

Psychometric Validation of Stress–Resilience Constructs for eXplainable Software Developer Digital Twin

Taiwo Olapeju Olaleye

agsobaolaleyetaiwo@gmail.com

Federal university of Agriculture Abeokuta <https://orcid.org/0000-0001-6222-7575>

Oluwasefunmi Arogundade

federal university of agriculture abeokuta

Taiwo David Ajayi

Federal University of Agriculture Abeokuta

Research Article

Keywords: Psychometric Validation, SDG3, Ensemble XGBoost, Explainable AI, Human Digital Twin, Software Development, SD5, unsupervised K-Means

Posted Date: September 9th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10943597/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Psychometric Validation of Stress–Resilience Constructs for eXplainable Software Developer Digital Twin

Taiwo O. Olaleye ¹[0000-0001-6222-7575], Oluwasefunmi T. Arogundade ²[0000-0001-9338-491X], Taiwo. D. Ajayi ³[0009-0004-7920-9700]

Abstract

Software development environments expose developers to sustained technical demands, rapid technological change, workload pressures, and interpersonal challenges that may affect mental health, resilience, and service delivery and developer attrition. Although Human Digital Twins (HDTs) offer opportunities for well-being tracking, existing solutions lack psychometric validation of the behavioural data layers used to represent human states. This study addresses this gap by developing a psychometrically grounded HDT for modelling developer stress–resilience states and simulating personalised interventions. The proposed pipeline integrates survey-based feature engineering, exploratory and confirmatory factor analysis, reliability and validity assessment, unsupervised state learning, XGBoost prediction, and eXplainable counterfactual simulation. Using data from 605 female software developers, the measurement analysis demonstrated strong sampling adequacy ($KMO = 0.942$; Bartlett's $\chi^2 = 3217.824$, $p < 0.001$), while K-Means identified High-Stress Resilience (HSR) and Low-Stress Moderate Resilience (LSMR) dominance. The initial scenario simulation revealed framework's ability to distinguish state-preserving and state-transition interventions for HSR and LSMR situations respectively, while consistently identifying Openness–Adaptability as the intervention lever. Counterfactual simulation however shows that interventions that amplified resilient features raised the predicted probability of HSR by User Empathy (+0.2027), Perspective Diversity (+0.1691), Goal Tracking (+0.1268), Technical Innovation (+0.1200), Work Discipline (+0.1181), and Flexibility (+0.0779). Probability improvement ranged from 0.0004 to 0.9176, demonstrating that equal simulated intervention strategy did not produce equivalent benefits across developers. Findings revealed the ability of a construct-validated DT framework to reliably move developer stress–resilience assessment beyond adaptive management towards a reliable and interpretable intervention-oriented understanding of instance-based driving factors.

Keywords: Psychometric Validation, SDG3, Ensemble XGBoost, Explainable AI, Human Digital Twin, Software Development, SD5, unsupervised K-Means

1. Introduction

Persistent cognitive demand, chronic role overload and high accountability have been highlighted as essential job descriptions in software development profession. In most cases, tight deadlines, continuous technical shifts, and complex architectures also predisposes developers to stress even when guided by structured software development process models like Agile, which is reputable for sprint-based schedules. Understanding the drivers of developer stress and resilience is therefore crucial, necessitating moving beyond simply predicting who is at risk to unravelling why the state occurs and how it can be influenced. Conventional approaches, including regression, classification,

and other predictive analytics, primarily estimate outcomes from observed patterns without essentially modelling the individual's evolving state. A Digital Twin (DT), however, can maintain a dynamic representation of a developer (Chen et al., 2025) and model his state-awareness, as well as intervention-oriented analysis (Kumi et al., 2025). This would allow the manipulation of potentially influential stress–resilience drivers and consequently avail the examination of their effects on predicted states before decision support. Such adaptive modelling is particularly valuable where stress and resilience emerge from interacting technical, behavioural, and human-centred factors rather than isolated predictors. Thus, stress-resilience management framework would be better than a predictive system, but an adaptive decision-support and intervention simulation framework. However, the reliability of a DT framework is fundamentally dependent on the quality of the feature space used to represent its physical counterpart (Chen et al., 2025). Therefore, the reliability of such attributes must be ascertained prior to modelling. Although conventional data pre-processing procedures such as missing-value treatment, normalization, outlier detection, correlation analysis, and multicollinearity diagnostics are useful for identifying statistical irregularities and redundant predictors (Olaleye et al., 2025), they do not establish whether observed variables adequately represent the underlying theoretical constructs of domain case study (Hossain et al., 2025). Similarly, feature-selection and dimensionality-reduction techniques such as Principal Component Analysis improves computational efficiency, they potentially obscure the latent conceptual structure required for meaningful DT representation (Wen et al., 2022). Consequently, statistical cleanliness alone cannot guarantee that a feature space provides a valid representation of the phenomena being modelled (Henley et al., 2020).

