Exploration of AI Applications of Neural Networks in TFT-LCD Film Thickness Prediction

Hong Yanping, Liu Ziyuan, Zhang Jie, Luo Peng, Liu Bin

Wuhan BOE Optoelectronics Technology Co., Ltd., WUHAN 430000, China

Abstract

This paper presents a virtual model for predicting the thickness of TFT LCD films based on neural networks. The Python programming environment, coupled with neural networks, was utilized to conduct machine learning training on over 100,000 sets of FGI coating parameters and their corresponding film thicknesses. The output of this training is an FGI film thickness prediction model, and a VM (Virtual Model) management system was developed to implement the model's application. The results indicate that the residuals between the model's predicted values and the actual measured values remain within 30 (with an FGI tolerance range of ±629), and the standardized residuals are 0.48, meeting the ±2 range, demonstrating high prediction accuracy of the model. This virtual model enables the prediction of film thickness within 10 seconds of panel output from the CVD coating unit, significantly improving the monitoring efficiency of film thickness compared to traditional measurement methods that require 10 minutes to output measured values. Additionally, the VM management system visually displays the trends of coating parameters, allowing technical departments to quickly adjust parameters based on trend fluctuations, effectively reducing film thickness fluctuations. Through monitoring with the virtual model, we discovered periodic variations in FGI film thickness, which are related to chamber cleanliness. This discovery breaks through the limitations of conventional monitoring. This case successfully demonstrates the application value of neural networks in predicting FGI film thickness in TFT LCD production.

Author Keywords

TFT LCD film thickness; neural networks; virtual model prediction; AI

1. Introduction

The film thickness of TFT LCD is a critical monitoring item for the panel, as it has a direct and profound impact on the panel's optical, electrical, and mechanical properties. Currently, the existing production process primarily relies on sampling measurement combined with SPC (Statistical Process Control) charts for management and control. However, this method presents two major issues: firstly, the measurement process is time-consuming, affecting production efficiency, and measurements may be delayed due to production Q Time constraints; secondly, the sampling mode may result in some samples not being detected, posing a potential risk of quality defects escaping detection. With the rapid advancements in display technology and evolving market demands, customers have imposed stricter requirements on the accuracy of TFT LCD film thickness control. How to precisely regulate film thickness and achieve real-time online monitoring has become one of the key technical challenges that urgently need to be addressed in the TFT LCD manufacturing field.

With the continuous development and popularization of intelligent technologies, model prediction [1] has become increasingly widespread and deeply integrated into various fields. This case plans to adopt model prediction for virtual measurement of film thickness. Common model prediction methods include linear regression [2], time series analysis, decision trees, support vector machines, neural networks [3] and random forests [4].

To select the appropriate method, we analyzed and tested the same dataset. It is worth noting that coating parameters and film thickness are continuous variables, and there exists strong linear correlation between some coating parameters. This correlation not only increases the complexity of data analysis but also poses higher requirements for the accuracy and robustness of the prediction model. We conducted comparative validations using four methods: linear regression, decision trees, neural networks, and random forests. The results showed that linear regression had the lowest R-squared and R-squared prediction values, indicating its unsuitability for this case. The R-squared prediction values of decision trees and random forests were lower than those of neural networks, indicating that neural networks had the best prediction capabilities. Therefore, this case adopted neural networks to learn and train from a large amount of historical coating parameters and their corresponding film thicknesses, thereby constructing a virtual model.

