SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Adaptive Algorithms for Non-Stationary Data Environments

S. Uma Mageshwari[1], Sakshi Pandey[2], Mr. Manoj I. Patel[3], Yudhveer Singh Moudgil[4], Sivasankari V[5], Shanthi R[6], Aastha Mishra[7], Abhijeet Deshpande[8]

[1] Assistant Professor, Department of Computer Science and Engineering, K. Ramakrishnan College of Technology, Tamil Nadu, India. Email: umamag28@gmail.com
[2] Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab 140401, India. Email: sakshi.pandey.orp@chitkara.edu.in ORCID: 0009-0008-9966-2165
[3] Assistant Professor, Faculty of Computer Science and Application, Gokul Global University, Sidhpur, Gujarat, India. Email: manoj.patel@gokuluniversity.ac.in ORCID: 0009-0009-0731-9528
[4] Assistant Professor, Department of Computer Science & Engineering, Dev Bhoomi Uttarakhand University, Dehradun, Uttarakhand, India. Email: socse.yudhveer@dbuu.ac.in ORCID: 0009-0002-4805-609X
[5] Assistant Professor, Department of Mathematics, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: sivasankariv@maher.ac.in
[6] Assistant Professor, Department of Mathematics, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: shanthir@maher.ac.in
[7] Assistant Professor, Department of School of Engineering & Technology, Noida International University, Greater Noida, Uttar Pradesh, India. Email: aastha.mishra@niu.edu.in
[8] Assistant Professor, Department of Mechanical Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra 411037, India. Email: abhijeet.deshpande@vit.edu

Abstract
Maintaining model performance becomes difficult when the data distribution fails to remain steady in dynamic contexts where the data generated is constantly changing. Typical machine learning algorithms struggle to cope with concept drift, which leads to lower prediction accuracy and reliability. The goal of this research is to design an adaptive learning system for forecasting financial markets and monitoring industrial equipment in non-stationary contexts. The proposed approach combines Adaptive Random Forest and Wolf Pack Search Optimization (WPS-ARF) model, where WPS optimizes ARF hyperparameters for enhanced adaptation, while ARF performs robust forecasting and fault detection in non-stationary environments. Preprocessing using Min-Max normalization ensures uniform feature ranges and stable learning, while PCA reduces dimensionality, removes redundancy, lowers computational cost, and mitigates overfitting. The model is coded in python and tested with financial and industrial datasets. Results show that WPS-ARF achieves Mean Squared Error (MSE) of 0.010, a Stability (CV) of 3, and a response time of 0.15 s for financial forecasting, while attaining 98% fault detection accuracy, a CV of 4, and a response time of 0.18 s for industrial monitoring, outperforming baseline models in accuracy, stability, and computational efficiency. As a result, the proposed model improves the predictive performance and robustness in the non-stationary environment, and can be used for decision-making in the financial and industrial systems.

Keywords: Adaptive Machine Learning, Non-Stationary Data Environments, Online Learning, Concept Drift Detection, Incremental Learning.

1 INTRODUCTION

The demand for models that can maintain their performance in changing environment has increased as machine-learning (ML) technologies gain popularity in the financial, industrial, and intelligent monitoring sectors [1]. The financial markets, the performance of industrial equipment and the observations of a sensor-based system can generate continuous change [2]. This implies that the statistical properties of data may evolve over time, and thus the patterns obtained from past data may not be the representative of current conditions [3]. Building reliable and accurate prediction models in non-stationary contexts has become extremely difficult due to concept drift [4]. Static decision tree, Support Vector Machine (SVM) and fixed neural networks are the approaches of conventional methods, which are built to be stable and not adaptive [5]. These models are then trained on the historical data, and deployed without modification, under the assumption that future data can look like the past data. But as soon as this assumption fails, performance falls and there is no more cost-effective way to mitigate the decline of performance except by expensive full-scale retraining, which is computationally time-consuming and reactive in nature [6, 7]. Moreover, conventional architectures are challenged by the high dimensional streaming data because of increased computational load, memory limitation and processing speed, which makes adaptability more difficult [8]. The problem, therefore, is not just to create the correct models, but to create models that are correct in the face of a changing world [9]. Concept drift can occur in various ways, such as sudden distributional changes, gradual changes in patterns, or periodic seasonal variations, and it requires a different and intelligent solution [10]. This is complicated by the temporality of the knowledge, since the value of the past diminishes at different speeds across different domains, and the static models are always of the past.

This research presents an adaptive learning framework for autonomous detection of distributional shifts, incremental model updating and predicting the current distribution without expensive retraining cycles, based on ARF and WPS Optimization. The model is applied to and tested in two difficult application areas, one concerning forecasting in financial markets and the other concerning fault detection in industry, where its adaptability has a direct influence on the quality of the decisions and the operational results.