This study argues that a validated psychometric feature space embedded within a DT can move developer stress–resilience assessment beyond static predictive classification toward an adaptive, interpretable, and intervention-oriented understanding of the factors that drive developers' resilience states. Existing predictive approaches can classify stress or resilience states with reasonable accuracy but often provide limited evidence that the underlying features are psychometrically sound, structurally valid, and actionable for understanding why developers occupy particular states or how those states may respond to interventions. The main aim is therefore to develop and psychometrically validate a multidimensional feature space for an adaptive Developer Stress–Resilience DT capable of identifying, interpreting, and supporting the management of stress–resilience states. Specific objectives include to:

1. Establish the reliability of the constructs and the convergent and discriminant validities of proposed stress–resilience feature space.
2. Identify and quantify the latent factors and key features that significantly characterise developers' stress–resilience states.
3. Integrate the validated feature space into a DT framework for personalised state assessment, intervention and counterfactual simulation, and adaptive stress–resilience management.

Exploratory Factor Analysis (EFA) is first employed to investigate the underlying dimensionality of the feature space and confirm the convergence of observed indicators into theoretically meaningful latent constructs. This is supported by sampling adequacy and factorability diagnostics. Reliability is subsequently assessed to establish the internal consistency of the resulting constructs while avoiding sole dependence on the assumptions underlying alpha. Thereafter, Confirmatory Factor Analysis (CFA) is employed to test the empirically derived measurement structure. These will provide stronger evidence that the feature space is not merely

statistically tractable but also reliable, construct-valid, and structurally interpretable. Within a DT framework, such validation is particularly important because the feature space could subsequently support state classification, eXplainable Artificial Intelligence (XAI), scenario simulation, and personalised intervention recommendation. The psychometric validation therefore serves as a methodological bridge between raw developer-level observations and their transformation into a defensible computational representation of developer stress and resilience, strengthening the empirical foundation of the proposed Developer DT modelling framework. Our contribution therefore is the integration of validated latent constructs, state discovery, explainable prediction and counterfactual intervention simulation into a one-stop individual-level Developer Stress–Resilience DT. The study is logically structured. Recent related patterns in literature are discussed in chapter 2 and our methodological approach is presented in chapter 3. Results are presented and discussed in chapter 4 with implications for stress management, female developer well-being

2. Literature Review

Stress and stress–resilience are now largely viewed as dynamic and multidimensional concepts rather than workload constructs alone. In software development, stress may arise when technical, cognitive, emotional, social and organisational demands exceed an individual’s perceived capacity (Kumi et al., 2025) and available resources to cope (Chen and Xie, 2025). Stress–resilience therefore reflects the individual’s capacity to maintain or regain adaptive functioning under such demands (Liu et al., 2024), shaped by several validated factors in literature. Contemporary research consequently views developer stress and resilience as an interacting system of risk and defensive factors, providing a foundation for predictive and intervention-oriented modelling. In several cases, DT has been largely presented as the central processing unit of recent data-driven conceptual frameworks designed to improve on physical entities for future possibilities. It is described as the virtual representation of a physical object through real time update of data layer (Chen et al., 2025). It is also capable of simulating invented data on the current virtual twin in order to investigate future behavioural possibilities (Kumi et al., 2025). It is constructed across use cases to improve the accuracy of maintenance of the physical entity (AR et al., 2025) or the performance predictions for individual products (Kobryn 2020). The possibility of the virtual twin to evolve over the lifecycle of the system through counterfactual simulation or real-time feature update supports the derivation of solutions or insights which assist the real time optimization of the physical copy (Boschert and Rosen 2016). Extend uses cases have resulted into what has been widely described as the Human Digital Twin (HDT). The paradigm has been widely applied in diverse fields, including humanitarian services (Lin et al., 2024), medical science (Corral-Acero et al. 2020), sports performance enhancement (Barricelli et al. 2020), manufacturing biotechnology (Sharotry et al. 2020), product design optimization (Demirel et al. 2021), etc. HDT indeed outspreads the traditional DT from physical system representation to individual human subjects, where the twin represents a dynamic, data-driven virtual twin of a human being whose behavioural, physical, psychosocial, and contextual characteristics can be continuously modelled, simulated, optimized and updated (Olaleye et al., 2026). In this context, the HDT represents an adaptive representation of the individual; enabling state monitoring, factors that drive observed changes, scenario simulation, personalised decision support, and feedback-driven twin updating. The application of HDT for stress management is therefore an evolving study area across social sciences and empirical software engineering. This is highly motivated by the need to address the inherent stress-

induced job descriptions across the entire software development life cycle, notwithstanding the software development process model in operation.