Table 1. Comparison of Model Construction Methods

Result	Linear Regression	Decision Tree	Neural Network	Random Forest
R square	0.58	0.60	0.71	0.64
R square Prediction	0.38	0.42	0.70	0.59
MSE	35	34	22	30

2. Establishment of a Virtual Model for Film Thickness Prediction

2.1. Data Preparation and Processing

2.1.1 Selection of Research Objects

The research object selected in this paper is the 65AK FGI film thickness, with coating parameters as the independent variables and the average film thickness as the dependent variable. Based on the PECVD FGI coating process and equipment structure, the influencing factors of FGI film formation are analyzed. The FGI film formation parameters include coating time, operating power, gas flow rate, chamber pressure, substrate temperature, etc., totaling 20 quantifiable factors. Since uncontrollable factors mainly vary with the coating chamber, the model construction takes the chamber as the smallest unit. The required data is extracted from the CIM system, with the larger the data volume, the better.

2.1.2 Data Cleaning

Data cleaning [5] is a crucial step in the data preprocessing process, directly affecting the accuracy and reliability of

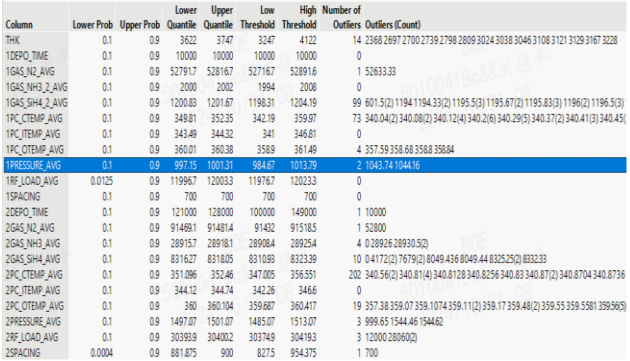
ISSN 0097-996X/25/5603-1606-$1.00 © 2025

subsequent data analysis and modeling. The following three types of abnormal data are primarily removed through data processing:

1) Missing values, usually caused by equipment interruptions.

2) Abnormal values, which are data that significantly deviate from other observations. Outlier detection methods such as quantile range outliers and robust fitting outliers are employed, with multiple methods attempted.

3) Redundant values, involving the identification and removal of duplicate records or data rows, with Decap re-submitted products eliminated based on repeated Glass IDs. Data cleaning is an iterative process that may require multiple checks and modifications to ensure data quality meets analysis or modeling requirements.



Column	Lower Prob	Upper Prob	Lower Quantile	Upper Quantile	Low Threshold	High Threshold	Number of Outliers	Outliers (Count)
THK	0.1	0.9	3622	3747	3247	4122	14	2368 2697 2700 2739 2798 2809 3024 3038 3046 3108 3121 3129 3167 3228
1DEPO_TIME	0.1	0.9	10000	10000	10000	10000	0	
1GAS_N2_AVG	0.1	0.9	52791.7	52816.7	52716.7	52891.6	1	52633.33
1GAS_NH3_2_AVG	0.1	0.9	2000	2002	1994	2008	0	
1GAS_SiH4_2_AVG	0.1	0.9	1200.83	1201.67	1198.31	1204.19	99	601.5(2) 1194 1194.33(2) 1195.5(3) 1195.67(2) 1195.83(3) 1196(2) 1196.5(3)
1PC_CTEMP_AVG	0.1	0.9	349.81	352.35	342.19	359.97	73	340.04(2) 340.08(2) 340.12(4) 340.2(6) 340.29(5) 340.37(2) 340.41(3) 340.45(
1PC_ITEMP_AVG	0.1	0.9	343.49	344.32	341	346.81	0	
1PC_OTEMP_AVG	0.1	0.9	360.01	360.38	358.9	361.49	4	357.59 358.68 358.8 358.84
1PRESSURE_AVG	0.1	0.9	997.15	1001.31	984.67	1013.79	2	1043.74 1044.16
1RF_LOAD_AVG	0.0125	0.9	11996.7	12003.3	11976.7	12023.3	0	
1SPACING	0.1	0.9	700	700	700	700	0	
2DEPO_TIME	0.1	0.9	121000	128000	100000	149000	1	10000
2GAS_N2_AVG	0.1	0.9	91469.1	91481.4	91432	91518.5	1	52800
2GAS_NH3_AVG	0.1	0.9	28915.7	28918.1	28908.4	28925.4	4	0 28926 28930.5(2)
2GAS_SiH4_AVG	0.1	0.9	8316.27	8318.05	8310.93	8323.39	10	0 4172(2) 7679(2) 8049.436 8049.44 8325.25(2) 8332.33
2PC_CTEMP_AVG	0.1	0.9	351.096	352.46	347.005	356.551	202	340.56(2) 340.81(4) 340.8128 340.8256 340.83 340.87(2) 340.8704 340.8736
2PC_ITEMP_AVG	0.1	0.9	344.12	344.74	342.26	346.6	0	
2PC_OTEMP_AVG	0.1	0.9	360	360.104	359.687	360.417	19	357.38 359.07 359.1074 359.11(2) 359.17 359.48(2) 359.55 359.5581 359.56(5)
2PRESSURE_AVG	0.1	0.9	1497.07	1501.07	1485.07	1513.07	3	999.65 1544.46 1544.62
2RF_LOAD_AVG	0.1	0.9	30393.9	30400.2	30374.9	30419.3	3	12000 28060(2)
2SPACING	0.0004	0.9	881.875	900	827.5	954.375	1	700

