International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181

Short-term Stock Market Price Trend Prediction using a Comprehensive Deep Learning System

Silpa Suneesh
MSc Data Science
University of Hertfordshire, United Kingdom

***Abstract*—In the age of big data, deep learning has gained increased traction for forecasting stock market prices and trends. I gathered two years' worth of data from the Chinese stock market and devised an extensive customization of feature engineering alongside a deep learning-based model to predict stock market price trends. My approach encompasses preprocessing the stock market dataset, employing various feature engineering techniques, and implementing a tailored deep learning system for accurate prediction of stock market trends. I conducted thorough assessments of commonly used machine learning models and determined that my approach excels, primarily due to the comprehensive feature engineering strategies I implemented. My system consistently achieves high accuracy in forecasting stock market trends. Through detailed examinations of prediction term lengths, feature engineering methodologies, and data preprocessing techniques, this study makes significant contributions to both financial and technical realms within the stock analysis research community.**

***Keywords*: Prediction, Deep learning, Stock market trend, Feature engineering.**

1. I. INTRODUCTION

The stock market is a primary area of focus for investors, making price trend prediction a perennially popular subject among researchers in financial and technical fields. This study aims to develop an advanced prediction model specifically designed for short-term price trend forecasting.

As noted by Fama in [26], predicting financial time series is widely recognized as challenging due to the semi-strong form of market efficiency and significant noise levels. In 2003, Wang et al. [44] employed artificial neural networks to predict stock market prices, focusing specifically on volume as a distinguishing feature. They discovered that volume did not significantly enhance forecasting accuracy in their studies using datasets such as S&P 500 and DJI. Ince and Trafalis [15] concentrated on short-term forecasting, applying support vector machine (SVM) models to predict stock prices. Their primary contribution involved comparing SVM with multi-layer perceptron (MLP) models, revealing that in many scenarios, SVM outperformed MLP, although outcomes were also influenced by various trading strategies. Concurrently, researchers in financial domains were exploring traditional statistical methods and signal processing techniques for analyzing stock market data.

Optimization techniques like principal component analysis (PCA) have also been utilized in forecasting short-term stock prices [22]. Over the years, researchers have expanded their focus beyond analyzing stock prices to include studying stock market transactions such as volume burst risks, thereby broadening the scope of stock market analysis research [39]. With advancements in artificial intelligence techniques, recent studies have increasingly integrated machine learning and deep learning methods based on earlier approaches, introducing new metrics as training features, as demonstrated by Liu and Wang [23]. These previous works fall under the domain of feature engineering and serve as the inspiration for extending features in my own research. Liu et al. [24] proposed a model incorporating both a convolutional neural network (CNN) and a long short-term memory (LSTM) neural network to evaluate various quantitative strategies in stock markets. The CNN is employed for stock selection by automatically extracting features from quantitative data, while the LSTM retains time-series characteristics to enhance profitability.

The most recent study also introduces a comparable hybrid neural network design, combining a convolutional neural network with a bidirectional long short-term memory network for forecasting the stock market index [4]. As researchers continue to suggest various neural network architectures, it sparks ongoing debate regarding whether the substantial training costs of these models justify their outcomes.

There are three key contributions of my work: (1) a new dataset that I extracted and cleansed, (2) a comprehensive approach to feature engineering, and (3) a custom-designed deep learning model based on Long Short-Term Memory (LSTM).

I created the dataset myself using an open-source data API called Tushare [43]. What sets my proposed solution apart is the emphasis on feature engineering combined with a finely-tuned system, rather than relying solely on an LSTM model. Drawing insights from previous research, I identified gaps and developed an architecture that includes a thorough feature engineering process before training the prediction model. By successfully extending features using recursive feature elimination algorithms, I enabled various machine learning algorithms to achieve high accuracy in predicting short-term price trends. This validated the effectiveness of my approach to feature engineering.

Additionally, I introduced a customized LSTM model that further enhanced prediction scores across all evaluation metrics. The proposed solution surpassed the performance of both traditional machine learning and deep learning models used in similar prior studies.

The rest of this paper is structured as follows. The "Survey of related works" section reviews previous research in the field. The "Dataset" section outlines the extraction process from public sources and details the dataset preparation. In the "Methods" section, I define the research problems, methods, and the design of my proposed solution, including technical algorithms and implementation details. The "Results" section

presents a thorough evaluation of our model, comparing it with models commonly used in related studies. The "Discussion" section analyzes and compares the results obtained. Finally, the "Conclusion" section summarizes the findings. This research paper draws heavily from Shen [36] as its foundation.

2. II. RELATED WORKS

In this section, I discuss related works. I reviewed the related work in two different domains: technical and financial, respectively.

Kim and Han in [19] built a model as a combination of artificial neural networks (ANN) and genetic algorithms (GAs) with discretization of features for predicting stock price index. The data used in their study include the technical indicators as well as the direction of change in the daily Korea stock price index (KOSPI). They used the data containing samples of 2928 trading days, ranging from January 1989 to December 1998, and give their selected features and formulas. They also applied optimization of feature discretization, as a technique that is similar to dimensionality reduction. The strengths of their work are that they introduced GA to optimize the ANN. First, the amount of input features and processing elements in the hidden layer are 12 and not adjustable. Another limitation is in the learning process of ANN, and the authors only focused on two factors in optimization. While they still believed that GA has great potential for feature discretization optimization. Our initialized feature pool refers to the selected features. Qiu and Song in [34] also presented a solution to predict the direction of the Japanese stock market based on an optimized artificial neural network model. In this work, authors utilize genetic algorithms together with artificial neural network based models, and name it as a hybrid GA-ANN model.

