International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 6, Issue 4, 241 -245, 2025
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V6I4P133

Original Article

Hybrid CNN–LSTM Models for Cross-Project Fault Prediction: Robust Generalization Under Dataset Shift

[1]Sidharth Rao, [2]Neha Gupta, [3]Arjun Patel, [4]Kavya Shah
[1,3]Computer Science Department, SRM Institute of Science Department and Technology, Chennai, Tamil Nadu, India.
[2,4]Information Technology, SRM Institute of Science Department and Technology, Chennai, Tamil Nadu, India.

Received On: 03/11/2025 Revised On: 06/12/2025 Accepted On: 16/12/2025 Published on: 30/12/2025

Abstract: *Cross-project fault prediction (CPFP) aims to learn defect predictors from one or more source projects and deploy them on a different target project where labels may be unavailable, incomplete, delayed, or noisy. Despite decades of progress in software defect prediction, generalization in CPFP remains brittle because target projects commonly exhibit dataset shift: changes in metric distributions, process behaviors, architectural patterns, operational platforms, and data pipelines that violate the implicit i.i.d. assumptions of standard supervised learning. This manuscript proposes a shift-aware hybrid CNN–LSTM framework tailored to CPFP under realistic dataset shift. The model combines convolutional feature extractors (for local, interaction-level patterns among engineered metrics or change descriptors) with recurrent sequence modeling (for temporal and cross-change dependencies), and integrates training strategies that stabilize performance under covariate and conditional shift. We formalize the CPFP learning objective under shift, define a practical shift taxonomy grounded in modern enterprise software contexts, and introduce a training pipeline using robust scaling, project-conditional normalization, and shift-mitigation regularization. The approach is designed for operational feasibility: it supports constrained edge environments and embedded telemetry streams, aligns with CI/CD governance processes, and incorporates explainability requirements for high-stakes domains. A worked example illustrates how dataset shift can invert model rankings across projects and how shift-aware training reduces this volatility. The manuscript concludes with a reproducible evaluation protocol, threat analysis, and deployment guidance for fault prediction in cloud-native and regulated settings.*

Keywords: *Cross-Project Fault Prediction, Dataset Shift, Domain Generalization, Hybrid Deep Learning, CNN, LSTM, Software Defect Prediction, Ci/Cd Analytics, Robustness, Explainable AI.*

1. Introduction

The practical value of defect prediction is highest when a model trained on existing projects can be reused for a new or rapidly evolving project without requiring extensive labeled history. This setting is precisely the goal of cross-project fault prediction (CPFP), yet it is also where predictive systems most often fail in practice due to dataset shift. Dataset shift broadly refers to any mismatch between the training distribution and the deployment distribution that alters model behavior in nontrivial ways, including changes in feature marginals, label priors, or conditional relationships between features and outcomes [1]. In CPFP, shift is not an exceptional corner case: it is routine, because target projects differ in codebase characteristics, development practices, operational platforms, security constraints, compliance obligations, and data pipelines.

Transfer learning provides a conceptual foundation for CPFP by formalizing how knowledge learned from source domains can improve performance in a target domain under distribution mismatch [2]. However, CPFP typically operates under additional constraints: limited target labels, different measurement conventions across projects, and varying defect definitions. Large-scale empirical evidence from cross-project experiments has shown that naïvely transferring defect predictors can result in highly variable outcomes, including strong performance on some targets and near-random performance on others [3].

Modern CPFP increasingly explores deep learning models to reduce manual feature engineering and to better capture nonlinear dependencies. Convolutional neural networks (CNNs) are well suited for extracting local patterns and interaction structures [4]. Long short-term memory (LSTM) networks are effective for capturing temporal dependencies and long-range interactions in sequential data [5]. Yet deep models do not automatically solve CPFP under dataset shift; in fact, they can amplify failures when trained representations latch onto spurious correlations specific to source projects.

