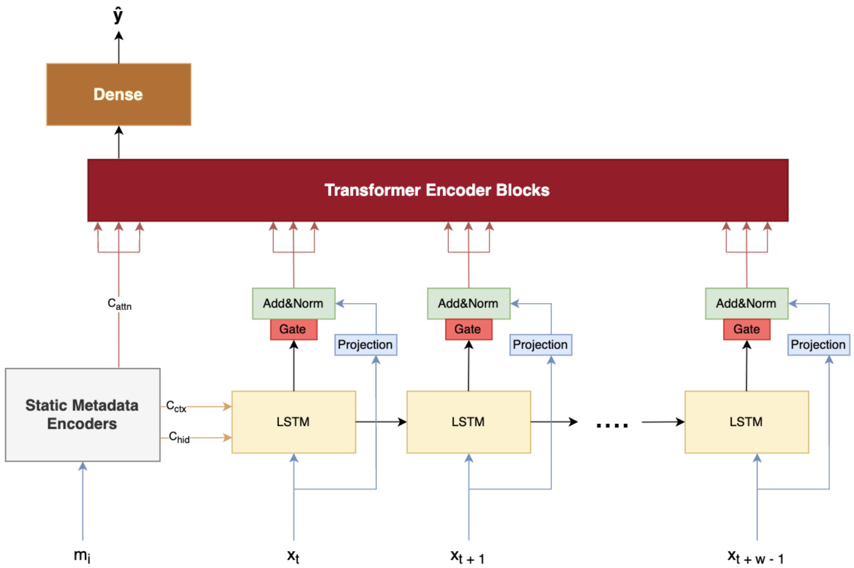

Figure 1. Illustration of the adapted TFT model

the recent research. And some research which are testing transformer's performance suffer from the low number of features associated with each day (Muhammad et al., 2023). Our approach differs from all mentioned research in the training objective which is predicting the next swing level. We believe that this approach can eliminate the confusion resulting from the attempt of predicting each single day correctly and direct the model towards learning the common patterns constructing the levels where the price is likely to receive an important reaction. On the other hand, some research provides valuable insights about how to capture and embed temporal relationships into the data points. (Lim et al., 2021) proposed Temporal Fusion Transfomer (TFT) for multi-horizon forecasting (predicting multiple future values in a time series based on the current and past sequence of data). Their idea is to feed the input sequence into an LSTM, injecting temporal and contextual information into the input, which is then passed to the attention layer. Inspired by this idea and replicating and/or replacing some other properties of that model, we constructed 2 models which differs from each one of the related work in the nature of the prediction target (being predicting the next swing level).

3. The Approach

Inspired by the model design of (Lim et al., 2021) that they call Temporal Fusion Transformer (TFT), we came up with 2 transformer based models. The first model that we call TFT as well in the rest of the paper, is very similar to the originally proposed one with slight modifications in the architecture due to the difference in the nature of the prediction target. The second model that we call Transformer Only Temporal Fusion Model (TOTFM) is created by replacing a core component of the TFT called "Temporal Enrichment Module". We will discuss these architectures more in detail in Sections 3.1 and 3.2.

3.1. Temporal Fusion Transformer

The TFT model in Figure 1 contains 4 components. The first one is called "Static Metadata Encoders". The purpose of this module is to encode the metadata containing global information about the context of the current sequence. It is responsible for generating the initial cell and hidden states for the "Temporal Enricher Module" built on LSTM as well as a token for the "Transformer Encoder Blocks". Static Metadata Encoder is composed of a number of GRN blocks depending on the number of LSTM layers in Temporal Enricher Module. Each GRN block, which are provided with the 26 dimensional metadata vector first maps this vector to a higher intermediate dimension (that we set as $4 \times model\_dimension$) which is followed by an activation function. Then, it is mapped to the output dimension, which is determined by the feature dimension of the Transformer

Temporal Enricher		
bidir.	**#layers**	**dropout**
False	1	0.4

Transformer Block				
pos.emb.	**dim**	**#layers**	**#heads**	**dropout**
Sinusoidal	64	1	4	0.4

Table 1. Adapted TFT model parameters.