Piramuthu in [33] conducted a thorough evaluation of different feature selection methods for data mining applications. He used datasets, which were credit approval data, loan defaults data, web traffic data, tam, and kiang data, and compared how different feature selection methods optimized decision tree performance. The feature selection methods he compared included probabilistic distance measure: the Bhattacharyya measure, the Matusita measure, the divergence measure, the Mahalanobis distance measure, and the Patrick-Fisher measure. For inter-class distance measures: the Minkowski distance measure, city block distance measure, Euclidean distance measure, the Chebychev distance measure, and the nonlinear (Parzen and hyper-spherical kernel) distance measure. The strength of this paper is that the author evaluated both probabilistic distance-based and several inter-class feature selection methods. Besides, the author performed the evaluation based on different datasets, which reinforced the strength of this paper. However, the evaluation algorithm was a decision tree only. We cannot conclude if the feature selection methods will still perform the same on a larger dataset or a more complex model.

Hassan and Nath in [9] applied the Hidden Markov Model (HMM) on the stock market forecasting on stock prices of four different Airlines. They reduce states of the model into four states: the opening price, closing price, the highest price, and the lowest price. The strong point of this paper is that the approach does not need expert knowledge to build a prediction model. While this work is limited within the industry of Airlines and evaluated on a very small dataset, it may not lead to a prediction model with generality. One of the approaches in stock market prediction related works could be exploited to do the comparison work. The authors selected a maximum 2 years as the date range of training and testing dataset, which provided us a date range reference for our evaluation part.

Lei in [21] exploited Wavelet Neural Network (WNN) to predict stock price trends. The author also applied Rough Set (RS) for attribute reduction as an optimization. Rough Set was utilized to reduce the stock price trend feature dimensions. It was also used to determine the structure of the Wavelet Neural Network. The dataset of this work consists of five well-known stock market indices, i.e., (1) SSE Composite Index (China), (2) CSI 300 Index (China), (3) All Ordinaries Index (Australian), (4) Nikkei 225 Index (Japan), and (5) Dow Jones Index (USA). Evaluation of the model was based on different stock market indices, and the result was convincing with generality. By using Rough Set for optimizing the feature dimension before processing reduces the computational complexity. However, the author only stressed the parameter adjustment in the discussion part but did not specify the weakness of the model itself. Meanwhile, we also found that the evaluations were performed on indices, the same model may not have the same performance if applied on a specific stock.

Nekoeiqachkanloo et al. in [29] proposed a system with two different approaches for stock investment. The strengths of their proposed solution are obvious. First, it is a comprehensive system that consists of data pre-processing and two different algorithms to suggest the best investment portions. Second, the system is also embedded with a forecasting component, which also retains the features of the time series. Last but not least, their input features are a mix of fundamental features and technical indices that aim to fill in the gap between the financial domain and technical domain. However, their work has a weakness in the evaluation part. Instead of evaluating the proposed system on a large dataset, they chose 25 well-known stocks. There is a high possibility that the well-known stocks might potentially share some common hidden features.

As another related latest work, Idrees et al. [14] published a time series-based prediction approach for the volatility of the stock market. ARIMA is not a new approach in the time series prediction research domain. Their work is more focusing on the feature engineering side. Before feeding the features into ARIMA models, they designed three steps for feature engineering: Analyze the time series, identify if the time series is stationary or not, perform estimation by plot ACF and PACF charts and look for parameters. The only weakness of their proposed solution is that the authors did not perform any customization on the existing ARIMA model, which might limit the system performance to be improved.

A significant deficiency identified in previous studies is the limited development and utilization of data preprocessing methods. Technical studies typically prioritize constructing prediction models. When selecting features, researchers often compile a comprehensive list based on previous literature and apply feature selection algorithms to identify the most effective features. In contrast, research in investment domains has shown greater emphasis on behavioral analysis, such as examining how herd behavior impacts stock performance or assessing the influence of insider ownership on stock performance.

Analyzing these behaviors often requires preprocessing involving standard technical indices and expertise in investment practices to properly identify and understand their implications. In the related works, often a thorough statistical analysis is performed based on a special dataset and concludes new features rather than performing feature selections. Some data, such as the percentage of a certain index fluctuation has been proven to be effective on stock performance. I believe that by extracting new features from data, then combining such features with existing common technical indices will significantly benefit the existing and well-tested prediction models.

III. THE DATASET

This section details the data that was extracted from the public data sources, and the final dataset that was prepared. Stock market-related data are diverse, so first compare the related works from the survey of financial research works in stock market data analysis to specify the data collection directions. After collecting the data, define a data structure of the dataset. Given below, describes the dataset in detail, including the data structure, and data tables in each category of data with the segment definitions.

The dataset comprises 3,558 stocks from the Chinese stock market. In addition to daily price data and fundamental data for each stock ID, the dataset includes information on suspensions and resumptions, as well as details on the top 10 shareholders, among other factors. There are two primary reasons for selecting a 2-year timeframe for this dataset: (1) Most investors typically analyze stock market price trends using data from the past 2 years, and (2) Using more recent data enhances the accuracy of the analysis. I gathered this data using the open-source API Tushare [43], supplemented by web scraping techniques from sources such as Sina Finance and the SWS Research website.