Real-world shift is also driven by operational factors beyond code metrics. Edge and embedded systems increasingly perform real-time environmental data processing with lightweight AI, producing telemetry streams whose characteristics depend on device constraints and on-device preprocessing [6]. At the enterprise layer, reliability engineering practices—including automated testing frameworks in Java systems—shape defect discovery and reporting processes [7]. In data-intensive architectures, predictive monitoring and error mitigation in change data capture pipelines introduces additional distribution changes as the pipeline evolves or as remediation policies are updated [8]. Comparative studies of defect prediction models further show that algorithm rankings depend strongly on the dataset and context [9].

Tooling trends also matter. Democratized AutoML systems reduce the barrier to model construction and tuning, but they can inadvertently overfit to the source project if shift is not explicitly handled [10]. Intelligent CI/CD risk detection in banking environments demonstrates that outcomes may be tightly coupled to deployment practices and governance [11]. Unified real-time integration across heterogeneous data pipelines can create multi-source shifts that ripple into downstream predictive tasks [12]. Explainability requirements for regulatory-grade decision making and high-stakes contexts further motivate interpretability [13], [14].

Healthcare and pharmacy operations provide concrete examples of cross-domain and cross-platform shift. Cloud-native OCR and microservices pipelines used for fax-to-digital prescription processing include multiple error modes spanning OCR, extraction, and integration layers [15]. Deep learning frameworks for emotion understanding in human–computer interaction show that learned representations can be domain-sensitive [16]. Decision intelligence methodologies emphasize that model deployment must align with architecture-centered governance [17]. Privacy constraints increasingly push organizations toward decentralized learning; federated learning introduces new shift dynamics due to client heterogeneity [18], [19]. Secure microservices architectures for compliant processing demonstrate that infrastructure choices influence data flow, latency, and failure modes [20].

Operational pipelines are shaped by optimization objectives and automation. AI-augmented CI/CD optimization emphasizes that predictive recommendations should consider deployment constraints [21]. Evaluations of CNN and RNN models for fault prediction underscore the need to test robustness under changing conditions [22]. Enterprise transformations such as SAP transitions change module boundaries and performance profiles, often invalidating historical predictors [23]. Alternative representation learning, such as scalable graph neural networks, suggests future directions for incorporating structural information [24].

Platform comparisons between Pivotal Cloud Foundry and OpenShift show that migration decisions can change deployment mechanics and observability—common drivers of dataset shift [31]. Monitoring and deployment optimization in cloud-native systems likewise changes defect detection timing [33]. Covariate shift adaptation provides theoretically grounded mitigation methods such as importance reweighting [34]. Architecture transitions from monoliths to microservices introduce structural changes that can invalidate historical predictors [35].

Problem Statement: Given labeled historical data from one or more source projects and limited or no labels from a target project, learn a fault prediction model that maintains stable, calibrated performance under dataset shift. Research Gap: Existing CPFP approaches often optimize performance under implicit stationarity or evaluate transfer on narrow shift scenarios. There is a need for hybrid architectures that jointly model local metric interactions and temporal change dependencies, and for training objectives explicitly designed for shift robustness and deployability in modern enterprise contexts.

2. Background and Motivation

Cross-project defect prediction differs from within-project prediction in that the training and deployment distributions are usually mismatched. Large-scale experiments have shown that transfer success is inconsistent across targets [3]. This inconsistency is expected under dataset shift due to measurement, process, platform, architectural, and pipeline differences [1], [7], [31], [35].

Hybrid CNN–LSTM modeling is motivated by the complementary strengths of CNNs for local interaction patterns [4] and LSTMs for temporal dependencies [5]. Temporal signals can be more transferable than raw feature magnitudes, improving robustness under scale and process changes. Shift-aware learning is an operational requirement because enterprise systems evolve continuously through deployment optimization [21], monitoring policy changes [33], and remediation automation [26]. In high-stakes domains, robustness and transparency must be co-designed [13], [14].

3. Problem Statement and Research Gap

Let S = {P1, …, Pm} be labeled source projects providing samples (x, y) and Pt be a target project with unlabeled or partially labeled samples. CPFP seeks a predictor that minimizes expected target risk under distribution mismatch $p\_s(x, y) \neq p\_t(x, y)$ [1].

Three gaps limit CPFP practice: (1) representation gap from relying on static metrics rather than temporal development dynamics [5]; (2) shift robustness gap from optimizing objectives for source performance rather than target stability [34]; and (3) operational gap where offline evaluations ignore CI/CD workflows, migrations, and explainability requirements [11], [31], [14].