Blocks (64). Following that, a gate module that uses a sigmoid function and maintains the original vector dimension is applied to remove redundant dimensions from the resulting vector. Eventually, the result goes into an add and norm layer with the projected version (to match to the output dimension) of the metadata and that forms the output of a GRN block. You can find the described process in the following equations:

$$GRN(m_i) = LayerNorm(W_3(m_i) + Gate(y_i)) \quad (1)$$

$$Gate(y_i) = \sigma(W_m y_i + b_m) \odot (W_v y_i + b_v) \quad (2)$$

$$y_i = W_2(ELU(W_1 m_i)) \quad (3)$$

After the second component "Temporal Enricher Module" being an LSTM receives the initial cell and hidden states, it processes the input sequence composed of 5 dimensional 336 tokens. Again, the hidden dimension of the LSTM is determined by the model dimension. Then, the output of LSTM goes into a gate (see Equation 2) to eliminate its redundant dimensions and an add and normalization layer is applied with the projected version of the original token. Before the Transformer Encoder processes the tokens, firstly the metadata token generated by the Static Metadadata Encoder is enriched with a learnable bias vector according to the direction of the prediction. Even though this approach seems to be cheating, in a real life scenario, that wouldn't harm the usefulness of the model as the target is not to predict the direction of the trend. Then, the metadata token is concatenated to the beginning of each sequence, followed by the addition of sinusoidal embeddings to make them position aware. Even though the authors of the original TFT model claimed the unnecessity of the addition of poisitional embeddings, according to the ablation in Section 4.4 we found useful to include them. After the Transformer Encoder processes the input sequence of length 337 (including the metadata token), to obtain the final output, we followed a BERT style approach and get the output corresponding to the metadata token processed by the last component, being a fully connected layer. The resulting continuous value is the next swing level prediction of the model. You can find the model parameters in Table 1

3.2. Transformer Only Temporal Fusion Model

Suffering from the sequential processing of the Temporal Enricher Module (LSTM) in the TFT model which causes an increased amount of training time and questioning it's usefulness, by replacing that module with a GRN block, we came up with a new model architecture that we call Transformer Only Temporal Fusion Model (TOTFM). The components of this model can be listed as: Static Metadata Encoder, Feature Augmentation Layer, Transformer Encoder, and Output Layer.

In this model, the function of Static Metadata Encoder is solely the generation of the metadata token which will be processed by the Transformer Encoder. To enrich the input sequence elements with the metadata information which was the case in TFT due to the generated initial hidden and cell state of the LSTM, we needed to integrate this information in another manner. To achieve this, we used the dual version of the GRN block. The GRN block receiving the sequence element and the raw metadata, first maps them to an intermediate dimension which we determined as $4 \times model\_dimension$. Then the results are summed before going into the activation function. After that, the resulting vector is processed by a fully connected layer and the dimension is reduced to the model dimension. This process is followed by a gate as in the Equation 2. Finally, an add and normalization layer is applied with the projected version of the input token. You can see the Figure 2 for the illustration of dual GRN block.

After the GRN block processes the input sequence, the metadata token enhanced with the learnable directional bias vector is concatenated as the first sequence element and the sequence is summed with static positional embeddings and dropout is applied. After the Transformer Encoder processes the sequence the output corresponding to the metadata token is passed to the Output Layer to obtain the swing level prediction. You can find the model parameters in Table 2

3.3. Training

For each model, we followed a similar training pipeline. First the models are trained on a wide range of data including 19 forex pairs hourly candles from January 2020 to September 2022. We followed a shuffled approach meaning that we didn't train the models chronologically. The purpose behind this decision is to prevent models fitting to the recent patterns observed in the data. The objective of this step that