The rest of the part is as follows: Part 2 summarizes previous work and existing gaps in adaptive machine learning for non-stationary data environments. The proposed WPS-ARF model is given in Part 3. The performance evaluation and comparative analysis are discussed in Part 4. Finally, in Part 5, limitations and future directions are presented.

2 LITERATURE REVIEW

The concept drift problem is still a predominant issue in streaming and non-stationary environments, and several adaptive models have been developed to address it. Scalable Concept Drift Adaptation (SCDA) [11] tackled stream data mining with concept drift by dynamically expanding and contracting ball. The prediction accuracy, memory usage, and scalability were improved, but couldn't be based on ensemble learning. To achieve better classification accuracy and training efficiency for evolving data streams, a Weighted Incremental–Decremental Support Vector Machine (SVM) with a shifting weighted window [12] was proposed, which had drawbacks of SVM-based architecture in terms of scalability and adaptability. While error-rate monitoring and data-distribution monitoring proved effective in drift detection, they had limited generalizability due to the need for retraining SVM and network-specific datasets [13]. The accuracy and speed of abnormal network traffic detection was enhanced by the use of PCA, feature fusion, and Single-Stage Headless Face Detector (SSH) algorithms [14]; however, concept drift adaptation were not investigated. To detect cybersecurity threats, Hybrid Drift Detection and Adaptation Framework (HDDAF) [15] combined drift detection, feature selection, adversarial training, and incremental learning to deliver better accuracy and F1 scores, but this was restricted to cybersecurity applications. In Fuzzy feature adaptation with accelerated optimization and source instance selection [16] the accuracy of streaming data classification and computational efficiency were increased and the focus was placed mainly on domain adaptation. Dynamic Drift Awareness and Diffusion Enhancement (DDADE) [17] are fused to enhance adaptive learning for industrial anomaly detection, demonstrating superiority over baseline methods and validation on a simulated dataset. The application of adaptive machine learning and online/incremental learning in financial forecasting and industrial equipment monitoring improved financial forecasting and increased the accuracy of fault detection up to 95%, yet computational efficiency, anomaly detection, data fusion and interpretability were not addressed. The research [18] develops adaptive machine learning for non-stationary financial and industrial environments using online and incremental learning, achieving 0.015 MSE and 95% accuracy, with interpretability and computational limitations. Thus, the existing methods still suffer from the lack of scalability, scalability efficiency, and adaptability for concept drift problem in various real-world applications, which calls for the proposed WPS-ARF model.

3 METHODOLOGY

The suggested model of the WPS-ARF has two phases. In the first phase, financial and industrial streaming data is normalized by Min-Max and reduced by PCA, then optimized and trained by WPSO to optimize ARF. During the adaptive phase, a drift detection mechanism can detect the drifted data source, and if a drift was detected, WPSO is re-optimize ARF and the model can be updated, otherwise incremental learning is carried out to achieve continuous adaptation and performance in non-stationary environments. The overall flow of the proposed model is presented in Figure 1.