4. Proposed Hybrid CNN–LSTM Framework

Input representations include (i) snapshot vectors of static and process metrics and (ii) sequence representations over commits, sprints, or release windows. Sequence representations support temporal robustness and integration of operational signals [33]. The architecture includes a CNN feature extractor to learn local feature interactions [4], an LSTM temporal model to capture long-range dependencies [5], and a prediction head producing calibrated defect probabilities for triage [25].

Shift-aware training optimizes supervised loss plus representation alignment regularization and calibration penalties. Project-conditional normalization, robust scaling, and latent moment matching are used to reduce project-specific overfitting [34]. Explainability hooks provide feature-level attributions and temporal saliency maps, supporting auditability and high-stakes interpretability requirements [13], [14].

5. Robust Generalization under Dataset Shift

Shift modes include covariate shift, label shift, and conditional shift. Platform migrations, architectural transitions, and process changes induce conditional shift in CPFP [31], [35]. Covariate shift mitigation can be implemented via importance weighting using density ratio estimation [34]. Robust scaling and project-conditional normalization reduce sensitivity to measurement differences and outliers. Feedback loops from remediation can change the distribution of observed defects and labels over time; drift monitoring and recalibration are therefore essential [26], [8].

6. Integration with Enterprise Software Lifecycles

Integration into CI/CD is critical for operational value. Risk detection in banking CI/CD highlights that predictive models must align with governance and decision points [11], while CI/CD optimization emphasizes deployment constraints and platform conditions [21]. Decision intelligence methodologies provide a governance structure for incorporating predictive outputs [17]. Platform and architecture transitions are major shift drivers. Migration studies and cloud-native monitoring optimization show that observability and failure surfaces change, affecting defect detection and labels [31], [33]. Microservices transitions further induce conditional shift requiring robustness-first modeling [35]. Regulated contexts require explainability and security-by-design. HIPAA-compliant architectures and secure fax communication add specialized monitoring and failure patterns [20], [29]. Explainable AI frameworks support auditable decision pathways [13], [14].

7. Worked Example: How Dataset Shift Breaks CPFP and How Shift-Aware Training Helps

Consider two source projects with different testing cultures. When defect labels reflect testing and reporting processes, a model trained on one project may miscalibrate risk on a target project with different practices [7]. Platform migrations can further alter monitoring and deployment mechanics, inducing additional shift [31], [33]. A hybrid CNN–LSTM model mitigates this through local interaction learning [4], temporal modeling [5], representation alignment, covariate shift weighting [34], and calibrated outputs aligned with governance workflows [17].

8. Evaluation Protocol for Reproducible CPFP under Shift

A robust evaluation includes single-source and multi-source transfer, temporal splits to simulate drift, and stratification by platform or architecture when metadata is available [31], [35]. Metrics should include AUC-ROC, AUC-PR, top-k precision for triage, and calibration measures such as expected calibration error [37]. Baselines should include classical defect prediction models [9] and deep CNN-only and RNN-only variants [22], with ablation studies for each shift-mitigation component. Operational relevance requires incorporating process and deployment signals [11], [33] and explainability evaluation for high-stakes settings [13], [14].

9. Discussion

CPFP failures often arise because deployment contexts change faster than learning assumptions. Continuous pipeline optimization, monitoring evolution, and remediation automation induce drift and shift [21], [33], [26]. Robustness-first CPFP requires shift detection, adaptation with minimal labels, calibration, and interpretability. AutoML can accelerate adoption but risks silent overfitting if cross-project validation and stability objectives are not used [10]. Federated learning is relevant under privacy constraints but introduces client heterogeneity that reinforces the need for alignment and calibration [18], [19]. Structural learning via scalable graph neural networks may enhance representation of dependencies, but generalization under dataset shift remains essential [24].

10. Threats to Validity

Construct validity is threatened when defect labels reflect reporting and testing processes rather than intrinsic fault-proneness [7], [11]. Internal validity can be affected by leakage when target data is used for weighting or normalization. External validity depends on domain diversity, including regulated healthcare pipelines [15], [20] and enterprise platform transitions [31]. Conclusion validity depends on metric choice and shift scenario.