Fig 1. Data cleaning

2.1.3 Data Normalization

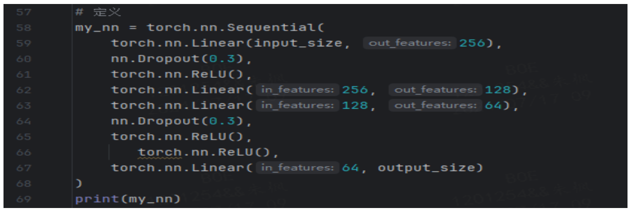
Data normalization can accelerate algorithm convergence and improve model prediction accuracy. Common data normalization methods include Min-Max normalization (also known as linear normalization) and standard deviation normalization. After verification, different normalization methods have minimal impact on the neural network model structure. Therefore, standard deviation normalization is selected in this paper.

2.2. Model Construction

This case primarily uses the Pytorch deep learning framework for model construction and training, with the main processes including data preparation, model design, model training, model evaluation, and model tuning.

Data Preparation: Use pandas to read the data from the dataset and separate the feature variables (features, also known as predictors) from the target variables (labels, also known as responses). Convert the data to numpy array format for data splitting. Use sklearn's train_test_split to divide the data into training and testing sets, and use StandardScaler to standardize the features and labels of the training set. Simultaneously, save the standardization parameters for data processing in the testing set or after model deployment.

Model Design: Using Pytorch define the model structure by combining the layers of the neural network using torch.nn. Sequential. Define the input and output of each layer and the activation function to construct the model. Add Dropout layers between layers to prevent overfitting.

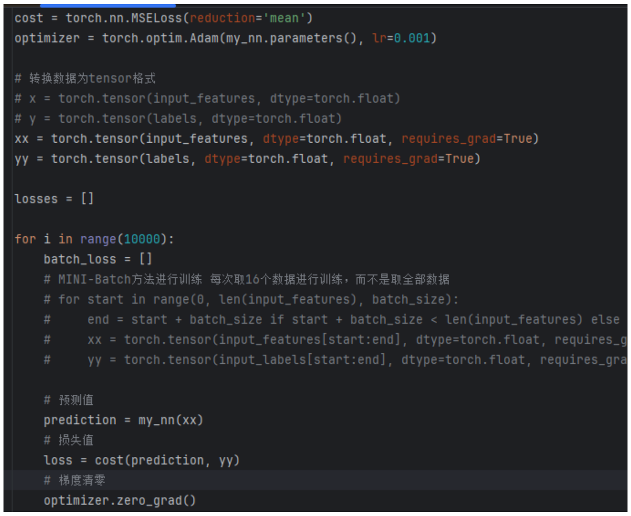


```
# 定义
my_nn = torch.nn.Sequential(
    torch.nn.Linear(input_size,  out_features: 256),
    nn.Dropout(0.3),
    torch.nn.ReLU(),
    torch.nn.Linear( in_features: 256,  out_features: 128),
    torch.nn.Linear( in_features: 128,  out_features: 64),
    nn.Dropout(0.3),
    torch.nn.ReLU(),
        torch.nn.ReLU(),
    torch.nn.Linear( in_features: 64, output_size)
)
print(my_nn)
```