GRN		
inp.dims	**int_dim**	**dropout**
5, 26	256	0.2

Transformer Block				
pos.emb.	**dim**	**#layers**	**#heads**	**dropout**
Sinusoidal	64	1	4	0.4

Table 2. Transformer Only Temporal Fusion Model parameters. The first value in GRN input dimension field corresponds to token dimension of the input sequence and the second one to the metadata token

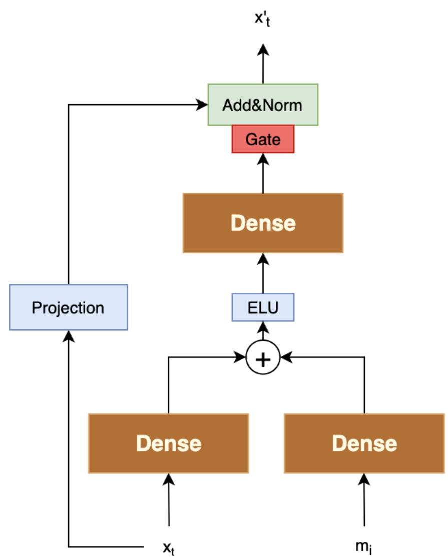

Figure 2. Illustration of the dual version of a GRN bock

Model	LR	WD	Epochs	t/epoch(mins)
Baseline	5*1e-5	1e-4	19	8.3
TFT	8*1e-4	1e-3	12	12.3
TOTFM	8*1e-4	1e-3	7	9.2

Table 3. Training hyperparameters for the baseline and the proposed models. LR stands for "Learning Rate", WD for "Weight Decay", and t/epoch for "time it requires for a single epoch".

we call pretraining is to teach the model general patterns forming the swing levels. Then, the models are finetuned for a few epochs with a very decreased learning rate on a specific pair to learn pair specific properties. Therefore, the resulting model is expected to be used and evaluated on the specific pair that it was finetuned on. Due to the time constraints, we were only able to make experimental evaluation on AUDUSD pair (see Section 4.3).

We trained all models on a singe V100 GPU provided by the Google Colab. We used the Mean Square Error loss function to optimize the model predictions. We used AdamW as the default optimizer and 16 as the default batch size for each model. As the baseline model contains less number of parameters, we used a relatively smaller learning rate for a stable training. As you can observe in the Table 3, the TOTFM model requires less training time as the LSTM component creates bottleneck for the other models.

4. Experimental Setup

In this section we will present the dataset that we used to train and evaluate the models. We will also provide the method to preprocess the data and prepare it to train the models. Then, we will introduce our baseline and evaluation metrics that we used to compare the performances of the proposed models TFT and TOTFM. Finally, we will share our experimental results and the ablations.

4.1. Dataset

To train and test the models, we collected hourly candles of 19 pairs from TradingView platform. We selected each pair from foreign exchange (forex) market to increase the similarity in the nature of price movements. We were able to collect 4 years of data for each pair between January 2020 and January 2024 due to the limitations of TradingView. Each candle contains open, high, low, close, and volume (OHLCV) information associated with it. We also needed regression labels corresponding to the swing levels. As the "swing level" doesn't have an exact definition, we used an indicator ZigZag++ which makes use of the previous and future candles to determine whether the price levels visited by a candle corresponds to a swing level or not, according to some predetermined hyperparameters. After collecting these data, we split it into training, validation, and test sets chronologically with proportions 0.7, 0.2, and 0.1 respectively.

$$mid = \frac{1}{2}(pw\_high + pw\_low) \tag{4}$$

$$half\_range = \frac{1}{2}(pw\_high - pw\_low) \tag{5}$$

$$x'_{OHLC} = \frac{x_{OHLC} - mid}{half\_range} \tag{6}$$

One significant problem in exchange market is the non-stationary nature of the data. To overcome this problem we first extracted weekly information from the hourly candles. The weekly data contains the weekly OHLC values. Additionally, we extracted the Volume Profile Visible Range (VPVR) indicator values for each week that we adapted the source code from an external resource[1]. This indicator provides the trading volumes corresponding to the price levels which is assumed to be beneficial to determine important price levels. We also included the next Monday's OHLC values in the weekly data as it is considered to form institutional price levels. Although including Monday data results in a direct shortcoming by disabling the model to make a prediction on Mondays, we believe the benefit of adding it as a metadata outweights this shortcomming. To address the non-stationarity problem and to embed some of the important price levels into the sequence inherently, we normalized the OHLC values in input sequence and the related metadata (previous weekly data and the Monday data)