2.1 Stress–Resilience predictors in existing literature

Recent literature largely views software-developer stress and well-being as multidimensional concept stemming from the interrelatedness of developer's psychological, job, team and organisational dynamics. According to the study by Godliauskas and Šmite (2025), a review on 16,086 professionals from 42 countries established well-being in software engineering as a meta-construct of pleasure maximization (hedonic), purposeful living (eudaimonic), and integrated (hedaimonic) qualities. At the occupational level, the study identified high workload and time pressure, tight deadlines, task interdependence, and uncertainty, etc. consistently as stress-related conditions. Tulili et al. (2023) had through a systematic mapping study on software-engineering burnout identified work tension, job overload and job demands as important contributors to burnout risk. Recently, Palacio et al. (2026) work on Agile software development additionally highlighted coding-induced pressure, sleep deprivation and balancing job demands and work control as germane components of the stress–resilience relationship. Autonomy and work control (Zeuge et al., 2023), emotional intelligence (Rezvani and Khosravi, 2019), social support and interpersonal competence (Zhang et al., 2025), positive team interaction, adaptability and opportunities for skill development (Suárez and Vizcaíno, 2024) were also identified as factors that can buffer adverse effects of demanding software work. In particular, Chen and Xie (2025) opined that emotionally intelligent developers have been associated with stress management and resiliency abilities. Notably, recent evidence indicates that these stress-induced and resiliency dimensions can coexist within workspace dynamics. For example, software professionals may experience demanding technical environments while also benefiting from autonomy, engagement, supportive teams and meaningful work. Consequently, contemporary research is moving from single-factor stress explanations toward multidimensional and interaction-sensitive models (Perlo et al., 2026).

2.2 Stress predictor and resiliency validation and measurement approaches

Recent stress and resilience studies employ a combination of established psychometric instruments and empirical construct-validation procedures. Common measurement approaches include perceived-stress, burnout, anxiety/depression, work-demand, psychosocial-work and resilience scales, while newer studies increasingly evaluate whether such instruments retain adequate validity within specific occupational and demographic contexts. A 2025 systematic review of work-related stress assessment by Kallio et al. (2025) concluded that there is no single gold-standard stress measure, and recommended consideration of complementary subjective, external and biological contexts. Psychometric validation commonly progresses from item assessment and factorability testing to EFA/CFA, reliability estimation and convergent/discriminant validity assessment. For example, Wang et al. (2025) had employed Kaiser-Meyer-Olkin (KMO) and Bartlett's test before EFA, followed by factor extraction and reliability assessment. Contemporary resilience-scale validation has additionally reported Cronbach's alpha, McDonald's omega, composite reliability (CR) and average variance extracted (AVE) as handy construct validation tools (Srisuk et al., 2026). This methodological progression is important because predictor identification without measurement validation can confound substantive relationships with measurement error (Luijken et al., 2019). The present study therefore follows the emerging direction of combining

psychometric validation with computational prediction rather than treating questionnaire variables as automatically valid predictors.

2.3 Data analysis tools and computational approaches in recent studies

The analytical landscape has also shifted from predominantly descriptive and correlational approaches toward multivariate, longitudinal and predictive analytics methods. The Godliauskas and Šmite (2025) systematic review study found structural equation modelling and regression analysis to be the most frequently used quantitative approaches, followed by multilevel modelling and correlation analysis. In parallel, software-engineering research is increasingly exploiting digital traces and computational data, including software repositories (Samaraweera et al., 2026), developer communications (Fariha et al., 2025), issue reports (Obi et al., 2025), chat messages (Sajadi et al., 2023) and other software artefacts. Systematic mapping studies largely reported approaches toward data-driven methods, with machine learning becoming increasingly prominent for stress-resilience detection. Sentiment-analysis research has similarly demonstrated the use of neural networks (Samaraweera et al., 2026) and transformer-based approaches such as BERT (Fariha et al., 2025) for extracting affective information from developer communication. Recent stress-prediction research provides particularly strong evidence for the value of ensemble machine learning combined with explainability. Bozorgmehr and Weltermann (2023) used XGBoost to classify chronic stress and Explainable Artificial Intelligence (XAI) to identify feature contributions and protective factors, demonstrating how predictive modelling can move beyond classification toward interpretable identification of intervention-relevant characteristics. More recent work has similarly combined XGBoost-based feature contribution analysis with XAI to examine how predictors influence stress across working conditions.

Studies employing DT have also advanced stress-resilience conceptual frameworks with some discovered commonalities, exhibiting similar patterns across ecosystems. Panfilova et al. (2026) evaluated whether five advanced Large Language Models could simulate individuals' responses to psychological questionnaires using semi-structured interview transcripts from 645 Russian-speaking participants, assessing accuracy across 38 psychological scales with correlation, error, bias, agreement, and calibration metrics. It found substantial model- and construct-level variation, with GPT-5 achieving the best overall accuracy, but concluded that LLM simulations remain probabilistic approximations rather than reliable DTs, particularly for less verbally transparent psychological constructs. Vallu et al. (2025) developed a framework for personalized real-time stress detection and management, employing physiological, behavioural, social, and IoT-derived data. Study implemented Principal Component Analysis for feature reduction and gradient boosting for stress prediction. Continuous decision-making was empowered by adaptive reinforcement learning, while Bayesian optimisation was used to tune the models for computational efficiency. Tan and Wei (2025) investigated a stress visualisation and surveillance system running on the DT methodological framework. The system integrates wearable biometric data, office DT modelling, and AI-driven emotion recognition to enable real-time stress monitoring, visualisation through stress heatmaps, tiered alerts, and intervention support. The design compared users with a control group over 12 weeks, thereby examining how a DT could move mental-health management from conventional periodic assessment toward continuous sensing, timely detection, and data-informed intervention within a smart-city environment. The findings reinforce the broader pattern in emerging DT-based mental-health research: continuous, real-time sensing coupled with personalised feedback and intervention to improve both early risk identification and responsiveness. Similar work has shown the value of linking

