Proceedings of the 14th Asian Control Conference
July 5-8, 2024, Dalian, China

Prediction of Yield in Functional Testing of Motherboards in Laptop Manufacturing

1st Yunbo Zhao
Department of Automation and Institute of Advanced Technology University of Science and Technology of China Hefei, China
Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
Hefei, China
ybzhao@ustc.edu.cn

2nd Shaojie Dong
Institute of Advanced Technology University of Science and Technology of China Hefei, China
shaojiedong@mail.ustc.edu.cn

3rd Yu Kang
Department of Automation and Institute of Advanced Technology University of Science and Technology of China Hefei, China
Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
Hefei, China
kangduyu@ustc.edu.cn

4th Kangcheng Wang
Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
Hefei, China
kcwang@iai.ustc.edu.cn

5th Longxin Chen
Department of Automation University of Science and Technology of China Hefei, China
lxchen@mail.ustc.edu.cn

6th Peng Bai
Department of Automation University of Science and Technology of China Hefei, China
baipeng@lenovo.com

***Abstract*—Functional testing stands as a pivotal quality control step in the production process of laptop motherboards, aiming to validate the proper functioning of various components. However, due to the multitude of functional modules involved on the motherboard, testing all of them requires a significant amount of time and resources. As a result, production line engineers often rely on empirical selection of modules with low yield rates for testing. However, such empirical yield estimation is often inaccurate. To address this challenge, this study proposes a hybrid model based on XGBoost and Long Short-Term Memory (LSTM) networks to predict the yield of each functional module. By harnessing the feature learning capability of XGBoost and the sequential modeling power of LSTM, this model efficiently explores the intricate correlations among motherboard functional modules, thereby accurately forecasting their yields. We extensively train and validate the model using historical production data and successfully deploy it on real laptop motherboard production lines. Experimental results demonstrate that our hybrid model accurately predicts the yield of each functional module, providing crucial guidance for the functional testing process. Through in-depth analysis of the predicted yield results, engineers can systematically choose testing projects to save time and resources. This research offers a novel approach and pathway for enhancing motherboard production efficiency and quality.**

***Index Terms*—Functional testing, Laptop motherboard manufacturing, Time series prediction**

I. Introduction

Functional testing represents a pivotal aspect of quality control in laptop manufacturing [1]–[4], as depicted in “Fig. 1”, encompassing two primary stages [5]: first, the production of motherboards through Surface Mount Technology (SMT) on printed circuit boards (PCBs), followed by functional testing and repair of defective components; second, rigorous quality control requirements for finished laptops, necessitating functional testing and repair of defective units. This study focuses on a typical laptop manufacturing factory, referred to as ”Factory X,” one of the world’s largest facilities. Testing of typical laptop motherboards at Factory X involves over 30 different test items, consuming considerable time, making it impractical for the production line. Thus, Factory X adopts a selective testing approach, wherein a subset of potentially faulty functional modules is tested from the pool of over 30 test items. Due to data confidentiality, access is restricted to internal technical staff at Factory X and a few academic collaborators. Consequently, academic exploration of this issue is limited, resulting in Factory X’s selective testing strategy relying solely on internal engineers’ expertise, lacking theoretical underpinnings. This paper aims to provide theoretical foundations for the selection of test items during functional testing by accurately predicting the yield of each functional module through the construction of a time series prediction model.

In time series forecasting algorithms, traditional predictive models include the Vector Autoregression (VAR) model, Autoregressive Moving Average (ARMA) model, Support Vector Regression (SVR) model, among others. Although they find application across various tasks [6]–[8], they are primarily suitable for univariate time series and unable to handle the complex nonlinear relationships within multivariate time series data. With the development of deep learning, Deep Neural Networks (DNNs) have been widely applied to time series

This work was supported by the National Natural Science Foundation of China (No. 62173317), and the Key Research and Development Program of Anhui (No. 202104a05020064). Corresponding author: Yunbo Zhao (e-mail: ybzhao@ustc.edu.cn).

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 10:40:30 UTC from IEEE Xplore. Restrictions apply.