Fig 2. Neural Network Model Structure

Model Training: Follow the standard process, using the MSE Loss loss function and Adam optimizer. Add lr_scheduler to implement dynamic learning rates. Set weight decay in the optimizer to prevent overfitting and improve model generalization. Convert the data to tensor format for model training. In each epoch, perform forward propagation to obtain predictions and calculate the loss, then use the optimizer to optimize the model, including gradient clearing, backward propagation, and parameter updating. Simultaneously, record and print the loss to help assess the training effect.

Model Evaluation: After training, evaluate the model using the testing set. Set the model to evaluation mode using the eval method. Input the standardized testing set data into the model to obtain predictions. After reversing the standardization of the predictions, calculate the model evaluation metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), and Goodness of Fit (R^2).



```
cost = torch.nn.MSELoss(reduction='mean')
optimizer = torch.optim.Adam(my_nn.parameters(), lr=0.001)

# 转换数据为tensor格式
# x = torch.tensor(input_features, dtype=torch.float)
# y = torch.tensor(labels, dtype=torch.float)
xx = torch.tensor(input_features, dtype=torch.float, requires_grad=True)
yy = torch.tensor(labels, dtype=torch.float, requires_grad=True)

losses = []

for i in range(10000):
    batch_loss = []
    # MINI-Batch方法进行训练 每次取16个数据进行训练，而不是取全部数据
    # for start in range(0, len(input_features), batch_size):
    #     end = start + batch_size if start + batch_size < len(input_features) else
    #     xx = torch.tensor(input_features[start:end], dtype=torch.float, requires_g
    #     yy = torch.tensor(input_labels[start:end], dtype=torch.float, requires_gra

    # 预测值
    prediction = my_nn(xx)
    # 损失值
    loss = cost(prediction, yy)
    # 梯度清零
    optimizer.zero_grad()
```

Fig 3. Neural Network Model Structure

Model Tuning [6]: Use the skorch library to combine PyTorch models with scikit-learn functionalities for parameter optimization. Define the model using skorch's Neural Net Regressor combined with Pytorch, and define a hyperparameter grid for the learning rate and maximum number of training epochs. Use GridSearchCV (grid search) to optimize the hyperparameters of the constructed model to obtain the best hyperparameters and results. This method can be used for model hyperparameters tuning when training models using datasets from various chambers.

2.3. Confirmation of Model Validity

Model accuracy (commonly referred to as the accuracy or precision of the model) holds an extremely important position in the fields of machine learning and data science. It is a core metric for evaluating model performance and directly relates to the model's ability to effectively solve practical problems. In this case, model accuracy is tested through three dimensions:

1）Model Self-Validation: This primarily includes the R-squared (coefficient of determination), Mean Absolute Error (MAE), and the average of the absolute differences between predicted and actual values. In this case, the R-squared for the training set is 0.71, and for the validation set, it is 0.70. The mean errors are all within 30, indicating good model fit and acceptable model accuracy.

2）Residual Evaluation: The residuals in this case conform to a normal random distribution, with standardized residuals of 0.48 falling within the range of -2 to 2, demonstrating sufficient explanatory power of the model.