physiological/psychological monitoring to timely, adaptive interventions, while DT frameworks increasingly emphasise continuous state monitoring and personalised decision support. Kumi et al (2025) presented a DT framework for modelling stress management by integrating Large Language Models. The model availed XAI capabilities in natural language explanations for non-expert healthcare stakeholders. AR and Mathumathi (2025) likewise implemented a DT based smart safety system for occupational stress management. An integration of IoT devices, Machine Learning, and DT technology was deployed, driven by advanced sensors for heart rate, temperature, and GSR readings. Neural network models detected stress patterns in the real time readings more accurately. Chronic stress modelling for real-time intervention was achieved by Ezcurra et al. (2025) using DT methodology. Authors modelled trauma history, records of adverse childhood experiences, chronic health conditions, genetically motivated weaknesses, emotion regulation metrics, gender identity, and environmental stressors into the feature space.

2.4 Research gap and positioning of the present study

Despite this methodological progression, the literature remains fragmented across measurement, prediction and intervention. Psychometric studies primarily establish reliable constructs; conventional statistical studies estimate associations among predictors; and machine-learning studies increasingly improve prediction and explainability. However, these strands are rarely integrated into a single framework that validates latent stress–resilience constructs, discovers developer states, predicts individual state membership, explains feature contributions, and simulates targeted movement between states. This gap motivates the present DT approach. Rather than treating stress assessment as a static classification problem, the proposed framework integrates psychometric validation, latent construct formation, unsupervised stress–resilience profiling, ensemble XGBoost state prediction, XAI explanation, What-If and counterfactual simulation. The resulting HDT architecture therefore extends existing stress-prediction research from “*Who is stressed?*” and “*Which factors predict stress?*” toward the more actionable question of “*Which modifiable factors could potentially move a particular developer toward a more resilient state?*”, through feature validation approach. This positioning is consistent with the recent call for quantitative, multidimensional and actionable approaches to software-engineer well-being (Arriel et al. (2026); Godliauskas and Šmite (2025); Montes (2026)), particularly those that account for gender-specific experiences and challenges faced by female software developers (Oliveira et al. (2025); Guenes et al. (2025); Singh et al. (2024)). Across reviewed DT studies however, relatively limited attention is given to the systematic validation of the feature space itself before it becomes the basis for an HDT: Panfilova et al. exposed construct-dependent limitations in simulated psychological responses; Vallu et al. applied PCA to reduce feature redundancy; Tan and Wei, Kumi et al., AR and Mathumathi, and Ezcurra et al. demonstrated increasingly sophisticated use of physiological, behavioural, psychological, environmental and IoT data, but the emphasis remains predominantly on prediction, sensing, adaptation and intervention rather than establishing whether the underlying variables constitute a psychometrically sound and contextually coherent representation of the human state being twinned. This is a critical methodological gap because an HDT is fundamentally data-oriented, hence unreliable, redundant, poorly structured or insufficiently validated features can propagate measurement error into clustering and prediction (Chen et al., 2025), distort XAI-based driver attribution (Halimeh et al., 2023), weaken contextual interpretation (Hossain et al., 2025), and ultimately produce interventions that are computationally plausible but poorly grounded in the actual human construct (Pawlicki, 2023). Thus, whether data are collected once through surveys or continuously through real-time sensors, rigorous data/feature

validation should precede HDT construction and intelligence. Feature engineering techniques, such as Principal Component Analysis, multicollinearity detection or model-performance evaluation, should rather complement feature validation (Srisuk et al., 2026), not completely replacing it. The present study addresses these gaps by introducing psychometric validation of the HDT feature space through EFA, reliability analysis, CFA, and construct-validity assessment before state discovery, prediction, explainability and counterfactual intervention simulation, thereby strengthening the reliability, interpretability and contextual validity of the resulting HDT. For the rest of the study, materials and methods are presented in section 3 while experimental results are discussed in chapter 4 before concluding the study in chapter 5.

3. Materials and Methods

The eight-stage framework of the personalized DT simulation is presented on Figure 1, operationalizing a closed-loop for developer stress–resilience assessment. It progresses from physical-entity data acquisition to validated human digital representation, stress-resilience state intelligence, what-if simulation, personalized counterfactual generation and simulation, and continuous twin updating.