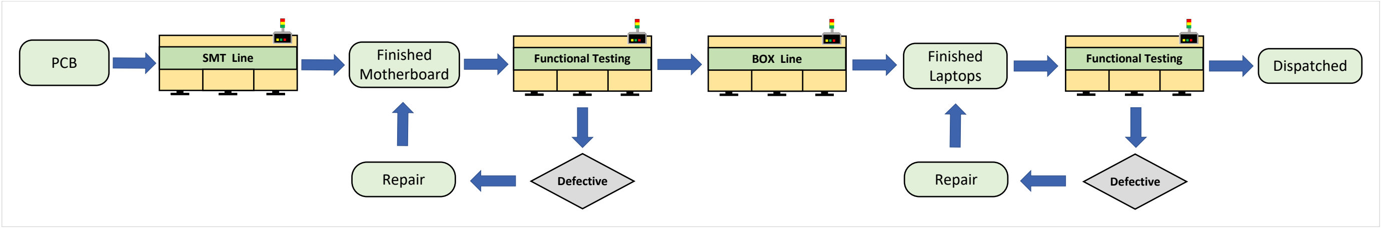

Fig. 1. The process of functional testing in laptop manufacturing.

forecasting problems [9]–[11]. Compared to classical models, predictive tasks can be more effectively accomplished through DNN-based models. However, they still face challenges in predicting the complex spatial characteristics prevalent in industrial domains, and pure deep learning models suffer from poor interpretability, making them unsuitable for industrial applications.The prediction of module yields faces two main challenges [12]: firstly, the time series of module yields exhibit significant fluctuations, making it difficult to capture temporal patterns of module failures; secondly, there exists coupling between various functional modules, leading to mutual influence and interference among modules, complicating the extraction of coupling features.To address these challenges, we propose a hybrid model combining XGBoost and LSTM for yield prediction. Specifically, the Long Short-Term Memory (LSTM) [13] network captures time series information using gate units, circumventing the vanishing gradient problem. However, LSTM networks struggle to extract inter-sequence correlation information and cannot output variable importance. Therefore, we introduce the statistical model XGBoost [14] to extract inter-sequence correlation features. Based on the aforementioned analysis, this paper presents the XGB-LSTM model to tackle the yield prediction problem of motherboard functional modules.

The organization of this paper is as follows: Section 2 introduces the relevant knowledge of XGBoost and LSTM; Section 3 elaborates on the proposed XGB-LSTM model; Section 4 conducts experimental validation and result analysis of the proposed model; Section 5 concludes the paper.

II. Theoretical Background

A. LSTM

Long Short-Term Memory (LSTM) is a variant of Recurrent Neural Networks (RNNs) [15] designed specifically for handling sequential data, capable of effectively capturing long-term dependencies within sequences. Compared to traditional RNNs, LSTM introduces a mechanism called gated units, which include input gates, forget gates, and output gates, controlling the flow of information to finely regulate sequential information.The specific structure of LSTM is shown in “Fig. 2”.

In LSTM, the update process of the memory cell c_t is described by the following formula:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{1}$$

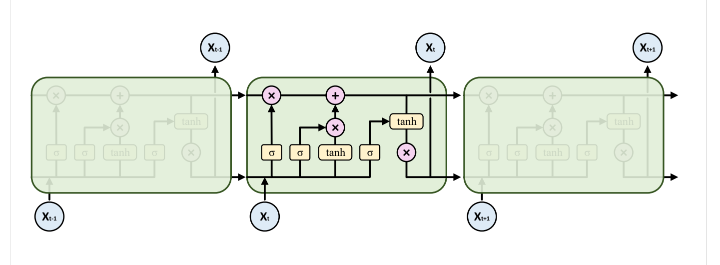

Fig. 2. LSTM structure diagram.

where c_t is the memory cell at time step t, f_t is the output of the forget gate, i_t is the output of the input gate, and $\tilde{c}_t$ is the update value of the candidate memory cell. The calculation of the forget gate and the input gate is as follows:

The calculation formula of the forget gate f_t is:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$

The calculation formula of the input gate i_t is:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

where W_f and W_i are the weight matrices of the forget gate and the input gate, b_f and b_i are the corresponding bias terms. Through the gated mechanisms above, LSTM networks can selectively remember or forget past information when processing sequential data, thereby better capturing long-term dependencies within sequences, and improving the model's expressive power and generalization performance.

During training, LSTM networks utilize the backpropagation through time (BPTT) algorithm [16] to update their parameters and minimize the loss function. Given a sequence of input-output pairs $(x_1, y_1), (x_2, y_2), ..., (x_T, y_T)$, where x_t represents the input at time step t and y_t represents the corresponding target output, the network's parameters are iteratively updated using gradient descent.