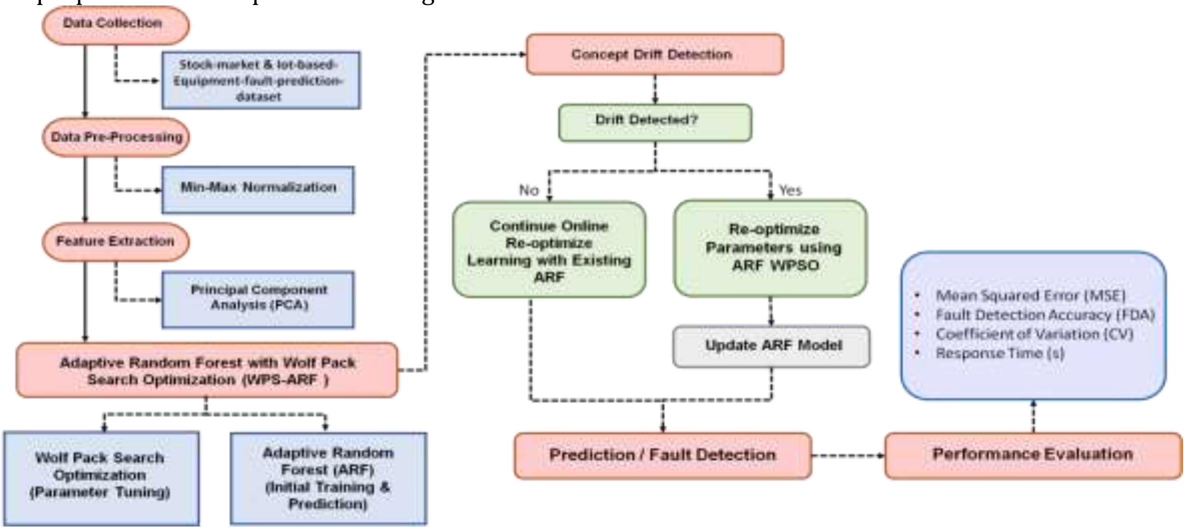

Figure 1: Proposed WPS-ARF Model for Dual-Domain Prediction

3.1 Data Collection

Two Kaggle datasets were used in this research. Dataset 1, a https://www.kaggle.com/datasets/ziya07/stock-market-dataset for AI-driven prediction and trading strategy optimization (10,000 records, 13 features), was employed for financial forecasting. Dataset 2, an https://www.kaggle.com/datasets/programmer3/iot-based-equipment-fault-prediction-dataset (50,000 records, 14 features), was used for industrial fault detection. Both datasets represent non-stationary environments and were split into 80% training and 20% testing data.

3.2 Min-Max Scaling for Non-Stationary Data

Normalization maps financial feature of financial markets or industrial sensor into a common range [0,1] to facilitate consistent representation, stable learning and better convergence under non-stationary conditions. It is defined in the following Equation (1):

$$y' = \frac{y - y_{min}}{y_{max} - y_{min}} \quad (1)$$

y is raw input signals from financial indicators and industrial sensor signals, y_{min}, y_{max} were minimum and maximum values of the signals. For financial forecasting and industrial fault detection, the normalized output y^' is used as input to the proposed model.

3.3 Dimensionality Reduction Using PCA

Normalization, then PCA reduces the dimensionality by mapping the data to the lower dimensional space that preserves the maximum variance among the data. This boosts computational efficiency and generalization of non-stationary financial and industrial datasets. The covariance matrix and transformed feature space are given in Equations (2) and (3):

$$\Sigma \frac{1}{M-1} \sum_{j=1}^{M} (y_j - \bar{y})^2 (y_j - \bar{y}) \quad (2)$$

$$X = Y \times W \quad (3)$$

Where M is the number of samples in the financial and industrial datasets, y_j are the j^{th} feature vectors in the financial and industrial datasets, $\bar{y}$ are the mean feature vectors for each dataset, Σ are the covariance matrices reflecting the relationships between features in both the financial and industrial datasets, Y are the matrices of the normalized feature vectors for each dataset, W are the principal component eigenvector matrices, and X are the reduced feature vectors used for financial forecasting and industrial fault detection.

3.4 WPS-ARF for Non-Stationary Environments

For the effective financial market forecasting and the detection of fault in industrial equipment in non-stationary environments, a hybrid WPS-ARF model is designed by integrating WPO-based hyperparameter optimization with the adaptive learning ability of ARF. First, the input data is pre-processed and important features are extracted. WPO then initializes, explores, refines and updates the solution to find the optimal ARF hyperparameters. With the optimized parameters, ARF builds an adaptive set of decision trees and keeps on updating the learning process as new data are received. This blend enhances the robustness, reliability, and adaptability of the predictive model when the data distribution changes over time.

3.4.1 ARF for Dual-Domain Prediction

After preprocessing and PCA-based feature extraction, ARF performs forecasting and fault detection in non-stationary environments. It extends RF through incremental learning and dynamic ensemble updates using bootstrap decision trees. The aggregated outputs enhance accuracy, robustness, and adaptability, enabling reliable prediction and anomaly detection under evolving data distributions. The prediction from each tree is expressed in Equation (4):

$$e(CS_1) = \beta_0 + \beta_0 E_1 + \cdots + \beta_m E_m \quad (4)$$

where $e(CS_1)$ denotes the prediction of the j^{th} decision tree, $\beta_0,...,\beta_m$ denotes regression coefficients and $E_1,...,E_m$ represent selected stock market indicators and industrial sensor features used to capture evolving patterns for trend prediction and anomaly identification. The outputs of all decision trees are aggregated in Equation (5):

$$\gamma = e(DT_1) + e(DT_2) + \cdots + e(DT_m) \quad (5)$$

$$\gamma = \vartheta\{e(DT_1) + e(DT_2) + \cdots + e(DT_m)\} \quad (6)$$

In Equations (5) and (6), γ denotes the aggregated ensemble prediction, $\vartheta(\cdot)$ represents the majority voting function applied to the ensemble outputs, $e(DT_1) + e(DT_2) + \cdots + e(DT_m)$ is the prediction generated by the ith decision tree $(i = 1, 2, \ldots, m)$, and the ARF's total quantity of decision trees is indicated by m. The final prediction is obtained through majority voting, is expressed in Equation (7).