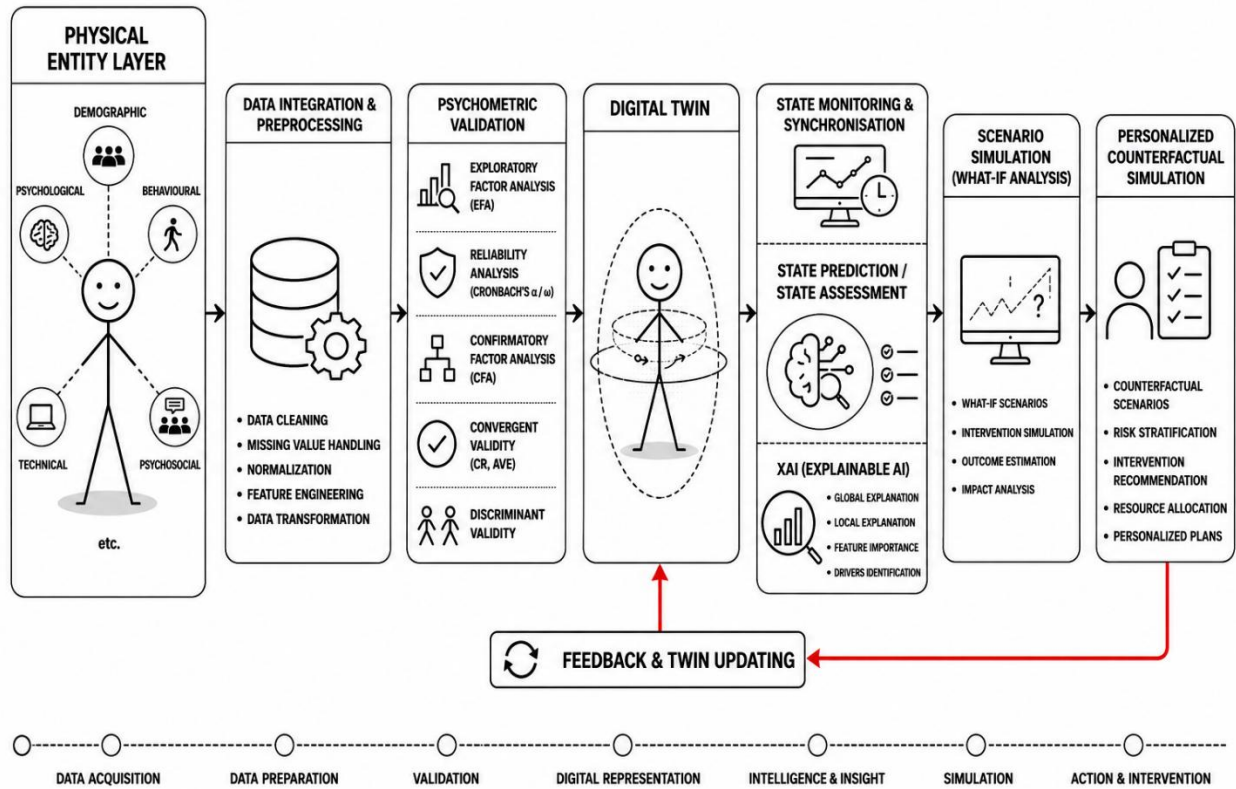


Fig. 1 Proposed Stress-Resilience DT Framework for Personalized Counterfactual Simulation and Adaptive Intervention. A closed-loop feedback mechanism continuously updates the DT engine to support adaptive, evidence-driven intervention.

3.1 Physical data layer, preprocessing and integration

The Physical Entity layer captured the multidimensional representation of the developer from demographic, psychological, behavioural, technical, and psychosocial characteristics, through digital questionnaires. The survey was conducted between September and November 2025 as an observation representing the developer's self-impressions, which provided the empirical basis for constructing their DT. Details of the surveyed features are explained on Table 1.

Table 1 Overview of the surveyed feature space

S/N	Surveyed feature	Variable name	Data Typet
1	Age	Age	Continuous
2	Years of Experience	Tech_Exp	
3	Education	Edu_Level	
4	Role in Tech	Tech_Role	Categorical
5	Social Media Choice	Social_Use	
6	Preferrable Device	Device	
7	Explore New Technologies & Experiment with Unconventional Solutions	Tech_Innov	5-Point Likert Scale
8	Seek Diverse Perspectives on Tech Trends	Persp_Diversity	
9	Prefer Structured Workflows & Attention to Detail	Work_Discipline	
10	Set Goals & Track Progress	Goal_Tracking	
11	Energy from Collaboration & Sharing Ideas	Collab_Energy	
12	Initiate Discussions/Brainstorming	Team_Initiative	
13	Consider Impact on Users & Teammates	User_Empathy	
14	Stress/Anxiety with Complex Technical Challenges	Tech_Stress	
15	Willingness to Compromise for Team/Users	Flexibility	
16	Worry About Mistakes Affecting Performance/UX	Error_Worry	

The data integration and preprocessing phase of the framework convert the heterogeneous raw observations into a clean, consistent and model-ready representation. The individual records in Excel spreadsheet were integrated and subjected to data cleaning, missing-value treatment, normalization, feature engineering and transformation. Categorical variables are appropriately encoded, while numerical variables are normalized where required. Categorical missing values were treated by median analysis, mean for numeric/continuous values. The output is a unified feature matrix in which each row represents a female developer and each column represents a measurable attribute of the developer's state. Feature engineering and data transformation converted the raw survey responses into analytically consistent representations suitable for psychometric modelling. The process included variable recoding, categorical encoding, numerical transformation, standardization, and construction of theoretically informed indicator sets representing stress, resilience, adaptability, conscientiousness, collaboration, and human-centred orientation. These transformations ensured that heterogeneous survey variables were placed on compatible analytical scales while preserving their substantive meaning for subsequent stages of the conceptual framework.