11. Conclusion and Future Work

This manuscript proposed a shift-aware hybrid CNN–LSTM framework for CPFP under dataset shift, combining local interaction learning and temporal modeling with alignment and calibration mechanisms. The approach aligns with modern enterprise realities including platform migrations, monitoring optimization, remediation feedback loops, and explainability requirements [31], [33], [26], [14]. Future work includes federated training for cross-organization CPFP [18], [19], integrating structural representations [24], drift detection in CI/CD workflows [11], [21], and richer explainability evaluations for compliance audits [13], [14].

References

[1] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, Dataset Shift in Machine Learning. Cambridge, MA, USA: MIT Press, 2009.

[2] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[3] T. Zimmermann, N. Nagappan, H. Gall, E. Giger, and B. Murphy, “Cross-project defect prediction: a large-scale experiment on data from one project,” in Proc. 7th Joint Meeting Eur. Softw. Eng. Conf. ACM SIGSOFT Symp. Found. Softw. Eng. (ESEC/FSE), 2009, pp. 91–100.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[5] S. K. Gunda, "Analyzing Machine Learning Techniques for Software Defect Prediction: A Comprehensive Performance Comparison," 2024 Asian Conference on Intelligent Technologies (ACOIT), KOLAR, India, 2024, pp. 1-5, https://doi.org/10.1109/ACOIT62457.2024.10939610.

[6] I. Manga, “Edge Software Engineering for Lightweight AI: Real-Time Environmental Data Processing with Embedded Systems,” Journal of Computational Analysis and Applications (JoCAAA), vol. 34, no. 6, pp. 88–104, Jun. 2025.

[7] S. R. Gudi, “Enhancing Reliability in Java Enterprise Systems through Comparative Analysis of Automated Testing Frameworks,” International Journal of Emerging Trends in Computer Science and Information Technology, vol. 4, no. 2, pp. 151–160, 2023, doi: 10.63282/3050-9246.IJETCSIT-V4I2P115.

[8] V. K. Reddy Mittamidi, “Leveraging AI and ML for Predictive Monitoring and Error Mitigation in Change Data Capture Pipelines,” IJETCSIT, vol. 6, no. 3, pp. 104–111, Aug. 21, 2025. https://ijetcsit.org/index.php/ijetcsit/article/view/515

[9] S. K. Gunda, “Comparative Analysis of Machine Learning Models for Software Defect Prediction,” in Proc. Int. Conf. Power, Energy, Control and Transmission Systems (ICPECTS), Chennai, India, 2024, pp. 1–6, doi: 10.1109/ICPECTS62210.2024.10780167.

[10] I. Manga, “AutoML for All: Democratizing Machine Learning Model Building with Minimal Code Interfaces,” in Proc. 3rd Int. Conf. Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 2025, pp. 347–352, doi: 10.1109/ICSCDS65426.2025.11167529.

[11] R. R. Thalakanti and S. S. Goud Bandari, “Intelligent Continuous Integration and Delivery for Banking Systems using Machine Learning Driven Risk Detection with Real World Deployment Evaluation,” International Journal of AI, BigData, Computational and Management Studies, vol. 5, no. 4, pp. 168–175, 2024, doi: 10.63282/3050-9416.IJAIBDCMS-V5I4P118.

[12] I. Manga, “Unified Data Engineering for Smart Mobility: Real-Time Integration of Traffic, Public Transport, and Environmental Data,” in Proc. 5th Int. Conf. Soft Computing for Security Applications (ICSCSA), Salem, India, 2025, pp. 1348–1353, doi: 10.1109/ICSCSA66339.2025.11170800.

[13] S. S. G. Bandari, S. D. Sivva, and R. R. Thalakanti, “Regulatory Grade Fraud Detection using Explainable Artificial Intelligence with Auditable Decision Pathways and Empirical Validation on Banking Data,” International Journal of Artificial Intelligence, Data Science, and Machine Learning, vol. 5, no. 3, pp. 139–147, 2024, doi: 10.63282/3050-9262.IJAIDSML-V5I3P115.