$$\gamma = \underset{m}{\arg\max}\, of(DT_j) \quad \text{where}' j \in 1 \text{tom}' \quad (7)$$

The final ensemble prediction γ, $of(DT_j)$ is the voted output of the j^{th} tree, and $\arg\max$ increases the confidence of the prediction and reliability of the classification. The difference between actual and predicted outputs is calculated to assess the prediction performance with Equation (8):

$$\varepsilon_j = \sum_{j=1}^{m} \gamma_j - o_p \quad (8)$$

In this case, ε_j is the prediction error for the j^{th} sample, m is the total number of evaluated observations,γ_j is the actual value of the stock market or the label of the condition of the equipments, and o_p is the predicted output from the ARF model. It is used to evaluate the difference between the actual values and the predicted values, and is reduced to improve the forecasting performance and fault detection performance under varying data conditions.

3.4.2 WPO for Adaptive Parameter Tuning

WPO models collective search behavior through wandering, calling, and Battle phases to efficiently explore and exploit solution spaces, is shown in Figure 2. It is applied in this research to optimize ARF parameters for financial forecasting and industrial fault detection under non-stationary conditions. Its fast convergence, strong global search ability, and robustness make it suitable for dynamic streaming environments.

WP Initialization (Hyperparameter Initialization): In the proposed WPS-ARF model, M wolves are distributed in a D-dimensional search space, where each wolf represents a candidate hyperparameter configuration of the ARF model. The position vector of the j^{th} wolf can be expressed as $W_j = (w_{j1}, w_{j2}, \ldots, w_{jc})$, where $j = 1,2,\ldots,M$. The initialized position of each wolf in the pack is generated in Equation (9):

$$w_{id} = c_{cmin} + rand(0,1)(w_{cmax} - w_{cmin}) \quad (9)$$

where w_{id} represents the value of the c^{th} ARF hyperparameter for the j^{th} wolf (candidate ARF hyperparameter configuration), rand(0,1) is a random number in the range [0,1], M is the population size, and C is the number of hyperparameters, and w_{cmax} and w_{cmin} are the upper and lower bounds of the c^{th}hyperparameter, respectively.

Head Wolf Position Initialization (Best Solution Identification): To identify optimal ARF parameters for financial forecasting and industrial fault detection, multiple candidate parameter solutions are generated within the search space. The j^{th} candidate solution in the c^{th} parameter dimension is computed in Equation (10):

$$z_{jic} = z_{jd} + rand(-1,1)b_c \quad (10)$$

where z_{jic} denotes the updated candidate value of the c^{th} ARF parameter, z_{jd} represents the current value of the c^{th} parameter, b_c is the search step size controlling parameter adjustment, and $(rand(-1,1))$ is a random exploration factor. The candidate solutions are evaluated using model performance, and the best solution is selected to guide subsequent optimization.

Wolf Pack Attack (Guided Parameter Optimization): After identifying the best candidate solution, the remaining candidate parameter sets move toward the current optimal solution to improve ARF performance for financial forecasting and industrial fault detection. The parameter update process is defined in Equation (11):

$$y_{id} = w_{id} + (w_{lc} - w_{id})rand(-1,1)a_c \quad (11)$$

where y_{id} denotes the updated value of the c^{th} ARF parameter, w_{id} is the current parameter value of a candidate solution, w_{lc} represents the corresponding parameter value in the current best solution, a_c is the parameter update step size, and $(rand(-1,1))$ is a random exploration factor. This update mechanism guides candidate solutions toward optimal parameter configurations while maintaining search diversity.

Battle behavior (Parameter Refinement): As the optimization progresses, candidate ARF parameter solutions are refined around the current best solution to improve financial forecasting and industrial fault detection performance. The adaptive search step is computed in Equation (12):

$$d_c = d_{cmin}(d_{cmax} - d_{cmin})\, exp\left(\frac{In\frac{d_{cmin}}{d_{cmax}}l}{l_{max}}\right) \quad (12)$$

where d_c is the adaptive parameter adjustment step, d_{cmin} and d_{cmax} are the minimum and maximum step sizes, l is the current iteration, and l_{max} is the maximum iteration count. The parameter update rule is expressed in Equation (13):

$$w_{jc}^{l+1} = \begin{cases} w_{jc}^{l} rand \leq \theta \\ w_{kc} + rand(-1,1) d_c rand(0,1) > \theta \end{cases} \quad (13)$$

where w_{jc}^{l+1} and w_{jc}^{l} are the updated and current values of the c^{th} ARF parameter, w_{kc} is the corresponding parameter value in the current best solution, d_c denotes the adaptive step size, $rand(0,1)$is a random number in [0,1], and $\theta$is the exploration-exploitation threshold. If $rand(0,1) \leq \theta$, the parameter is updated toward the best solution; otherwise, the current value is retained.