3.2 Psychometric validation of constructed indicators

This phase is intended to establish that the psychological and behavioural measures used by the proposed twin are reliable and construct-valid. EFA identifies the underlying latent constructs:

$$x = \Lambda f + \epsilon \quad (1)$$

where x is the vector of observed questionnaire items, Λ is the factor-loading matrix, f represents latent psychological constructs, and ϵ is the measurement error. Communality quantifies the proportion of each item's variance explained by the retained factors:

$$h_i^2 = \sum_{j=1}^m \lambda_{ij}^2 \quad (2)$$

where h_i^2 depicts communality of item i , λ_{ij}^2 is its loading on factor j and m is the number of retained factors. Cronbach's alpha (α) and McDonald's omega (ω) provides a factor-loading-based reliability estimate. Cronbach's alpha:

$$\alpha = \frac{k}{k-1} \left(1 - \frac{\sum_{i=1}^k \sigma_i^2}{\sigma_T^2} \right) \quad (3)$$

where k is the number of items, σ_i^2 is the variance of item i , and σ_T^2 is the variance of the total scale. And for omega,

$$\omega = \frac{(\sum_{i=1}^k \lambda_i)^2}{(\sum_{i=1}^k \lambda_i)^2 + \sum_{i=1}^k \theta_i} \quad (4)$$

where λ_i is the standardized factor loading of item i , θ_i is its error variance, and k is the number of items. Kaiser-Meyer-Olkin (KMO) assesses whether the data have sufficient shared variance to justify factor analysis as higher values indicate greater sampling adequacy. For completeness, KMO is formulated as

$$KMO = \frac{\sum_{i \neq j} r_{ij}^2}{\sum_{i \neq j} r_{ij}^2 + \sum_{i \neq j} p_{ij}^2} \quad (5)$$

where r_{ij} is the observed correlation between items i and j , and p_{ij} is their partial correlation after controlling for the remaining items. Bartlett's Test evaluates $H_0: R = I$ (no correlations among variables), which is a significant p -value indicating sufficient intercorrelation to support factor analysis.

CFA then subsequently tests the hypothesised measurement structure by

$$x = \Lambda f + \delta \quad (6)$$

and evaluates model adequacy using established fit indices such as χ^2/df , CFI, TLI, RMSEA and SRMR, where Λ represents CFA factor loadings and δ represents measurement errors. Thus, Composite Reliability (CR) is calculated as:

$$CR = \frac{(\sum \lambda_i)^2}{(\sum \lambda_i + \sum \theta_i)} \quad (7)$$

where λ_i is the standardized loading of item i and θ_i is its error variance. Average Variance Extracted (AVE) is also formulated

$$AVE = \frac{\sum \lambda_i^2}{\sum \lambda_i^2 + \sum \theta_i} \quad (8)$$

where AVE represents the proportion of indicator variance captured by the latent construct. Discriminant validity is subsequently established by demonstrating that each construct shares more variance with its own indicators than with other constructs, commonly assessed using the Fornell–Larcker criterion or HTMT ratio.

3.3 Human DT construction

The HDT achieves creates a computational illustration of each developer that mirrors their measurable psychological, behavioural, technical and contextual state. For a developer i , its twin is represented by a state vector:

$$z_i(t) = [z_{i1}(t), z_{i2}(t), \dots, z_{ip}(t)] \quad (9)$$

where $z_i(t)$ is the digital-twin state of developer i at time t , p is the number of validated attributes, and $z_{ij}(t)$ is the current value of attribute j . The twin therefore does not merely store a developer's profile; it maintains a state representation that can be evaluated, modified and simulated under alternative conditions. State monitoring and Synchronisation follows in different stages. State Prediction and Assessment identifies the developer's current stress–resilience state and estimate the probability of belonging to each state.

For a binary state model:

$$P(Y_i = 1|z_i) = \sigma(f(z_i)) \quad (10)$$

where Y_i is the stress–resilience state of developer i , $f(\cdot)$ is the trained predictive model, and $\sigma(\cdot)$ converts the model output into a probability. Predicted state will therefore be:

$$Y_i = \begin{cases} 1, & P(Y_i = 1|z_i) \geq \tau \\ 0, & P(Y_i = 1|z_i) < \tau \end{cases} \quad (11)$$

where τ is the selected classification threshold. At this stage, the following constructs on Table 2 are developed respectively. Therefore, standardized factor scores are computed for each developer after validating the measurement model:

$$S_{ic} = \sum_{j=1}^{k_c} w_{jc} x_{ij} \quad (12)$$

where S_{ic} is developer i 's score on construct c , x_{ij} is the standardized score of item j , w_{jc} is its factor-score/loading weight, and k_c is the number of validated items defining construct c .