[1] https://gist.github.com/4skinSkywalker/d5e42f46851decf69054e0d0287ab6f5

		1R			2R			3R		
Model	MSE	accuracy	# setups	profit	accuracy	# setups	profit	accuracy	# setups	profit
Baseline	0.294	0.64	34	10	0.41	43	11	0.37	45	23
TFT (adapted)	0.301	0.82	23	15	0.48	29	13	0.31	32	8
TOTFM (ours)	0.297	0.85	28	20	0.52	33	18	0.39	38	22

	Absolute Error Bin Limits									
Model	<0.1	<0.2	<0.3	<0.4	<0.5	<0.6	<0.7	<0.8	<0.9	<1.0
Baseline	0.169	0.339	0.503	0.642	0.752	0.833	0.894	0.938	0.960	0.970
TFT (adapted)	0.139	0.323	0.503	0.643	0.763	0.849	0.894	0.922	0.941	0.955
TOTFM (ours)	0.197	0.360	0.521	0.664	0.740	0.813	0.858	0.904	0.926	0.937

Table 4. Comparison of the models performances on AUDUSD pair. Error calculation is based on the half range of the previous week. Table at top shows the Mean Square Error (MSE) and Trade Setup Revenue Metric (TSRM) scores. Table at bottom presents the percentages of absolute error falling into the corresponding cumulative bin.

with the mid level, and the half of the range of the previous week. This approach results in a zero centered data, and prevents large deviations in the price except for the cases of major trend changes. Additionally, the Fibonacci levels (considered as important levels by price action community) determined by the range formed in the previous week are inherently embedded in the data allowing the model to create a positive bias for such important levels. For the volume, we used a more straightforward method which is standardizing the values considering the sequence elements. You can find the described normalization process for OHLC values in Equations 4 - 6

4.2. Baseline and Metrics

Unfortunately, most of the related work was focused on determining the next candles' closing price. Therefore, we weren't able to pick a baseline from them. As the LSTM based architecture was the most prevalent one among the related work, we built our own baseline as an LSTM based one. The baseline contains a Metadata Encoder which is composed of 4 GRN blocks. Each block is responsible for the generation of the initial hidden or cell states for the LSTM component. We used a bidirectional architecture in this component to better capture the relationships between the sequence elements. Then the outputs corresponding to the first and last sequence elements goes into a gate (see Equation 2) to eliminate unnecessary dimensions and an add and normalization layer is applied with the projected versions of the raw input. Finally, these 2 resulting vectors are summed and passed to a fully connected layer to obtain the swing level prediction. You can see the baseline model in Figure 3.

To measure the model performances we used several metrics. First one being the usual one for regression tasks is the Mean Square Error (MSE) which is also the metric that the models are optimized for. We used this metric to measure the overall performances of the models. However, as the eventual target of this project is to build a model that can be beneficial for trading decisions, we used Binned Absolute Error to observe the nature of the errors that the models are doing. Therefore, the purpose of this metric is to have an idea about the proportion of the errors that are less than a given value. Finally, we designed a custom task specific metric that we call Trade Setup Revenue Metric (TSRM). The metric works as the following:

$$StopLoss = entry - \frac{take\_profit - entry}{R} \tag{7}$$

At a given point in time, the model is prompted to generate a pair of predictions, one for a swing high and one for swing low. Then, for a predetermined number of days, model waits for the price reach to one of these 2 levels. If none of the levels are reached during this time period, the model is prompted to make another pair of predictions. If one of the levels are reached, it is considered as the entry level. Then, the model is prompted to predict another swing level in the opposite direction which will be the take profit level. According to the metrics hyperparameter (risk-reward ratio, called R), a stop-loss is determined based on the distance between the entry level and the stop-loss. You can see