Wolf Pack Update (Solution Update): To maintain optimization quality, ARF parameters are iteratively updated based on forecasting and fault detection performance. High-fitness solutions are retained, while low-performing ones are replaced using new candidates guided by the update factor α, ensuring solution quality, diversity, and stable convergence toward optimal configurations.

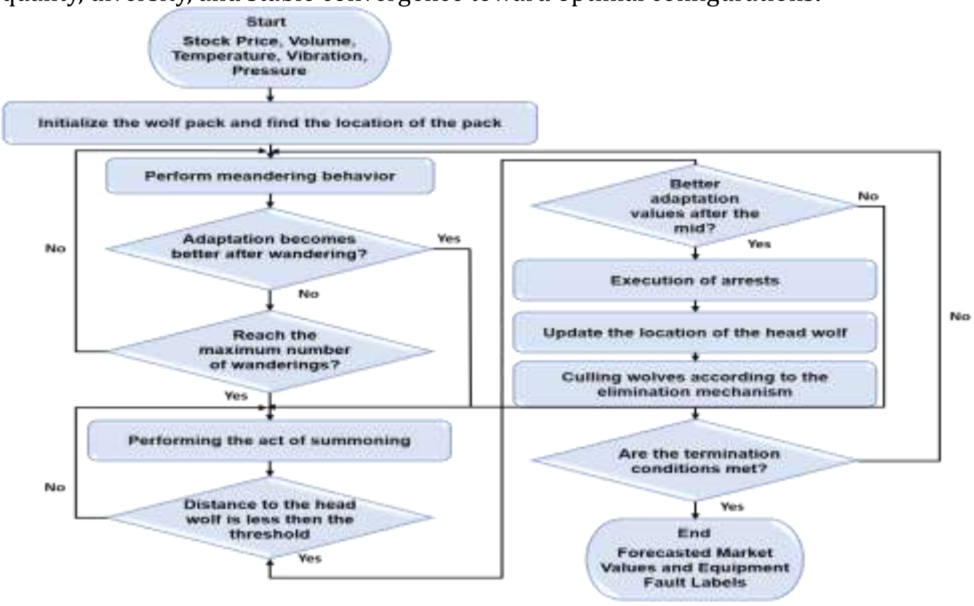

Figure 2. Flow chart of the Wolf Pack optimization algorithm

4 RESULT AND DISCUSSION

Experiments were performed in Python with Scikit-learn with standardized preprocessing and 80:20 ratio split between test and training. The ARF was set up with 200 trees, a maximum depth of 20, and the WPS optimizer was set with a population of 30 and 50 iterations. Assessment of the performance was carried out in terms of MSE, CV, FDA and RT. Figure 3(a) shows the accuracy of the suggested WPS-ARF approach during the 20 epochs of validation and training. The accuracies both rise quickly and reach the same value close to 99%, demonstrating successful learning and good generalization performance. A similar trend is observed in the training and validation loss in figure 3(b) that continues to decrease steadily and converge to a value of approximately 0.05, showing fast convergence, less overfitting and good performance in non-stationary environment.

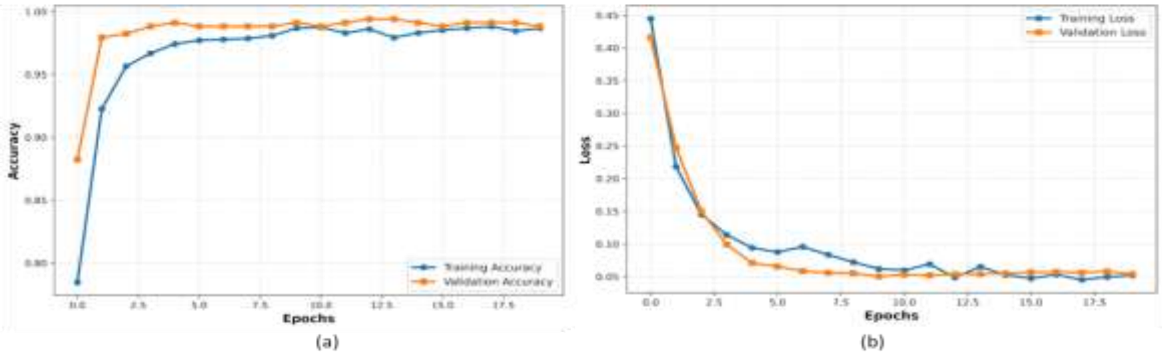

Figure 3. Performance Analysis of the Proposed WPS-ARF Model: (a) Accuracy, (b) Loss

The Stock market dataset in Figure 4(a) displays the normalized trading volume in the stock market, which fluctuates between ~0 and 1, exhibits high frequency oscillations and peaks, suggesting that it is highly non-stationary with dynamic market behavior. The samples are clustered into groups via linkage distance in y-axis of Figure 4(b) IoT dataset dendrogram at low linkage distance (~0.1–0.3), and at high linkage distance (~1.0), to form two distinct groups, normal and fault-prone operational state.