[14] I. Manga, “Towards Explainable AI: A Framework for Interpretable Deep Learning in High-Stakes Domains,” in Proc. 5th Int. Conf. Soft Computing for Security Applications (ICSCSA), Salem, India, 2025, pp. 1354–1360, doi: 10.1109/ICSCSA66339.2025.11170778.

[15] S. R. Gudi, “AI-Driven Fax-to-Digital Prescription Automation: A Cloud-Native Framework Using OCR, Machine Learning, and Microservices for Pharmacy Operations,” International Journal of Emerging Research in Engineering and Technology, vol. 5, no. 1, pp. 111–116, 2024, doi: 10.63282/3050-922X.IJERET-V5I1P113.

[16] G. V. Krishna, B. D. Reddy, and T. Vrindaa, “EmoVision: An Intelligent Deep Learning Framework for Emotion Understanding and Mental Wellness Assistance in Human Computer Interaction,” IJAIDSML, vol. 6, no. 4, pp. 14–20, Oct. 2025. https://ijaidsml.org/index.php/ijaidsml/article/view/295

[17] S. K. Gunda, S. D. R. Yettapu, S. Bodakunti, and S. B. Bikki, “Decision Intelligence Methodology for AI-Driven Agile Software Lifecycle Governance and Architecture-Centered Project Management,” IJAIDSML, vol. 4, no. 1, pp. 102–108, Mar. 30, 2023, doi: 10.63282/3050-9262.IJAIDSML-V4I1P112.

[18] I. Manga, “Federated Learning at Scale: A Privacy-Preserving Framework for Decentralized AI Training,” in Proc. 5th Int. Conf. Soft Computing for Security Applications (ICSCSA), Salem, India, 2025, pp. 110–115, doi: 10.1109/ICSCSA66339.2025.11170780.

[19] R. R. Thalakanti, S. S. Goud Bandari, and S. D. Sivva, “Federated Learning for Privacy Preserving Fraud Detection across Financial Institutions: Architecture Protocols and Operational Governance,” International Journal of Emerging Research in Engineering and Technology, vol. 5, no. 2, pp. 108–114, 2024, doi: 10.63282/3050-922X.IJERET-V5I2P111.

[20] S. R. Gudi, “Design and Evaluation of Secure Microservices Architecture for HIPAA-Compliant Prescription Processing on AWS and OpenShift,” International Journal of Artificial Intelligence, Data Science, and Machine Learning, vol. 5, no. 2, pp. 144–149, 2024, doi: 10.63282/3050-9262.IJAIDSML-V5I2P116.

[21] S. K. RaoVanama, “AI-Augmented CI/CD Pipeline Optimization for Scalable Cloud-Native Deployment,” IJAIDSML, vol. 5, no. 4, pp. 175–187, Dec. 30, 2024. https://ijaidsml.org/index.php/ijaidsml/article/view/368

[22] S. K. Gunda, “A Deep Dive into Software Fault Prediction: Evaluating CNN and RNN Models,” in Proc. Int. Conf. Electronic Systems and Intelligent Computing (ICESIC), Chennai, India, 2024, pp. 224–228, doi: 10.1109/ICESIC61777.2024.10846549.

[23] T. Raikar and V. Apelagunta, “Implementing SAP Fiori in S/4HANA transitions: Key guidelines, challenges,

strategic implications, AI integration recommendations," Journal of Engineering Research and Sciences, vol. 4, no. 11, pp. 1–9, 2025, doi: 10.55708/JS0411001.

[24] I. Manga, "Scalable Graph Neural Networks for Global Knowledge Representation and Reasoning," in Proc. 9th Int. Conf. Inventive Systems and Control (ICISC), Coimbatore, India, 2025, pp. 1399–1404, doi: 10.1109/ICISC65841.2025.11188341.

[25] S. R. Gudi, "Leveraging Predictive Analytics and Redis-Backed Caching to Optimize Specialty Medication Fulfillment and Pharmacy Inventory Management," International Journal of AI, BigData, Computational and Management Studies, vol. 5, no. 3, pp. 155–160, 2024, doi: 10.63282/3050-9416.IJAIBDCMS-V5I3P116.