IV. PROBLEM STATEMENT

I examined the optimal approach for predicting short-term price trends from multiple perspectives: feature engineering, financial domain expertise, and choice of prediction algorithm. I then addressed three key research questions in each area: How can feature engineering enhance prediction accuracy? How do insights from the financial domain inform the design of prediction models? And which algorithm proves most effective for predicting short-term price trends?

The initial research question concerns feature engineering and aims to understand how the selection method improves prediction model performance. Based on extensive prior research, it is evident that stock price data is inherently noisy and features often exhibit correlations, posing significant challenges for accurate prediction. This challenge underscores why the feature engineering component has been consistently introduced as an optimization module in most previous studies.

The second research question involves assessing the effectiveness of insights extracted from the financial domain. Unlike previous studies that primarily evaluate model performance metrics such as training costs and scores, my evaluation will focus on the impact of newly introduced financial domain features. I have incorporated specific findings from previous research, converting relevant raw data into actionable features. These features from the financial domain are then integrated with standard technical indices to identify those with the greatest influence through a voting mechanism. Given the multitude of potentially effective financial domain features, it is impractical for me to encompass all possibilities. Therefore, a critical aspect of my research involves determining how to effectively integrate these insights into the data processing framework of my system design.

The third research question focuses on selecting algorithms for modeling our data. Previous studies have predominantly concentrated on precise price prediction. In this paper, I approach the problem by breaking it down into predicting trends first, rather than exact values. This transforms the objective into solving a binary classification problem, while also addressing the challenge of mitigating noise interference effectively. My strategy involves decomposing the problem into manageable subtasks with fewer interdependencies, tackling each sequentially, and then integrating these solutions into an ensemble model to aid investment decision-making.

Historically, researchers have employed various models to forecast stock price trends, with machine learning techniques often yielding the best results. In this study, I will compare my approach with these top-performing machine learning models during the evaluation phase to address this research question effectively.

V. PROPOSED SOLUTION

The overarching structure of my proposed solution can be divided into three main components. Firstly, there is the feature selection phase, where I ensure that the chosen features are highly impactful. Secondly, I conduct an analysis of the data and implement dimensionality reduction techniques. Lastly, the primary focus of my work lies in developing a prediction model for target stocks. Figure 1 depicts a high-level architecture of the proposed solution.

There are various ways to categorize stocks based on investor preferences, such as long-term versus short-term investments. It's not uncommon for stock reports to indicate average performance while the stock price experiences significant increases, highlighting the unpredictability in stock price prediction. Therefore, identifying effective features before training a model with data becomes crucial.

In this study, the focus is on predicting short-term price trends. Initially, I have raw data without labels. Therefore, the first step involves labeling the data. This is done by comparing the current closing price with the closing price from n trading days ago, where n ranges from 1 to 10, reflecting the short-term nature of our research. If the price trend increases, it is labeled as 1; otherwise, it is labeled as 0. Specifically, I use indices from the (n-1)th day to predict the price trend for the nth day.

Based on previous research, certain scholars integrated financial domain expertise and technical methods to filter high-quality stocks using specific rules. Drawing inspiration from their studies, I incorporated these rules into my feature extension design.

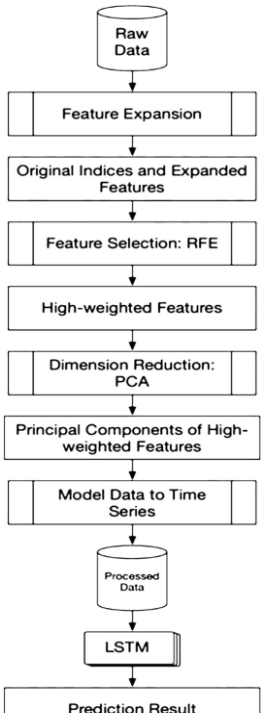

Figure 1: High Level Proposed System

However, to ensure optimal performance of the prediction model, I will begin by examining the data. The raw data contains a large number of features, and considering all of them would not only increase computational complexity significantly but could also complicate future unsupervised learning efforts. Therefore, I employ recursive feature elimination (RFE) to ensure that only the most effective features are selected.

Most previous studies in the technical domain analyzed all stocks broadly, whereas in the financial domain, researchers often focused on specific investment scenarios. To bridge this gap between the two domains, I incorporate a feature extension based on insights gathered from the financial domain before initiating the RFE procedure.

Given our intention to model the data as time series, a higher number of features would increase the complexity of training procedures. Thus, I plan to mitigate this by employing randomized PCA for dimensionality reduction at the outset of my proposed solution architecture.

Detailed technical design elaboration

This section elaborates on the detailed technical design, presenting a comprehensive solution that integrates, combines, and customizes various existing techniques in data preprocessing, feature engineering, and deep learning. Figure 2 outlines the detailed technical design from data processing to prediction, encompassing data exploration as well. The content is organized into main procedures, each consisting of algorithmic steps. Detailed algorithmic information will be provided in the subsequent section. Here, the focus is on illustrating the workflow of the data.