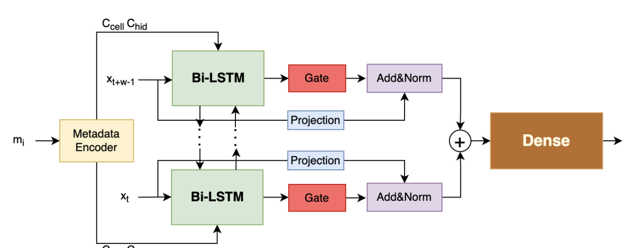

Figure 3. LSTM based baseline model

Equation 7 for the stop-loss level calculation.

Finally, if the price reaches stop loss before the take profit, the result is counted as (-1) and if it reaches to take profit before, it is counted as (+ Risk Reward Ratio).

4.3. Experimental Results

Due to time constraints, we were only able to finetune our pretrained models on AUDUSD pair. All the results shared in this study are evaluated on the test set of AUDUSD pair, which contains data between June 2023 and January 2024.

You can observe the results of the baseline and our models TFT and TOTFM in Table 4. We evaluated the models using 3 metrics that are described in Section 4.2. The MSE results suggests similar performances where the baseline is the leading one and the TFT model performing the worst. Observing this metric, even though it can be thought as the baseline is generalizing better than the other models, the Absolute Error Bins metric allowing a more detailed analysis of the nature of the errors shows that the the TOTFM model has much more predictions that are falling into first four bins than the other models. Additionally, observing the last column of the table, the baseline model has 3.5% (relative) less number of points that are falling out of 1 half range bin compared to the TOTFM. These two remarks partially explains the similarity of the MSE metric results between these two models, although the nature of the prediction errrors are highly different. Given the importance of accuracy in creating trading setups, a model making very close predictions at times and very far predictions at other times would be more preferable over a model that consistently has a moderate level of error. This claim is coherent with the results of the TSRM metric where the TOTFM model is clearly outperforming the others in each of 1R, 2R, and 3R setups.

4.4. Ablations

We conducted numerous ablations during this study and in this section, we will share some of the most interesting ones. The results in this section are obtained by the evaluation of finetuned models on the metrics described in Section 4.2. For the ablations, we used the validation set of the AUDUSD pair containing data between September 2022 to June 2023.

In their work, (Lim et al., 2021) suggests that the addition of positional encodings are unnecesarray before the Transoformer Encoder as the Temporal Enricher (LSTM) of the Temporal Fusion Transformer already embeds the temporal information in the vectors due to the sequential processing. To test the validity of their claim, we ablated on the positional embeddings of TFT model. You can see the results of the TFT model with no positional embeddings, with learnable embeddings, and with sinusoidal embeddings in Table 5. The results showed slight increases in each metric with the addition of positional embeddings. Therefore, considering the number of learnable parameters of the model as well, we preferred to use the sinusoidal embeddings.

We also ablated on the number of transformer encoder layers of TFT model (see Table 6). The results showed that the as the number of blocks increases, the model looses its capability of generalization and overfits to the training data very fast. It also resulted in an unstable training as even though the training loss decreases as expected, the validation loss fluctuates too much with a high amplitude.

We also shared our findings on the effect of the bidirectionality of the Temporal Enricher module of the TFT model in Table 7. As the unidirectional one performed slightly better, and considering the number of learnable parameters as well we decided to use a unidirectional LSTM in the Temporal Enricher module.

In Table 8, you can see the results comparing 2 versions of TOTFM with different model sizes. The one with 64 as the model dimension (transformer dimension) outperforms the one with greater model size in each of the metrics. We also conducted experiments with a model with dimension 32; however, as it was performing poorly in pretraining stage probably because of the underfitting, we didn't finetune it for further evaluation.