The loss function used for training LSTM networks typically involves comparing the predicted output $\hat{y}_t$ at each time step with the actual target output y_t. The gradients of the loss function with respect to the parameters are computed using the chain rule of calculus and backpropagated through time. This allows the gradients to be used to update the parameters in the direction that minimizes the loss function.

By adjusting the parameters iteratively based on the gradients of the loss function, LSTM networks learn to better model the sequential dependencies in the training data, ultimately improving their performance on unseen data during inference.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 10:40:30 UTC from IEEE Xplore. Restrictions apply.

B. XGBoost

XGBoost (eXtreme Gradient Boosting) is an ensemble learning algorithm based on Gradient Boosting Decision Trees [17], aimed at solving regression and classification problems. XGBoost sequentially trains a series of decision tree models and gradually corrects their prediction errors to improve the model's performance. The specific structure of XGBoost is shown in Figure 3.

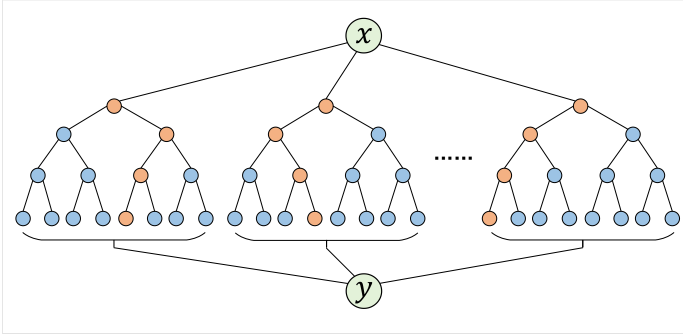

Fig. 3. XGBoost structure diagram.

In XGBoost, each decision tree T_i can be represented as a function $f_i(x)$, where x is the input feature vector. By combining multiple decision trees, the prediction output of the entire model can be obtained as follows:

$$\hat{y} = \sum_{i=1}^{n} f_i(x) \tag{4}$$

The key innovation of XGBoost lies in the definition and optimization process of the objective function. Its objective function consists of two parts: the loss function L and the regularization term R, which can be represented as:

$$\text{Obj} = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{i=1}^{n} \Omega(f_i) \tag{5}$$

where y_i is the true label value, $\hat{y}_i$ is the model's prediction value, and n is the number of samples. The loss function L measures the error between the predicted value and the true value, while the regularization term R controls the complexity of the model to prevent overfitting. The regularization term can be represented by the following equation:

$$\Omega(f_i) = \gamma(T) + \frac{1}{2}\lambda\|w\|_2^2 \tag{6}$$

where T is the number of leaf nodes, γ and λ are regularization parameters, and w is the weight of the leaf nodes. The first term $\gamma(T)$ controls the number of leaf nodes in the tree, and the second term $\frac{1}{2}\lambda\|w\|_2^2$ is the L2 regularization term, which penalizes the size of the leaf node weights.

XGBoost optimizes the objective function using the gradient boosting algorithm, which iteratively trains new decision trees and updates parameters based on the negative gradient of the objective function. In each iteration, a new decision tree is fitted using gradient information to minimize the objective function. Through this process, XGBoost can effectively learn complex patterns in the data and achieve outstanding performance in regression problems.

In summary, XGBoost is a powerful regression algorithm that optimizes the objective function and trains decision tree models using the gradient boosting algorithm, enabling efficient and accurate predictions in regression tasks.

III. Proposed Forecasting Framework

Considering the multivariate, time-series, and nonlinear characteristics of functional module testing data in laptop motherboard manufacturing, this paper proposes an XGB-LSTM model for predicting yield outcomes, as illustrated in"Fig. 4".