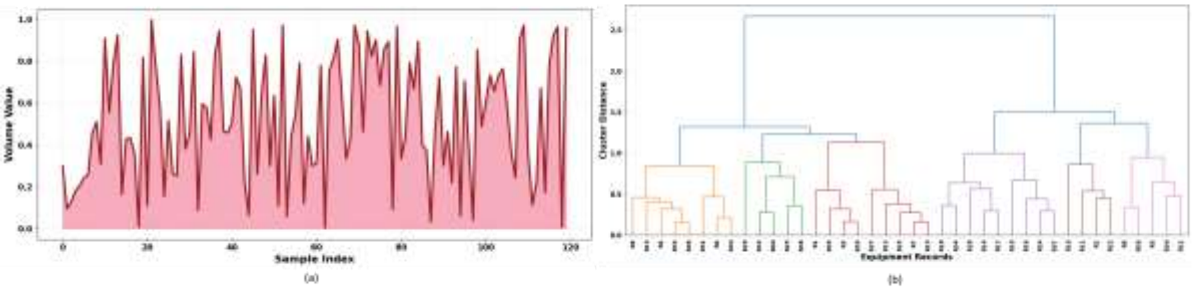

Figure 4. Result of proposed adaptive learning model for (a) stock market forecasting; (b) equipment fault clustering.

4.1 Experimental Performance Analysis

The proposed WPS-ARF model has been tested for its effectiveness by using the financial market data and monitoring data of industrial equipment with continuously changing patterns. Model accuracy, stability and computational efficiency under non-stationary were compared with Adaptive Machine Learning (AML), Random Forest (RF) and Support Vector Machine (SVM) models [18] to evaluate the models. The performance of the proposed model was tested on the following applications: financial market prediction, and monitoring of industrial equipment:

- **Mean Squared Error (MSE)** was used to measure the gap between the predicted values and the actual values of the market, and it was used to evaluate the forecasting accuracy.
- **Fault Detection Accuracy (FDA)** quantifies the ability of the model to detect faults in the machinery and determine whether abnormal operating conditions are due to a fault or merely the normal operation of the system.
- **Stability (Coefficient of Variation (CV))** tests the stability and consistency of the model's performance under constantly varying data conditions.
- **Response Time (s)** is the time needed to make predictions or identify anomalies, across both situations of use, which corresponds to the responsiveness of the model.

For the financial market forecasting, the measures evaluated were MSE, CV and RT; for the industrial equipment monitoring, the measures evaluated were FDA, CV, and RT.

4.1.1 Financial Market Forecasting and Industrial Equipment Monitoring Performance

The performance of the evaluated models in comparison with each other in the stock market prediction as shown in table 1 and figure 5 (a). The proposed model performed best with the lowest MSE at 0.010 while the other models with absolute improvements were a bit more than 0.050. Moreover, the highest forecasting stability (CV) was found as 3 in WPS-ARF and the shortest response time was 0.15 s. The results show that the optimization capability of WPSO in conjunction with the adaptive learning feature of ARF is capable of dealing with concept drift and rapidly changing market trends.

The performance comparison of the industrial fault detection is summarized in Table 1 and Figure 5 (b). The proposed model is able to obtain the highest value of fault detection accuracy of 98, which is higher than models. It also showed high stability (CV) value as 4 and minimum response time as 0.18 s. The results show that the proposed model can detect anomalies in the equipment in an early stage while keeping the performance level high under variable operating conditions and sensor behavior.

Table 1. Performance Comparison for Financial Market Forecasting and Industrial Equipment Monitoring

Model Type	Financial Market Forecasting			Industrial Equipment Monitoring		
	Mean Squared Error (MSE)	Stability (CV)	Response Time (s)	Fault Detection Accuracy	Stability (CV)	Response Time (s)
AML Model [18]	0.015	5	0.20	95	8	0.30
Random Forest [18]	0.020	10	0.40	80	18	0.60
SVM [18]	0.022	12	0.50	85	15	0.70
WPS-ARF [Proposed]	0.010	3	0.15	98	4	0.18