Table 2 Mapping of observed survey indicators to latent constructs and their subsequent use in psychometric validation and construct-score computation.

S/N	Constructs	Observed/Surveyed	EFA/CFA Analysis	Score Computation
1	Stress construct	Stress/Anxiety Worry About Mistakes	Establish if indicators reliably represent latent Stress construct.	$Stress_i = w_1(Stress/Anxiety) + w_2(Error\ Worry)$

2	Resilience construct	Goal tracking, Technology openness, Diverse perspectives & Willingness to Compromise	Establish the latent <i>Resilience</i> construct, verify measurement structure.	$Resilience_i = w_1(Goal\ Tracking) + w_2(Technology\ Openness) + w_3(Diverse\ Perspectives) + w_4(Willingness\ to\ Compromise)$
3	Supporting constructs	Conscientiousness, Adaptability, Collaboration and Innovation	Captures additional behavioural characteristics	$Supporting_{ic} = \sum_j w_{jc} x_{ij}$

State monitoring and synchronization employs unsupervised machine learning by K-Means to first provide the empirical grouping of developers into homogeneous stress–resilience profiles:

$$C_k = \arg \min_c \sum_{i=1}^n ||z_i - \mu_k||^2 \quad (13)$$

where C_k is cluster k , z_i is the feature vector of developer i , and μ_k is the centroid of cluster k . The resulting clusters are interpreted and labelled as the corresponding stress–resilience states. Then, the predictive model $f(\cdot)$, XGBoost, is trained with the labelled validated feature space using the state labels to learn the nonlinear relationship between developer characteristics and state membership:

$$p'_i = P(Y_i = 1|z_i) = \sigma(f_{XGB}(z_i)) \quad (14)$$

where Y_i is the cluster-derived stress–resilience state, f_{XGB} is the trained XGBoost model, and $\sigma(\cdot)$ converts the model score into a probability. The classification threshold τ then determines the predicted state:

$$Y'_i = \begin{cases} 1, & p'_i \geq \tau \\ 0, & p'_i < \tau \end{cases} \quad (15)$$

Thus, evaluated K-Means establishes the states whereas XGBoost learns to predict those states for individual developers. The predicted probabilities subsequently support XAI explanation and individual-based counterfactual simulation. The XAI component identifies *why* the twin predicts a particular state. For prediction $f(z)$, SHapley Additive exPlanations (SHAP) decompose the prediction as:

$$f(z) = \phi_0 + \sum_{j=1}^P \phi_j \quad (16)$$

where ϕ_j is the baseline prediction and ϕ_j is the contribution of feature j to the individual prediction. Thus, the system identifies the features that most strongly increase or decrease the probability of the predicted state, providing the evidence required for subsequent simulation.

The next stage of the Scenario Simulation (What-if Analysis) seeks to determine outcome of changes in controllable characteristics of a developer, before any real intervention is implemented. The system asks: “If this particular factor changes while keeping the other relevant characteristics unchanged, what does the model predict will happen?”. Candidate variables are selected using three criteria:

- i. the variable can realistically be influenced through an intervention; Actionability

- ii. the variable has meaningful influence on the state prediction, identified through SHAP/model importance; Predictive relevance
- iii. proposed change falls within empirically observed or intervention-feasible ranges; Plausibility

Let the developer's current state be:

$$z_i = [z_{i1}, z_{i2}, \dots, z_{ip}] \quad (17)$$

For an actionable variable j , a *what-if* scenario modifies its value:

$$z'_{ik} = z_{ik} + \Delta_j \quad (18)$$

while non-manipulated variables remain unchanged:

$$z'_{ik} = z_{ik}, k \neq j \quad (19)$$

where Δ_j is the proposed change to feature j and it is not arbitrarily invented, its derivable from empirically observed differences between favourable and unfavourable states or through predefined feasible intervention increments. For an identified actionable influential factor, the simulation evaluates a small, moderate and substantial improvement rather than simply changing the value randomly.

Thus:

$$\Delta_j \in D_j \quad (20)$$

where D_j is the empirically or domain-defined set of feasible changes for feature j . The modified state is passed through the trained predictive model:

$$P'_{i,j} = P(Y_i = 1 | z'_{i,j}) \quad (21)$$

and compared with the original prediction:

$$\Delta \hat{P}'_{i,j} = P'_{i,j} - P_i \quad (22)$$

where P_i is the original predicted probability and $\hat{P}'_{i,j}$ is the probability after the hypothetical change. This allows the system to report, for example, that if a selected feature increases by a feasible intervention amount, the predicted probability of the favourable stress–resilience state increases from P_1 to P_2 . Multiple variables are subjected to simultaneous:

$$z'_i = z_i + \Delta \quad (23)$$

allowing combinatorial interventions test.