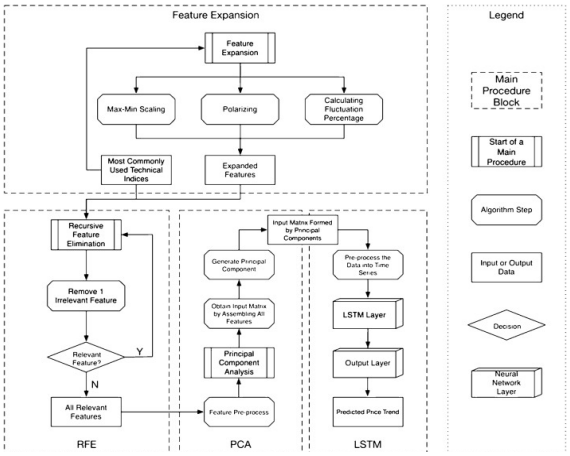

Figure 2 : Detailed technical design of the proposed solution

Based on the literature review, I have chosen the most commonly utilized technical indices and incorporated them into the feature extension process to expand the feature set. From this expanded set, I will select the most effective i features. Subsequently, I will apply the PCA algorithm to reduce the dimensionality of the data to j features. Once I determine the optimal combination of i and j, I will finalize the feature set and input it into the LSTM [10] model to predict price trends.

The novelty of my approach lies in not only applying technical methods to raw data but also implementing feature extensions commonly used by stock market investors. Detailed explanations of the feature extension process are provided in the following subsection. Insights gained from previous studies on optimizing deep learning solutions [37, 38] were instrumental in designing and customizing the feature engineering and deep learning components of this work.

Applying feature extension is the initial main procedure depicted in Fig. 2. Here, the input data consists of the most commonly utilized technical indices identified from related studies. The process involves three feature extension methods: max-min scaling, polarizing, and calculating fluctuation percentage. Not all technical indices are suitable for all three extension methods; rather, this procedure selectively applies appropriate extension methods based on the nature of each index. The selection of these methods is determined by examining the calculation methodologies of the indices themselves.

After the feature extension procedure, the expanded features will be combined with the most commonly used technical indices, i.e., input data with output data, and feed into RFE block as input data in the next step.

Applying recursive feature elimination (RFE) follows the feature extension phase mentioned earlier. Using the Recursive Feature Elimination (RFE) algorithm [6], I identify the most effective i features based on their coefficients and feature importance scores. During this process, I progressively remove one feature at each step while retaining those deemed relevant. The output from the RFE phase serves as the input for the subsequent step involving PCA.

Applying principal component analysis (PCA) begins with feature preprocessing as the initial step. Due to the diverse

nature of features post-RFE—some being percentage data and others large numerical values—there exists a disparity in their units. This variance can influence the outcome of principal component extraction. Therefore, prior to inputting data into the PCA algorithm [8], it is essential to conduct feature preprocessing to standardize the data. In the "Results" section, I provide an illustration of the effectiveness and compare methods used in this process.

Prior to utilizing principal component analysis (PCA), it is crucial to conduct feature pre-processing. This step is necessary because the features obtained from RFE include both percentage data and large numerical values, which are measured in different units. This variance in scales can impact the results of PCA's principal component extraction. Therefore, I perform feature preprocessing to standardize the data before feeding it into the PCA algorithm [8]. The effectiveness of this approach and a comparison of methods are further detailed in the "Results" section.

Following feature pre-processing, the next step involves inputting the processed data, containing the selected i features, into the PCA algorithm to reduce the dimensionality of the feature matrix to j features. This step aims to retain the most effective features while reducing computational complexity during model training. This study also evaluates the optimal combination of i and j that achieves higher prediction accuracy while minimizing computational resources, as detailed in the "Results" section. After PCA, the system obtains a reshaped matrix with j columns.

Fitting the long short-term memory (LSTM) model requires additional data pre-processing after PCA has reduced the dimensions of the input data. This pre-processing step is essential because the matrix formed by principal components lacks time steps, a crucial parameter for training LSTM models. Therefore, the data must be reshaped into corresponding time steps for both the training and testing datasets.

After completing the data pre-processing phase, the final step involves feeding the training data into the LSTM model and evaluating its performance using testing data. As a variant of recurrent neural networks (RNNs), an LSTM model is structured as a deep neural network, even with a single LSTM layer, capable of processing sequential data and retaining information in its hidden states over time. Each LSTM layer consists of LSTM units, which include cells and gates designed for classifying and predicting based on time series data.

The LSTM architecture in this study comprises two layers. The input dimension is defined by j, determined after applying the PCA algorithm. The first layer serves as the input LSTM layer, while the second layer functions as the output layer. The final output of the model will be a binary prediction (0 or 1), indicating whether the predicted stock price trend suggests a decrease or increase. This output serves as valuable guidance for investors in making informed investment decisions.

Design discussion:

Feature extension represents a significant innovation in our proposed system for predicting price trends. This process involves integrating technical indices with heuristic processing methods derived from investor practices, thereby bridging the gap between financial and technical research areas.

Given the focus on predicting price trends, feature engineering plays a pivotal role in determining the final prediction outcomes. The feature extension method ensures that potentially correlated features are not overlooked, while the feature selection process is essential for identifying and incorporating effective features. Including irrelevant features introduces noise into the model, underscoring the importance of each main procedure in contributing to the overall system design.