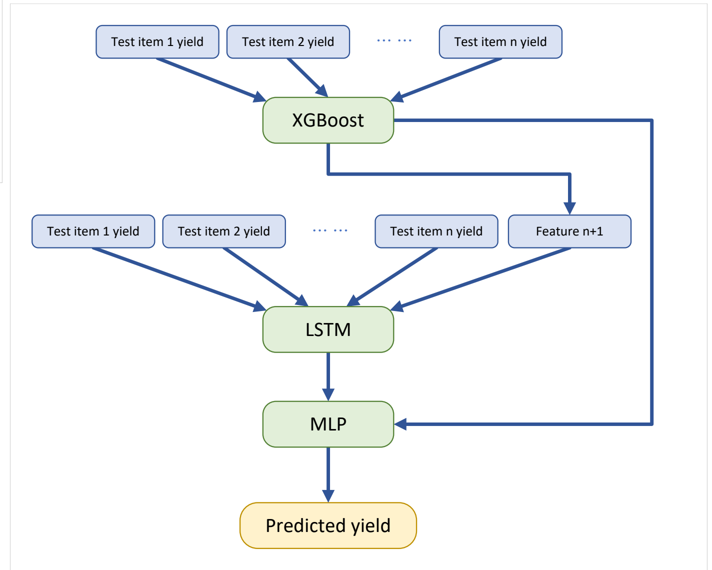

Fig. 4. Construction of XBG-LSTM model.

To address the multivariate nature of functional testing data for motherboards, we have employed a hierarchical deep learning model architecture. Initially, in the first layer, an XGBoost model was trained to extract hidden features from all variables. The XGBoost model is adept at capturing complex relationships within multivariate data and generates feature representations with high predictive power. Subsequently, leveraging the predictions from XGBoost as new features, we combine them with the original features to serve as inputs for the second layer of LSTM. In the second layer, we utilize Long Short-Term Memory (LSTM) networks to capture temporal information within the data. LSTM networks, equipped with memory cells and gate units, effectively handle time-series data and learn long-term dependencies. Moving to the third layer, we perform a weighted fusion of the outputs from XGBoost and LSTM, leveraging both their global and temporal features. This weighted fusion strategy enhances the predictive performance of the model, yielding more accurate predictions. Finally, the XGB-LSTM model produces prediction results and feature importance rankings. In industrial production, feature importance rankings are crucial for guiding production tasks as they aid production personnel in better

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 10:40:30 UTC from IEEE Xplore. Restrictions apply.

understanding the predictive process of the model and adjusting functional testing strategies to improve testing efficiency. Therefore, the proposed XGB-LSTM model not only addresses the complexity of industrial data but also mitigates the opacity issues inherent in traditional neural network models, providing essential decision support for industrial production processes.

IV. Experimental Results And Analysis

A. Data selection and preprocessing

This study is based on six months of functional testing data from a typical laptop motherboard at Factory X. We collected and aggregated the historical yield data of various functional modules of the motherboard model, which were obtained from testing 500 motherboards and subjected to statistical processing. To conform to the data format requirements of the XGB-LSTM network, we preprocessed the data using normalization and sliding window methods. Specifically, we normalized the historical yield data to ensure they are within the same numerical range. Then, we employed a sliding window approach to extract features from the historical yield data of functional modules from the first 500 to 1000 motherboards, with intervals of 50 motherboards, as inputs to the model. The aim is to utilize these input data to predict the yield of each functional module on the next motherboard.

B. Model training

When training the XGBoost model, grid search and cross-validation were employed to obtain the optimal parameter combination.

The parameters for XGBoost were set as follows: learning rate (*Learning_rate*) = 0.1, number of estimators (*n_estimators*) = 50, maximum depth (*max_depth*) = 3, minimum child weight (*min_child_weight*) = 1, gamma = 0, subsample = 0.7, colsample by tree (*colsample_by_tree*) = 0.9, scale pos weight (*scale_pos_weight*) = 1, regularization lambda (*reg_lambda*) = 3, regularization alpha (*reg_alpha*) = 0.

For the LSTM network constructed based on PyTorch, the parameters were set as follows: number of neurons (*Neurons*) = 128, epochs = 100, layers = 2,batch size (*batch size*) = 32. Adam optimizer was chosen to adjust the learning rate.

For the MLP network constructed based on PyTorch, the parameters were set as follows: number of neurons (*Neurons*) = 128, epochs = 100, layers = 1,batch size (*batch size*) = 32. Adam optimizer was chosen to adjust the learning rate.

C. Results

Due to commercial privacy constraints, we are unable to disclose real data in this context. However, after validating the yield prediction of thousands of motherboard functional modules, we have demonstrated the effectiveness of the model. “Fig. 5” illustrates the deployment of our algorithm within the factory environment, further affirming its efficacy in predicting the yield of typical motherboard functional modules at X factory.