5. Conclusions

In this project, we aimed to construct transformer-based models predicting swing levels in forex market based on the technical analysis tools. By focusing on spotting key reaction points in price action, we targeted at building a model that could benefit the trading decision-making process. With this purpose, we came up with 2 model architectures Temporal Fusion Transformer and Transformer Only Temporal Fusion Model (the base architecture of this study proposed by (Lim et al., 2021)). Additionally, we proposed a new preprocessing method embedding metadata information into the data itself, which is a new approach according to our knowledge. We also introduced a new evaluation metric that we call Trade Setup Revenue Metric (TSRM) which matches closely with our ultimate goal, measuring the performance of the model in a real-world scenario.

Even though the proposed models were not able to perform significantly better than the baseline in each metric, they gave promising results especially in the TSRM. Analyzing the models inference in more in detail, we realized that the models are generating poor predictions if the prediction target is outside of the price range covered by the current sequence. Therefore, the most significant limitation of the model is that it cannot generate accurate results during sharp

		1R			2R			3R		
Pos. Emb.	MSE	accuracy	# setups	profit	accuracy	# setups	profit	accuracy	# setups	profit
-	0.302	0.70	60	24	0.36	72	6	0.43	88	64
Learnable	0.300	0.66	54	18	0.54	66	42	0.39	82	46
Sinusoidal	0.300	0.72	58	26	0.46	78	30	0.39	86	50

	Absolute Error Bin Limits									
Pos. Emb.	<0.1	<0.2	<0.3	<0.4	<0.5	<0.6	<0.7	<0.8	<0.9	<1.0
-	0.177	0.352	0.510	0.665	0.773	0.823	0.864	0.906	0.922	0.938
Learnable	0.184	0.364	0.527	0.658	0.767	0.848	0.895	0.922	0.946	0.961
Sinusoidal	0.183	0.379	0.547	0.691	0.809	0.856	0.892	0.921	0.947	0.961

Table 5. Ablation on addition of positional encodings to the TFT model.

		1R			2R			3R		
# Trans. Lay.	MSE	accuracy	# setups	profit	accuracy	# setups	profit	accuracy	# setups	profit
2	0.328	0.86	46	34	0.48	58	26	0.32	80	24
1	0.298	0.78	46	26	0.52	58	32	0.50	44	84

	Absolute Error Bin Limits									
# Trans. Lay.	<0.1	<0.2	<0.3	<0.4	<0.5	<0.6	<0.7	<0.8	<0.9	<1.0
2	0.158	0.303	0.488	0.647	0.769	0.847	0.897	0.924	0.950	0.960
1	0.183	0.379	0.547	0.691	0.809	0.865	0.892	0.921	0.947	0.961

Table 6. Ablation on number of transformer layers in the TFT model.

		1R			2R			3R		
Bidirec.	MSE	accuracy	# setups	profit	accuracy	# setups	profit	accuracy	# setups	profit
True	0.300	0.72	58	26	0.46	78	30	0.39	46	50
False	0.298	0.78	46	26	0.52	58	32	0.50	44	84

	Absolute Error Bin Limits									
Bidirec.	<0.1	<0.2	<0.3	<0.4	<0.5	<0.6	<0.7	<0.8	<0.9	<1.0
True	0.181	0.381	0.546	0.680	0.801	0.850	0.877	0.906	0.932	0.946
False	0.183	0.379	0.547	0.691	0.809	0.865	0.892	0.921	0.947	0.961

Table 7. Ablation on bidirectionality of the LSTM component of TFT.