In addition to feature engineering, our approach integrates LSTM, a cutting-edge deep learning technique renowned for its efficacy in time-series prediction. LSTM models are adept at capturing intricate hidden patterns and temporal relationships, enhancing the prediction capabilities of our model.

Deep learning models, however, incur high training costs in terms of both time and hardware resources. Therefore, another advantage of our system design is the integration of PCA as an optimization procedure. PCA effectively reduces the dimensionality of the feature matrix while retaining its principal components. This reduction helps mitigate the training costs associated with processing large-scale time-series data matrices, thereby optimizing overall system performance.

Algorithm 1: Short-term stock market price trend prediction—applying feature engineering using FE + RFE + PCA

Algorithm 1

Algorithm 1: Short-term Stock Market Price Trend Prediction - Feature Engineering using FE + RFE + PCA

```
function FE(df)
  # Apply only the meaningful methods on data
  df_expandedfeatures  = Max-MinScaling(df)
  df_expandedfeatures  = Polarizing(df)
  df_expandedfeatures  = CalcFluctuationPercentage(df)
  return df_expandedfeatures
end function

function RFE(df)  # (Utilizing Recursive Feature Elimination function)
  Train the model on all the features of the training dataset in df
  Calculate performance of the model with samples from the test data
  Rank the weights of different features based on testing the model
  for each subset do
    Retain i most weighted features
    Train the model on all the features of the training dataset
    Calculate performance of the model with samples from the test data
  end for
  Calculate the overall performance profile for each feature over samples from the test data
  Rank and select top ranked features
  Train the model on the selected features using the training dataset in df
  return df_RFE   # (df_RFE is the processed data frame after RFE algorithm)
end function

function PCA (df)   # (Utilizing PCA to reduce dimension from i to j)
   df_PCA = applyPCA (n_components=j, whiten=False, copy=True, batchsize = 200)
   return df_PCA   # df_PCA is the optimized data frame after applying PCA algorithm)
end function

function MAIN()                # (Main function)
  df_alldata = load data
  df_partition = DataPartition(df_alldata, method = resampling)
  df_FE = FE(df_partition)
  df_RFE = RFE(df_FE)
  df_PCA = PCA(df_RFE)
  return df_PCA
```

Algorithm 2: Price trend prediction model using LSTM

Algorithm 2

Algorithm 2: Price Trend Prediction Model using LSTM

```
function TimeSeriesConversion (df, term_length, lag)
   # Utilizing time series conversion technique to convert the training data matrix, after applying PCA
from Algorithm 1, to time series
   cols = list()
   for i in range (term_length, 0, -1) do                    # Input sequence
      shift df by i
      append shifted df to cols
   end for
   for i in range (0, lag) do                                # Forecast sequence
      shift df by -1
      append shifted df to cols
   end for
   df_TS = concat(cols, axis = 1)                            # Put all sequences together
   return df_TS
end function

function ModelCompile()                                      # Applying LSTM model with given structure
and compiling it
   Stack_method = Sequential()
   Layer_1 = LSTM(50, input _shape=(train_X.shape[1], train_X.shape[2]))
   Layer_2 = Dense(1)
   Loss_Function=mae
   Optimizer=adam
   Metrics=f1, metrics.binary_accuracy, metrics.mean_squared_error, metrics.mean_absolute_error
   return LSTMmodel
end function

function MAIN()                                              # Main Function
   df_TS = TimeSeriesConversion(df_PCA, N_TIME_STEPS, LAG)
   DataPartition(df_TS, method = resampling)
   ModelCompile(j)
   FitModel(X, y, epochs=50, batch_size=3000)                # Train and fit the model
   EvaluateModel(X_test, y_test)                             # Calculate evaluation metrics on the trained
model using test data
```

VI. RESULT

To comprehensively evaluate our algorithm design, I structured the evaluation process based on the main procedures and assessed how each procedure impacts the performance of the algorithm. Initially, I conducted evaluations on two different computing environments: a machine with a 2.2 GHz i7 processor and 16 GB of RAM, and an Amazon EC2 instance equipped with a 3.1 GHz Processor, 16 vCPUs, and 64 GB of RAM.

In the implementation phase, I expanded 20 initial features into 54 features, subsequently retaining the 30 most effective ones. This section focuses on evaluating the feature selection process. The dataset was divided into two distinct subsets: the training dataset and the testing dataset. The testing procedure was divided into two parts: one subset, DS_test_f, was used for feature selection, while another subset, DS_test_m, served for model testing.

For the feature selection phase, two-thirds of the stock data were randomly selected by stock ID and labeled as DS_train_f. This dataset contained complete technical indices and the expanded features spanning throughout 2018. The Recursive Feature Elimination (RFE) algorithm was employed with Support Vector Regression (SVR) using linear kernels as the estimator. Features were ranked through voting, and the top 30 effective features were selected. These features were then processed using Principal Component Analysis (PCA) to reduce dimensionality, resulting in 20 principal components.

The remaining stock data constituted the testing dataset, DS_test_f, used to validate the effectiveness of the principal components extracted from the selected features. All data from 2018 were consolidated into the training dataset of the prediction model, noted as DS_train_m. The model testing dataset, DS_test_m, encompassed the first three months of 2019 data, ensuring no overlap with the data used in previous steps to prevent overfitting.