[26] V. K. Reddy Mittamidi, "AI/ML Powered Intelligent Root Cause Analysis and Automated Remediation for Multi System Data Integrity Issues," IJAIBDCMS, vol. 6, no. 4, pp. 133–141, Nov. 14, 2025. https://ijaibdcms.org/index.php/ijaibdcms/article/view/338

[27] S. K. Gunda, "The Future of Software Development and the Expanding Role of ML Models," International Journal of Emerging Research in Engineering and Technology, vol. 4, no. 2, pp. 126–129, 2023, doi: 10.63282/3050-922X.IJERET-V4I2P113.

[28] T. Raikar, "High-Performance In-Memory Computing: A Research Study on SAP S/4 HANA Database Layer," American Journal of Technology, vol. 4, no. 2, pp. 93–113, 2025, doi: 10.58425/ajt.v4i2.449.

[29] S. R. Gudi, "Ensuring Secure and Compliant Fax Communication: Anomaly Detection and Encryption Strategies for Data in Transit," in Proc. 4th Int. Conf. Innovative Mechanisms for Industry Applications (ICIMIA), Tirupur, India, 2025, pp. 786–791, doi: 10.1109/ICIMIA67127.2025.11200537.

[30] K. V. Alluri, "Using Salesforce CRM and Deep Learning (CNN) Techniques to Improve Patient Journey Mapping and Engagement in Small and Medium Healthcare Organizations," IJAIDSML, vol. 6, no. 4, pp. 101–109, Nov. 22, 2025. https://ijaidsml.org/index.php/ijaidsml/article/view/330

[31] S. R. Gudi, "A Comparative Analysis of Pivotal Cloud Foundry and OpenShift Cloud Platforms," The American Journal of Applied Sciences, vol. 7, no. 07, pp. 20–29, 2025, doi: 10.37547/tajas/Volume07Issue07-03.

[32] R. R. Thalakanti, "Enhancing Convergence in Fully Connected Neural Networks via Optimized Backpropagation," in Proc. 2nd Int. Conf. Computing and Data Science (ICCDS), Chennai, India, 2025, pp. 1–6, doi: 10.1109/ICCDS64403.2025.11209625.

[33] S. R. Gudi, "Monitoring and Deployment Optimization in Cloud-Native Systems: A Comparative Study Using OpenShift and Helm," in Proc. 4th Int. Conf. Innovative Mechanisms for Industry Applications (ICIMIA), Tirupur, India, 2025, pp. 792–797, doi: 10.1109/ICIMIA67127.2025.11200594.

[34] Khokrale, R. (2025). The Role of AI in Supply Chain Optimization: Enhancing Efficiency through Predictive Analytics. Journal of Procurement and Supply Chain Management, 4(2), 55–75. https://doi.org/10.58425/jpscm.v4i2.442

[35] S. R. Gudi, "Deconstructing Monoliths: A Fault-Aware Transition to Microservices with Gateway Optimization using Spring Cloud," in Proc. 6th Int. Conf. Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2025, pp. 815–820, doi: 10.1109/ICESC65114.2025.11212326.

[36] K. V. Alluri, "Salesforce CRM Framework for Real Time DeFi Portfolio Intelligence and Customer Engagement Forecasting in Web3 Based Decentralized Finance Ecosystems Using ML Techniques," IJAIBDCMS, vol. 6, no. 4, pp. 99–107, Nov. 6, 2025. https://ijaibdcms.org/index.php/ijaibdcms/article/view/319

[37] S. R. Gudi, "Enhancing optical character recognition (OCR) accuracy in healthcare prescription processing using artificial neural networks," European Journal of Artificial Intelligence and Machine Learning, vol. 4, no. 6, 2025, doi: 10.24018/ejai.2025.4.6.79.

[38] Sivva SD, Thalakanti RR, Bandari SSG, Yettapu SDR. AI-Driven Decision Intelligence for Agile Software Lifecycle Governance: An Architecture-Centered Framework Integrating Machine Learning Defect Prediction and Automated Testing. IJETCSIT 2023 Dec.30;4(4):167-72. Available from: https://ijetcsit.org/index.php/ijetcsit/article/view/554.