		1R			2R			3R		
Model Dim.	MSE	accuracy	# setups	profit	accuracy	# setups	profit	accuracy	# setups	profit
128	0.334	0.68	44	16	0.52	62	34	0.43	70	50
64	0.291	0.89	54	42	0.56	60	42	0.40	70	42

	Absolute Error Bin Limits									
Model Dim.	<0.1	<0.2	<0.3	<0.4	<0.5	<0.6	<0.7	<0.8	<0.9	<1.0
128	0.151	0.331	0.484	0.624	0.740	0.825	0.882	0.920	0.942	0.961
64	0.183	0.354	0.488	0.637	0.758	0.820	0.881	0.917	0.941	0.952

Table 8. Ablation on the dimension of the TOTFM.

trend changes, as price movement is affected by earlier candles that are not present in the current window.

Future work can focus on a wider evaluation on many other pairs in other trading markets as well to confirm the usefulness of the proposed models confidently. Additionally the regression task can be converted to a classification task by dividing the prediction range of the model into separate bins. Moreover, considering the success of the transformers in end-to-end learning, another interesting extension could be the usage of the candles tabular data to classify the candles, and making the transformer module learn the embeddings for them itself.

References

Banik, S., Sharma, N., Mangla, M., Mohanty, S. N., and S., S. Lstm based decision support system for swing trading in stock market. *Know.-Based Syst.*, 239(C), mar 2022. ISSN 0950-7051. doi: 10.1016/j.knosys.2021.107994. URL https://doi.org/10.1016/j.knosys.2021.107994.

Jeevan, B., Naresh, E., kumar, B. P. V., and Kambli, P. Share price prediction using machine learning technique. In *2018 3rd International Conference on Circuits, Control, Communication and Computing (I4C)*, pp. 1–4, 2018. doi: 10.1109/CIMCA.2018.8739647.

Juairiah, F., Mahatabe, M., Jamal, H. B., Shiddika, A., Rouf Shawon, T., and Chandra Mandal, N. Stock price prediction: A time series analysis. In *2022 25th International Conference on Computer and Information Technology (ICCIT)*, pp. 153–158, 2022. doi: 10.1109/ICCIT57492.2022.10056009.

Lim, B., Arık, S. , Loeff, N., and Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2021.03.012. URL https://www.sciencedirect.com/science/article/pii/S0169207021000637.

Liou, H.-I. and Huang, K.-C. A deep learning model for stock price prediction in swing trading. In *2023 9th International Conference on Applied System Innovation (ICASI)*, pp. 154–156, 2023. doi: 10.1109/ICASI57738.2023.10179575.

Muhammad, T., Aftab, A. B., Ibrahim, M., Ahsan, M. M., Muhu, M. M., Khan, S. I., and Alam, M. S. Transformer-based deep learning model for stock price prediction: A case study on bangladesh stock market. *International Journal of Computational Intelligence and Applications*, 22(03), apr 2023. doi: 10.1142/s146902682350013x. URL https://doi.org/10.1142%2Fs146902682350013x.

Prachyachuwong, K. and Vateekul, P. Stock trend prediction using deep learning approach on technical indicator and industrial specific information. *Information*, 12(6), 2021. ISSN 2078-2489. doi: 10.3390/info12060250. URL https://www.mdpi.com/2078-2489/12/6/250.

Sakhare, N. N. and Imambi, D. S. S. A technical analysis based prediction of stock market trading strategies using deep learning and machine learning algorithms. *International Journal of Intelligent Systems and Applications in Engineering*, 10(3):411–422, Oct. 2022. URL https://ijisae.org/index.php/IJISAE/article/view/2182.

Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., and Soman, K. P. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1643–1647, 2017. doi: 10.1109/ICACCI.2017.8126078.

Totakura, V., V, D., and Sake, M. Prediction of stock trend for swing trades using long short-term memory neural network model. *International Journal of Scientific Technology Research*, Volume 9:1918–1923, 03 2020.

Yan, H. and Ouyang, H. Financial time series prediction based on deep learning. *Wireless Personal Communications*, 102(2):683–700, 09 2018. ISSN 1572-834X. doi: 10.1007/s11277-017-5086-2. URL https://doi.org/10.1007/s11277-017-5086-2.