This approach ensures a rigorous evaluation of the algorithm's performance across different computational setups and verifies the effectiveness of each procedural step in enhancing prediction accuracy while managing computational efficiency.

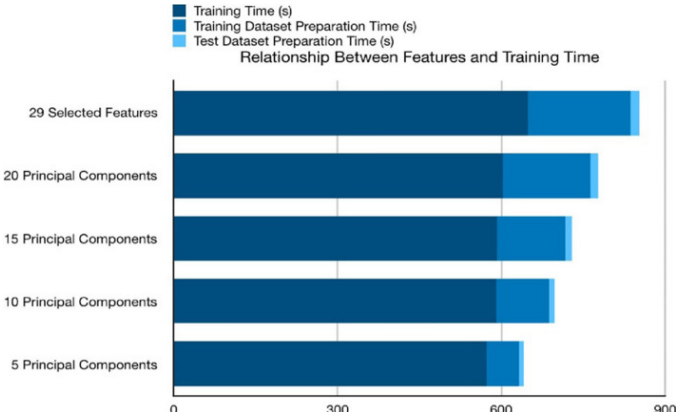

Figure 3 : Relationship between feature number and training time

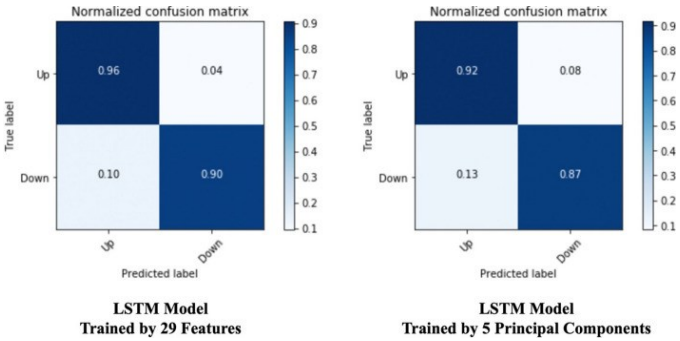

Figure 4:Proposed model prediction precision comparison—confusion matrices

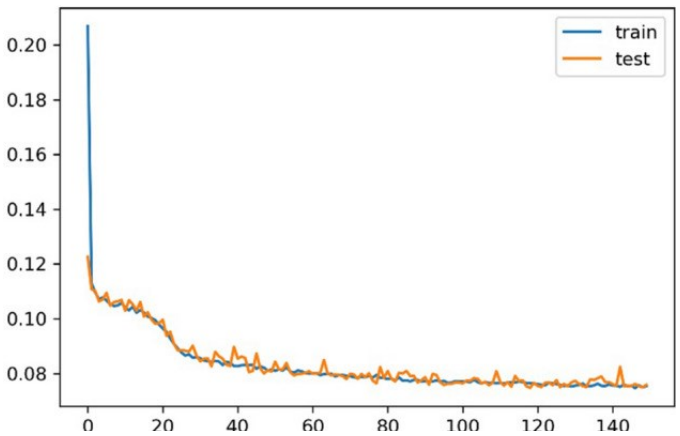

Figure 5 : Learning curve of proposed solution

VII. CONCLUSION

This study comprises three main components: the extraction and pre-processing of a dataset from the Chinese stock market, the development of feature engineering techniques, and the implementation of a stock price trend prediction model using Long Short-Term Memory (LSTM). Initially, I gathered, cleaned, and organized two years of data from the Chinese stock

market. I explored various techniques commonly employed by practitioners, introducing a novel algorithmic component named feature extension, which has proven effective.

The feature extension (FE) involved integrating recursive feature elimination (RFE) and principal component analysis (PCA) to construct a robust feature engineering framework that balances effectiveness and efficiency. This system was tailored by integrating the feature engineering process with an LSTM-based prediction model, achieving high prediction accuracy surpassing leading models in similar studies. A comprehensive evaluation of the approach was conducted, comparing our LSTM model with frequently used machine learning models in the context of feature engineering. This comparison yielded heuristic insights that pose potential future research questions in both technical and financial research domains.

The proposed solution represents a unique customization compared to previous approaches. Instead of merely introducing another state-of-the-art LSTM model, I have developed a finely-tuned and customized deep learning prediction system. This system integrates comprehensive feature engineering techniques with LSTM to enhance prediction accuracy. Drawing insights from previous studies, I have bridged the gap between investor needs and academic research by introducing a feature extension algorithm prior to recursive feature elimination, resulting in significant improvements in model performance.

While my current research has yielded promising results, there remains substantial potential for future exploration. Through my evaluation process, I observed that the effectiveness of the RFE algorithm varies significantly with different term lengths, particularly favoring 2-day, weekly, and biweekly intervals. Exploring how technical indices influence less conventional term lengths could be a promising avenue for future research.

Additionally, there is considerable potential in integrating advanced sentiment analysis techniques with feature engineering and deep learning models. This approach could lead to the development of a more comprehensive prediction system that incorporates diverse sources of information such as social media posts, news articles, and other text-based data. Such an integrated system could potentially enhance predictive accuracy by capturing broader market sentiments and trends.

1. Abbreviations and Acronyms

- LSTM Long short term memory
- PCA Principal component analysis
- RNN Recurrent neural networks
- ANN Artificial neural network
- DNN Deep neural network
- DTW Dynamic Time Warping
- RFE Recursive feature elimination
- SVM Support vector machine
- CNN Convolutional neural network
- SGD Stochastic gradient descent
- ReLU Rectified linear unit
- MLP Multi layer perceptron.