Table 2. The residual of the model prediction versus actual measurement

Sample NO.	Product Spec	Measured	Predicted	Residual
1	3700 ±629	3740	3731	-9
2	3700 ±629	3660	3659	-1
3	3700 ±629	3618	3629	11
4	3700 ±629	3727	3732	5
5	3700 ±629	3653	3630	-23
6	3700 ±629	3621	3633	12
7	3700 ±629	3660	3648	-12
……	……	……	……	……
36	3700±629	3745	3729	-16

3）Business Requirement Assessment: In this case, the difference between the model's predicted values and the actual measured values is relatively small, as shown in the graph 4 where the two almost coincide. The fluctuation ratio compared to the tolerance band is only 0.02, and the fluctuation ratio compared to the control band is approximately 0.05. The fluctuation is negligible within the business control range; therefore, the model can be put into use.

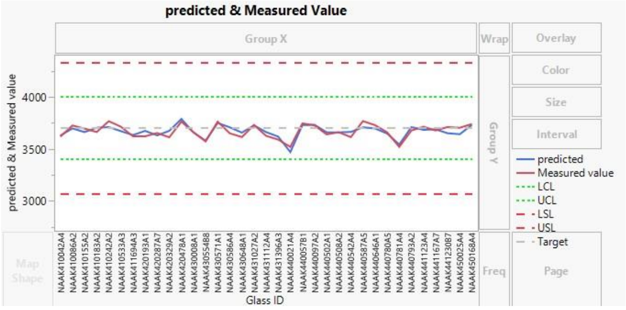


Fig 4. Model prediction and actual measurement trend chart

2.4. Model Deployment and Utilization

This case involves deploying model code, coating parameters retrieved from a data warehouse, and actual measured film thickness values onto a new system through a Python programming environment. The deployment framework is illustrated in Figure 5. The entire deployment system consists of four functional modules:

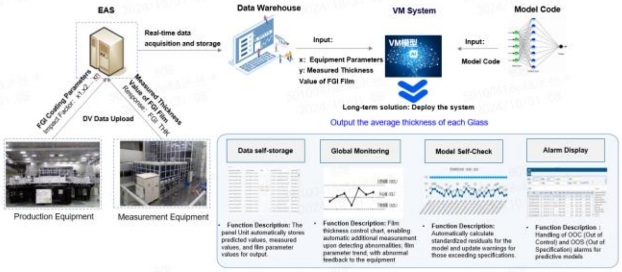
1）Data Auto-Storage: Ten seconds after the panel is produced from the CVD coating unit, the system automatically stores the panel coating parameters, the measured value of film thickness, and the model's predicted value.

2）Graphical Monitoring Interface: This includes an I-MR control chart for the average film thickness of FGI, trend charts displaying coating parameters, and a Pareto chart showing the importance of influencing factor characteristics, all for trend management and control.

3）Model Accuracy Self-Check: The system updates its judgment based on whether the standardized residuals fall within the range of -2 to +2.

4）Alarm Display Interface: Set alarm rules for the prediction model and conduct alarm management.

Currently, we have successfully deployed virtual models into the system, with the model deployment system interface depicted in Figure 6. Two virtual models have been implemented, catering to the prediction application of FGI film thickness for 16 products, achieving a 100% predictive monitoring of FGI film thickness through virtual models. Through the monitoring of these virtual models, we have identified periodic variations in FGI film thickness, as illustrated in Figure 6. Upon analysis by the technical department, these variations are correlated with chamber cleaning. The film thickness reaches its lowest level after cleaning and gradually increases thereafter, with the cycle of film thickness variation mirroring the chamber cleaning cycle. This discovery has surpassed the monitoring limitations of conventional sampling measurements. Furthermore, the virtual measurement management system can display coating parameter trend charts, allowing the technical department to visually identify parameter changes and make prompt adjustments, thereby effectively reducing film thickness fluctuations.