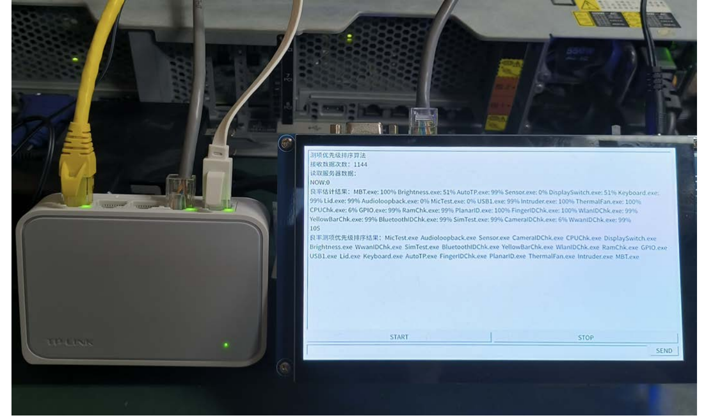

Fig. 5. Our yield prediction model has been implemented on the production line at X factory.

To demonstrate the effectiveness of our approach, we utilized a desensitized dataset derived from real-world data. From this dataset, we selected the yield prediction results of a test project for visualization analysis.

We select RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and R^2 (R-squared) as evaluation metrics for model performance, defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{7}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{8}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{9}$$

Where y_i represents the true value, $\hat{y}_i$ represents the predicted value, and $\bar{y}$ represents the mean of the true values. Smaller RMSE and MAE values and larger R^2 values indicate more accurate predictions.The performance evaluation results of the predictive model are shown in in TABLE I. It is evident from TABLE I that the model’s predicted values closely align with observed data, outperforming baseline models across various evaluation metrics.

TABLE I
Predictive Algorithm Performance

	Evaluation Metrics		
MODEL	RMSE	MAE	R^2
XGBoost	0.001094638	0.000837593	0.838365928
LSTM	0.001327406	0.001025459	0.793284638
XGB-LSTM	0.000889765	0.000634917	0.876809934

“Fig. 6” provides a more intuitive visualization of the accuracy of the model’s prediction results. From the graph, it can be clearly observed that the XGB-LSTM model utilized in this study accurately forecasts the yield of the motherboard functional modules. The predicted values closely match the

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 10:40:30 UTC from IEEE Xplore. Restrictions apply.

observed data, demonstrating the model's strong predictive capability. This further validates the effectiveness and reliability of the XGB-LSTM model for this task.

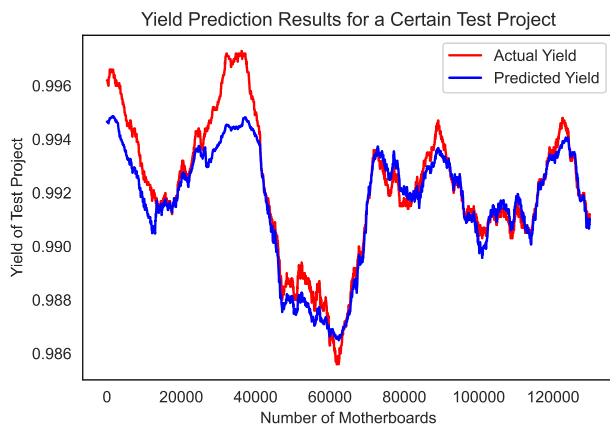

Fig. 6. Yield Prediction Results for a Certain Test Project.

Additionally, our XGB-LSTM model is capable of generating a ranking map illustrating the importance of the top 10 features, as depicted in "Fig. 7". To address commercial privacy concerns, symbols are utilized to represent feature names. The ranking of feature importance serves as a valuable guide for subsequent production operations, playing an indispensable role in industrial production. Through this ranking map, we can better understand which features have the most significant impact on production outcomes, thereby optimizing the production process and improving product quality.