VIII. REFERENCES

[1] Atsalakis GS, Valavanis KP. Forecasting stock market short-term trends using a neuro-fuzzy based methodology. Expert Syst Appl. 2009;36(7):10696–10707. doi: 10.1016/j.eswa.2009.02.043. [CrossRef] [Google Scholar]

[2] Ayo CK. Stock price prediction using the ARIMA model. In: 2014 UKSim-AMSS 16th international conference on computer modelling and simulation. 2014. 10.1109/UKSim.2014.67.

[3] Brownlee J. Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery. 2018. https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/

[4] Eapen J, Bein D, Verma A. Novel deep learning model with CNN and bi-directional LSTM for improved stock market index prediction. In: 2019 IEEE 9th annual computing and communication workshop and conference (CCWC). 2019. pp. 264–70. 10.1109/CCWC.2019.8666592.

[5] Fischer T, Krauss C. Deep learning with long short-term memory networks for financial market predictions. Eur J Oper Res. 2018;270(2):654–669. doi: 10.1016/j.ejor.2017.11.054. [CrossRef] [Google Scholar]

[6] Guyon I, Weston J, Barnhill S, Vapnik V. Gene selection for cancer classification using support vector machines. Mach Learn. 2002;46:389–422. doi: 10.1023/A:1012487302797. [CrossRef] [Google Scholar]

[7] Hafezi R, Shahrabi J, Hadavandi E. A bat-neural network multi-agent system (BNNMAS) for stock price prediction: case study of DAX stock price. Appl Soft Comput J. 2015;29:196–210. doi: 10.1016/j.asoc.2014.12.028. [CrossRef] [Google Scholar]

[8] Halko N, Martinsson PG, Tropp JA. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. 2001;53(2):217–288. doi: 10.1137/090771806. [CrossRef] [Google Scholar]

[9] Hassan MR, Nath B. Stock market forecasting using Hidden Markov Model: a new approach. In: Proceedings—5th international conference on intelligent systems design and applications 2005, ISDA'05. 2005. pp. 192–6. 10.1109/ISDA.2005.85.

[10]Hochreiter S, Schmidhuber J. Long short-term memory. J Neural Comput. 1997;9(8):1735–1780. doi: 10.1162/neco.1997.9.8.1735. [PubMed] [CrossRef] [Google Scholar]

[11]Hsu CM. A hybrid procedure with feature selection for resolving stock/futures price forecasting problems. Neural Comput Appl. 2013;22(3–4):651–671. doi: 10.1007/s00521-011-0721-4. [CrossRef] [Google Scholar]

[12] Huang CF, Chang BR, Cheng DW, Chang CH. Feature selection and parameter optimization of a fuzzy-based stock selection model using genetic algorithms. Int J Fuzzy Syst. 2012;14(1):65–75. doi: 10.1016/J.POLYMER.2016.08.021. [CrossRef] [Google Scholar]

[13] Huang CL, Tsai CY. A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. Expert Syst Appl. 2009;36(2 PART 1):1529–1539. doi: 10.1016/j.eswa.2007.11.062. [CrossRef] [Google Scholar]

[14]. Idrees SM, Alam MA, Agarwal P. A prediction approach for stock market volatility based on time series data. IEEE Access. 2019;7:17287–17298. doi: 10.1109/ACCESS.2019.2895252. [CrossRef] [Google Scholar]

[15] Ince H, Trafalis TB. Short term forecasting with support vector machines and application to stock price prediction. Int J Gen Syst. 2008;37:677–687. doi: 10.1080/03081070601068595. [CrossRef] [Google Scholar]

[16]. Jeon S, Hong B, Chang V. Pattern graph tracking-based stock price prediction using big data. Future Gener Comput Syst. 2018;80:171–187. doi: 10.1016/j.future.2017.02.010. [CrossRef] [Google Scholar]

[17]. Kara Y, Acar Boyacioglu M, Baykan ÖK. Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul Stock Exchange. Expert Syst Appl. 2011;38(5):5311–5319. doi: 10.1016/j.eswa.2010.10.027. [CrossRef] [Google Scholar]

[18] Khaidem L, Dey SR. Predicting the direction of stock market prices using random forest. 2016. pp. 1–20.

[19] Kim K, Han I. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. Expert Syst Appl. 2000;19:125–132. doi: 10.1016/S0957-4174(00)00027-0. [CrossRef] [Google Scholar]

[20]Lee MC. Using support vector machine with a hybrid feature selection method to the stock trend prediction. Expert Syst Appl. 2009;36(8):10896–10904. doi: 10.1016/j.eswa.2009.02.038. [CrossRef] [Google Scholar]

[21]Lei L. Wavelet neural network prediction method of stock price trend based on rough set attribute reduction. Appl Soft Comput J. 2018;62:923–932. doi: 10.1016/j.asoc.2017.09.029. [CrossRef] [Google Scholar]

[22] Lin X, Yang Z, Song Y. Expert systems with applications short-term stock price prediction based on echo state networks. Expert Syst Appl. 2009;36(3):7313–7317. doi: 10.1016/j.eswa.2008.09.049. [CrossRef] [Google Scholar]

[23]Liu G, Wang X. A new metric for individual stock trend prediction. Eng Appl Artif Intell. 2019;82(March):1–12. doi: 10.1016/j.engappai.2019.03.019. [CrossRef] [Google Scholar]

[24]Liu S, Zhang C, Ma J. CNN-LSTM neural network model for quantitative strategy analysis in stock markets. 2017;1:198–206. 10.1007/978-3-319-70096-0.