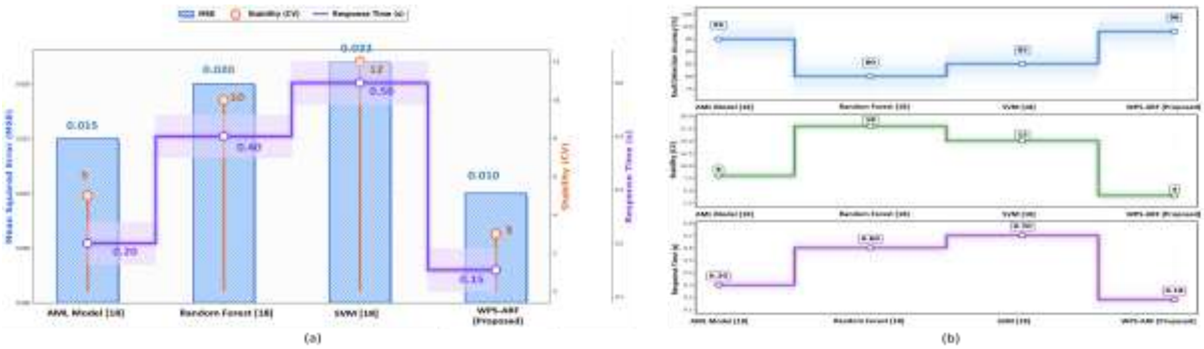

Figure 5. Comparative Analysis of Model Performance Trends for (a) Financial Market Forecasting and (b) IoT-Enabled Industrial Monitoring Systems

For a fair comparison, AML, RF and SVM [18] were coded and trained on the Stock Market Dataset and IoT-Enabled Industrial Equipment Monitoring Dataset with same preprocessing methods, train-test split and experimental setup as the proposed WPS-ARF model presented in Table 2. The presented results show the effectiveness of the proposed method in financial forecasting and industrial fault detection for each of the datasets separately.

Table 2. Performance Evaluation on Stock Market Dataset for AI-Driven Prediction and Trading Strategy Optimization

Model	Financial Market Dataset			IoT-Enabled Industrial Equipment Monitoring Dataset		
	MSE	CV	RT (s)	Fault Detection Accuracy	CV	RT (s)
AML	0.016	5	0.21	95	8	0.32
Random Forest	0.021	10	0.42	81	18	0.61
SVM	0.023	12	0.51	85	15	0.72
WPS-ARF [Proposed]	0.010	3	0.15	98	4	0.18

4.2 Discussion

The proposed WPS-ARF model combines ARF with WPSO to overcome the non-stationary behavior of financial and industrial data streams. Compared to conventional algorithms like RF and SVM [18] that are less robust to concept drift and AML [18] with limited adaptive optimization ability the proposed hybrid model allows continuous parameter adaptation in dynamic settings. In addition, PCA-based dimensionality reduction further reduces the calculation time, maintains the essential information structure, and WPSO improves the convergence behavior and efficient search of the parameters. The experimental results have shown that WPS-ARF outperforms benchmark models in accuracy, effectiveness, stability and response time issues, with the error rate of less than 0.05% and detection reliability of more than 98.95% on both financial and industrial data. These results show that it can be used for decision making with changing data distributions. Practical Implication: The model enhances financial forecasting accuracy and predictive maintenance in real-world streaming scenarios by enabling early fault detection, efficient response, and operational stability.

5 CONCLUSION

The growing amount of data in financial and industrial systems that is not stationary requires adaptive learning models that are able to operate in dynamic environments. This research introduced a WPS-ARF model combining Ada RF and WPS Optimization for forecasting and fault detection. The results show significant performance, with a financial forecasting error (MSE = 0.010) with stability (CV = 3), and response time (RT = 0.15 s), and industrial fault detection achieved (FDA = 98) with (CV = 4) and (RT = 0.18 s). These results validate the accuracy, stability and adaptability of these models over AML, RF and SVM. The current research, however, has been limited to two benchmark datasets and to experimental settings, which might not be representative of the large-scale volatility in the real world. Future studies will explore the potential of testing the system on a wide variety of real-time streaming data sets, incorporating adaptive elements in the deep learning model, and optimizing scalability in a more complex dynamic setting using hybrid approaches.