- Function Description: The panel Unit automatically stores predicted values, measured values, and film parameter values for output.
- Function Description: Film thickness control chart, enabling automatic additional measurement upon detecting abnormalities; film parameter trend, with abnormal feedback to the equipment
- Function Description: Automatically calculate standardized residuals for the model and update warnings for those exceeding specifications.
- Function Description : Handling of OOC (Out of Control) and OOS (Out of Specification) alarms for predictive models

Fig 5. Model Deployment Framework

Fig 6. VM Management System

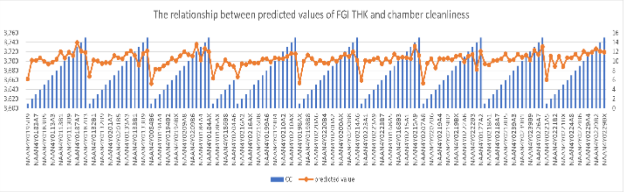


Fig.7 The relationship between predicted values of FGI THK and chamber cleanliness

3. Conclusion and Outlook

3.1. Conclusion

In this case, a neural network was utilized to construct a virtual prediction model for FGI film thickness. The model demonstrated good fit and explanatory power, meeting business needs and achieving precise film thickness prediction. Wuhan BOE Technology Co., Ltd. has deployed this virtual model into the system, with two models online, enabling FGI film thickness prediction for 16 products. Innovatively, this has achieved a 100% predictive application of virtual models for film thickness, revealing periodic variations in FGI film thickness that are correlated with chamber cleaning. Additionally, the system visually displays fluctuations in coating parameters, allowing the business department to quickly identify and adjust parameters, effectively reducing film thickness fluctuations. The successful application of this case has broken through the limitations of traditional measurement equipment, enabling film thickness monitoring to be independent of Q time, greatly improving the timeliness of monitoring and bringing significant convenience to TFT LCD production. In the future, we will continuously optimize this model to enable predictive feedback to the equipment through the model, allowing for dynamic adjustment of coating parameters. This will achieve precise control of film thickness and enhance its stability.

3.2. Outlook

Neural network prediction of TFT-LCD film thickness offers multiple benefits, including improved prediction accuracy, enhanced production efficiency, cost reduction, and promotion of intelligent production. The application prospects of neural networks in the TFT-LCD industry are broad. In the future, we can attempt to adopt neural network-based deep learning detection technology [1], which, through training and learning from massive samples of panel defects, can fully grasp image patterns and extract the essential characteristics of target defects, achieving high-precision defect recognition and classification. We can also explore the use of neural networks to optimize pixel layout, enhance color performance [2], and reduce power consumption. As technology continues to develop and the market evolves, neural networks will play an increasingly important role in the production, research and development, and design of TFT-LCD, driving the industry towards more intelligent, efficient, and personalized development.

4. Acknowledgements

The author would like to express his appreciation to all BOE staff who have contributed to this work, thank you for your unselfish help, as well as your comments and Suggestions in writing this paper.

5. References

[1] Zou Tao, Ding Baocang, Zhang Duan. An Introduction to Engineering Applications of Model Predictive Control [M]. Chemical Industry Press, 2010.

[2] Fox J. Applied Regression Analysis and Generalized Linear Models[J]. sage, 2008.

[3] Interdepartmental Committee on Biomedical Engineering, University of Science and Technology of China. Neural Networks and Their Applications [M]. University of Science and Technology of China Press, 1992.

[4] Jisoo Ham, Yangchi Chen, Melba M. Crawford, Joydeep Ghosh. Investigation of the random forest framework for classification of hyperspectral data [J], IEEE Transactions on Geoscience and Remote Sensing, 2005.

[5] Guo Zhimeng, Zhou Aoying. A Survey on Data Quality and Data Cleaning [J]. Journal of Software, 2002. DOI: CNKI: SUN: RJXB.0.2002.

[6] Gong Daoqing. Research on Evolutionary Tuning of Hyperparameters for Deep Neural Network Models and Its Applications [D]. Nanning Normal University, 2020.