[25]Long W, Lu Z, Cui L. Deep learning-based feature engineering for stock price movement prediction. Knowl Based Syst. 2018;164:163–173. doi: 10.1016/j.knosys.2018.10.034. [CrossRef] [Google Scholar]

[26]Malkiel BG, Fama EF. Efficient capital markets: a review of theory and empirical work. J Finance. 1970;25(2):383–417. doi: 10.1111/j.1540-6261.1970.tb00518.x. [CrossRef] [Google Scholar]

[27]McNally S, Roche J, Caton S. Predicting the price of bitcoin using machine learning. In: Proceedings—26th Euromicro international conference on parallel, distributed, and network-based processing, PDP 2018. pp. 339–43. 10.1109/PDP2018.2018.00060. [PMC free article] [PubMed]

[28]Nagar A, Hahsler M. News sentiment analysis using R to predict stock market trends. 2012. http://past.rinfinance.com/agenda/2012/talk/Nagar+Hahsler.pdf. Accessed 20 July 2019.

[29]Nekoeiqachkanloo H, Ghojogh B, Pasand AS, Crowley M. Artificial counselor system for stock investment. 2019. ArXiv Preprint arXiv:1903.00955.

[30]Ni LP, Ni ZW, Gao YZ. Stock trend prediction based on fractal feature selection and support vector machine. Expert Syst Appl. 2011;38(5):5569–5576. doi: 10.1016/j.eswa.2010.10.079. [CrossRef] [Google Scholar]

[31] Pang X, Zhou Y, Wang P, Lin W, Chang V. An innovative neural network approach for stock market prediction. J Supercomput. 2018 doi: 10.1007/s11227-017-2228-y. [CrossRef] [Google Scholar]

[32]Pimenta A, Nametala CAL, Guimarães FG, Carrano EG. An automated investing method for stock market based on multiobjective genetic programming. Comput Econ. 2018;52(1):125–144. doi: 10.1007/s10614-017-9665-9. [CrossRef] [Google Scholar]

[33]Piramuthu S. Evaluating feature selection methods for learning in data mining applications. Eur J Oper Res. 2004;156(2):483–494. doi: 10.1016/S0377-2217(02)00911-6. [CrossRef] [Google Scholar]

[34]Qiu M, Song Y. Predicting the direction of stock market index movement using an optimized artificial neural network model. PLoS ONE. 2016;11(5):e0155133. doi: 10.1371/journal.pone.0155133. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

[35]Scikit-learn. Scikit-learn Min-Max Scaler. 2019. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html. Retrieved 26 July 2020.

[36]Shen J. Thesis, "Short-term stock market price trend prediction using a customized deep learning system", supervised by M. Omair Shafiq, Carleton University. 2019. [PMC free article] [PubMed]

[37]Shen J, Shafiq MO. Deep learning convolutional neural networks with dropout—a parallel approach. ICMLA. 2018;2018:572–577. [Google Scholar]

[38]Shen J, Shafiq MO. Learning mobile application usage—a deep learning approach. ICMLA. 2019;2019:287–292. [Google Scholar]

[39]Shih D. A study of early warning system in volume burst risk assessment of stock with Big Data platform. In: 2019 IEEE 4th international conference on cloud computing and big data analysis (ICCCBDA). 2019. pp. 244–8.

[40] Sirignano J, Cont R. Universal features of price formation in financial markets: perspectives from deep learning. Ssrn. 2018 doi: 10.2139/ssrn.3141294. [CrossRef] [Google Scholar]

[41]Thakur M, Kumar D. A hybrid financial trading support system using multi-category classifiers and random forest. Appl Soft Comput J. 2018;67:337–349. doi: 10.1016/j.asoc.2018.03.006. [CrossRef] [Google Scholar]

[42] Tsai CF, Hsiao YC. Combining multiple feature selection methods for stock prediction: union, intersection, and multi-intersection approaches. Decis Support Syst. 2010;50(1):258–269. doi: 10.1016/j.dss.2010.08.028. [CrossRef] [Google Scholar]

[43]Tushare API. 2018. https://github.com/waditu/tushare. Accessed 1 July 2019.

[44]Wang X, Lin W. Stock market prediction using neural networks: does trading volume help in short-term prediction?. n.d.

[45] Weng B, Lu L, Wang X, Megahed FM, Martinez W. Predicting short-term stock prices using ensemble methods and online data sources. Expert Syst Appl. 2018;112:258–273. doi: 10.1016/j.eswa.2018.06.016. [CrossRef] [Google Scholar]

[46] Zhang S. Architectural complexity measures of recurrent neural networks, (NIPS). 2016. pp. 1–9.

[47]. Zubair M, Fazal A, Fazal R, Kundi M. Development of stock market trend prediction system using multiple regression. Computational and mathematical organization theory. Berlin: Springer US; 2019. [Google Scholar]

[48]

[49]Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[50]M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.