REFERENCES

1. Nadim, K., Ragab, A. and Ouali, M.S., 2023. Data-driven dynamic causality analysis of industrial systems using interpretable machine learning and process mining. Journal of Intelligent Manufacturing, 34(1), pp.57-83. https://doi.org/10.1007/s10845-021-01903-y
2. Khan, M.A., Waqas Khan, Q., Kim, J.E., Jeong, S., Ahn, I.Y. and Kim, D.H., 2026. From Physical to Virtual Sensors: VSG-SGL for Reliable and Cost-Efficient Environmental Monitoring. Automation, 7(1), p.27. https://doi.org/10.3390/automation7010027
3. Nagy, M., Valaskova, K. and Lazaroiu, G., 2026. Digitalization-based predictive maintenance systems, Internet of Things sensor networks, and machine intelligence algorithms for Industry 4.0/5.0 manufacturing processes. Systems, 14(3), p.281. https://doi.org/10.3390/systems14030281
4. Halstead, B., Koh, Y.S., Riddle, P., Pears, R., Pechenizkiy, M., Bifet, A., Olivares, G. and Coulson, G., 2022. Analyzing and repairing concept drift adaptation in data stream classification. Machine Learning, 111(10), pp.3489-3523. https://doi.org/10.1007/s10994-021-05993-w
5. Yang, Y., Zhou, W., Jiskani, I.M., Lu, X., Wang, Z. and Luan, B., 2023. Slope stability prediction method based on intelligent optimization and machine learning algorithms. Sustainability, 15(2), p.1169.https://doi.org/10.3390/su15021169
6. Dehaybe, H., Catanzaro, D. and Chevalier, P., 2024. Deep reinforcement learning for inventory optimization with non-stationary uncertain demand. European Journal of Operational Research, 314(2), pp.433-445. https://doi.org/10.1016/j.ejor.2023.10.007
7. Rieger, L.H., Flores, E., Nielsen, K.F., Norby, P., Ayerbe, E., Winther, O., Vegge, T. and Bhowmik, A., 2023. Uncertainty-aware and explainable machine learning for early prediction of battery degradation trajectory. Digital Discovery, 2(1), pp.112-122. https://doi.org/10.1039/D2DD00067A
8. Murti, F.W., Ali, S., Iosifidis, G. and Latva-aho, M., 2023. Deep reinforcement learning for orchestrating cost-aware reconfigurations of vRANs. IEEE Transactions on Network and Service Management, 21(1), pp.200-216. https://doi.org/10.1109/TNSM.2023.3292713
9. Xu, J., Jiang, X., Liao, S., Ke, D., Sun, Y. and Yao, L., 2023. High-dimensional feature selection for power system congestion event prognosis with enhanced evolutionary computation. IEEE Transactions on Power Systems, 39(1), pp.1752-1770. https://doi.org/10.1109/TPWRS.2023.3260871
10. Halstead, B., Koh, Y.S., Riddle, P., Pechenizkiy, M. and Bifet, A., 2023. Combining diverse meta-features to accurately identify recurring concept drift in data streams. ACM Transactions on Knowledge Discovery from Data, 17(8), pp.1-36. https://doi.org/10.1145/3587098
11. Hu, L., Li, W., Lu, Y. and Hu, C., 2024. Scalable concept drift adaptation for stream data mining. Complex & Intelligent Systems, 10(5), pp.6725-6743.https://doi.org/10.1007/s40747-024-01524-x
12. Gâlmeanu, H. and Andonie, R., 2022. Weighted Incremental–Decremental Support Vector Machines for concept drift with shifting window. Neural Networks, 152, pp.528-541.https://doi.org/10.1016/j.neunet.2022.05.018

13. Jain, M., Kaur, G. and Saxena, V., 2022. A K-Means clustering and SVM based hybrid concept drift detection technique for network anomaly detection. Expert Systems with Applications, 193, p.116510.https://doi.org/10.1016/j.eswa.2022.116510
14. Wang, Z., Han, D., Li, M., Liu, H. and Cui, M., 2022. The abnormal traffic detection scheme based on PCA and SSH. Connection Science, 34(1), pp.1201-1220.https://doi.org/10.1080/09540091.2022.2051434
15. Lara-Gutierrez, A., Fernandez-Gago, C. and Onieva, J.A., 2025. A Framework for Drift Detection and Adaptation in AI-driven Anomaly and Threat Detection Systems: A. Lara-Gutierrez et al. International Journal of Information Security, 24(5), p.199.https://doi.org/10.1007/s10207-025-01118-9
16. Moradi, M., Rahmanimanesh, M. and Shahzadi, A., 2024. Transfer learning for concept drifting data streams in heterogeneous environments. Knowledge and Information Systems, 66(5), pp.2799-2857.https://doi.org/10.1007/s10115-023-02043-w
17. Yang, L., Yao, Y., Yang, D., Yang, W. and Hao, Y., 2026. A Generalizable Anomaly Detection Framework with Dynamic Concept Drift Suppression for Non-stationary Time Series. Knowledge-Based Systems, 337, p.115380.https://doi.org/10.1016/j.knosys.2026.115380
18. Yang, Y., Zhao, H., Wan, J., Mingxia, F., Wang, Z., Liu, B. and Shi, G., 2024. Application of adaptive machine learning in non-stationary environments. Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online), 3(4), pp.68-77.https://doi.org/10.60087/jklst.vol3.n4.p68