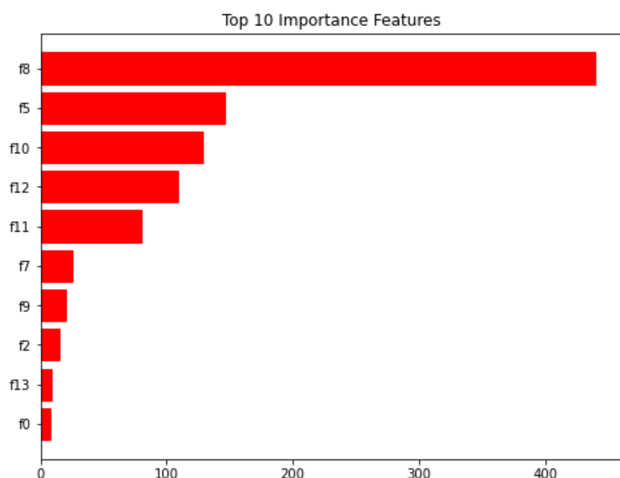

Fig. 7. Feature Importance.

V. Conclusion

This study provides significant insights into the selection of testing projects during the notebook motherboard functional testing phase at X Factory, with which we collaborate, and offers theoretical support for decision-making in the factory production process. In this paper, we adopt the XGB-LSTM model to forecast the future yield of motherboard modules based on the historical yield data of over 30 modules. XGBoost excels at extracting hidden features from multivariate data, while LSTM networks are proficient at capturing temporal information. Consequently, the fusion of these two models effectively addresses the challenges posed by multivariate time series problems, thereby facilitating the prediction of motherboard module yields. This, in turn, provides theoretical underpinnings for the selection of testing projects during the functional testing process.

References

[1] C. Houdek and C. Design, "Inspection and testing methods for pcbs: An overview," *Engineer/OwnerCaltronics Design & Assembly*, 2016.

[2] M. Serban, Y. Vagapov, Z. Chen, R. Holme, and S. Lupin, "Universal platform for pcb functional testing," in *2014 International Conference on Actual Problems of Electron Devices Engineering (APEDE)*, vol. 2, pp. 402–409, IEEE, 2014.

[3] J. Ribeiro, R. Lima, T. Eckhardt, and S. Paiva, "Robotic process automation and artificial intelligence in industry 4.0–a literature review," *Procedia Computer Science*, vol. 181, pp. 51–58, 2021.

[4] J.-S. Jwo, C.-S. Lin, and C.-H. Lee, "Smart technology–driven aspects for human-in-the-loop smart manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 114, pp. 1741–1752, 2021.

[5] D. Li, L. Wang, and Q. Huang, "A case study of sos-svr model for pcb throughput estimation in smt production lines," in *2019 International Conference on Industrial Engineering and Systems Management (IESM)*, pp. 1–6, IEEE, 2019.

[6] H. Khajavi and A. Rastgoo, "Improving the prediction of heating energy consumed at residential buildings using a combination of support vector regression and meta-heuristic algorithms," *Energy*, vol. 272, p. 127069, 2023.

[7] D. Gefang, G. Koop, and A. Poon, "Forecasting using variational bayesian inference in large vector autoregressions with hierarchical shrinkage," *International Journal of Forecasting*, vol. 39, no. 1, pp. 346–363, 2023.

[8] B. Zhang, J. C. Chan, and J. L. Cross, "Stochastic volatility models with arma innovations: An application to g7 inflation forecasts," *International Journal of Forecasting*, vol. 36, no. 4, pp. 1318–1328, 2020.

[9] L. Mohimont, A. Chemchem, F. Alin, M. Krajecki, and L. A. Steffenel, "Convolutional neural networks and temporal cnns for covid-19 forecasting in france," *Applied Intelligence*, vol. 51, no. 12, pp. 8784–8809, 2021.

[10] T. Swathi, N. Kasiviswanath, and A. A. Rao, "An optimal deep learning-based lstm for stock price prediction using twitter sentiment analysis," *Applied Intelligence*, vol. 52, no. 12, pp. 13675–13688, 2022.

[11] S. Pan, S. C. Long, Y. Wang, and Y. Xie, "Nonlinear asset pricing in chinese stock market: A deep learning approach," *International Review of Financial Analysis*, vol. 87, p. 102627, 2023.

[12] J. Yang, S. Li, Z. Wang, H. Dong, J. Wang, and S. Tang, "Using deep learning to detect defects in manufacturing: a comprehensive survey and current challenges," *Materials*, vol. 13, no. 24, p. 5755, 2020.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[15] Y. Miao, M. Gowayyed, and F. Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *2015 IEEE workshop on automatic speech recognition and understanding (ASRU)*, pp. 167–174, IEEE, 2015.

[16] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[17] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 10:40:30 UTC from IEEE Xplore. Restrictions apply.