Consequent upon the scenario simulation, smallest improvement to a developer's current state can be identified which can move a particular individual toward a desired stress–resilience state. The individual-based Counterfactual Simulation takes cue from the ordinary *what-if* analysis because the system does not test arbitrary scenarios, it searches for an individualized, feasible alternative state. for a developer i , the counterfactual engine receives the current DT state z_i , current predicted state $\hat{Y}_i$, desired target state Y_i^* , trained XGBoost prediction model $f(\cdot)$, influential/actionable

variables identified through XAI and feasible ranges or intervention constraints for those variables. The counterfactual state z_i^{cf} is formulated as:

$$z_i^{cf} = \arg \min_{z^i} D(z^i, z_i) \quad (24)$$

subject to:

$$f(z^i) = Y_i^* \quad (25)$$

and

$$L_j \leq z_j' \leq U_j, j \in A \quad (26)$$

where L_j, U_j define feasible intervention limits and A denote the set of actionable variables identified from domain knowledge and model explanations. In probability form, the target can instead be expressed as:

$$P(Y_i^* | z_i^{cf}) \geq \tau^* \quad (27)$$

where τ^* is the desired confidence level for the target state. The counterfactual engine effectively seeks to ask about the minimum feasible change in a particular developer's actionable characteristics that would make the model predict a desired state. The exact magnitudes are determined by the optimization process subject to the empirical/domain constraints, not manually selected after seeing the desired outcome.

The Feedback and Twin Updating stage ensure that the DT remains harmonized with the developer's new state after intervention and subsequent observation. After an intervention, newly observed data z_i^{new} are incorporated into the twin:

$$z_i(t+1) = (1-\lambda)z_i(t) + \lambda z_i^{new} \quad (28)$$

where $\lambda \in [0,1]$ controls the influence of newly observed information. The updated state is then re-evaluated:

$$Y'_i(t+1) = f(z_i(t+1)) \quad (29)$$

creating a closed feedback loop in which observed outcomes are used to update the twin, reassess the developer's state, and generate new simulations or counterfactual recommendations when necessary.

4. Results and Discussions

The experimental implementation was conducted in Python using the Google Colaboratory (Colab) cloud computing environment. Colab provided a browser-based Jupyter Notebook platform for the data preprocessing, psychometric analysis, clustering, machine-learning model development, explainable AI, and What-If/counterfactual simulation. Python libraries including Pandas, NumPy, Scikit-learn, XGBoost and SHAP were employed to implement the experimental pipeline, while Colab's computational environment facilitated reproducible execution and evaluation of the proposed framework.

4.1 Identification of latent behavioural constructs

The suitability of the observed behavioural indicators for EFA was assessed with a 0.9418 KMO, indicating excellent sampling adequacy, while Bartlett's test of sphericity was statistically significant, confirming that the correlation matrix contains sufficient shared variance for factor extraction. These results established strong empirical support for proceeding with factor analysis. The eigenvalue structure initially suggested substantial dimensional reduction, with the first factor accounting for an eigenvalue of 5.7720. However, parallel analysis indicated a two-factor solution in the extracted loading structure, with the retained factors explaining 52.16% of total variance (Factor 1 accounted for 40.10% and Factor 2 contributed a further 12.06%). Observed behavioural indicators therefore contained a dominant common behavioural dimension alongside a distinct secondary dimension. The EFA measurement path diagram is presented on Figure 2 with the summary measurement model on Table 3. Factor 1 demonstrated strong loadings for Team Initiative, Perspective Diversity, User Empathy, Collaboration Energy, Goal Tracking, Technology Innovation, and Work Discipline. These loadings suggest a broad behavioural orientation encompassing proactive collaboration, adaptability, structured work behaviour and human-centred engagement. Technical Stress however showed a comparatively weaker association with this dominant dimension, exhibiting a cross loading. Factor 2 was primarily demarcated by the Error Worry and Flexibility pair, while their loadings on Factor 1 were negligible (Figure 3). This separation provides empirical evidence that error-related concern and flexibility exhibit a distinct behavioural dimension in the present sample. Importantly, the very low cross-loadings for most indicators indicate relatively clean factor separation. The communalities on Figure 4 further indicated that the retained factors explained considerable amounts of variance for several indicators, while Technical Stress had the lowest, suggesting that its variance is less adequately represented by the extracted factor structure and should therefore be cautiously considered. The EFA thus provided empirical evidence that the original ten behavioural indicators did not function as ten independent dimensions but rather a dominant shared behavioural structure and a secondary dimension centered mainly on error concern and developer flexibility. Therefore, when the surveyed feature space spoke for itself, the observed items naturally grouped into 2 broad groups as elucidated by the variance explained per loadings on Table 4 and the item-factor mapping on Figure 5. These findings are critical for the proposed DT since it supports data-driven dimensionality reduction before constructing the latent developer profile.

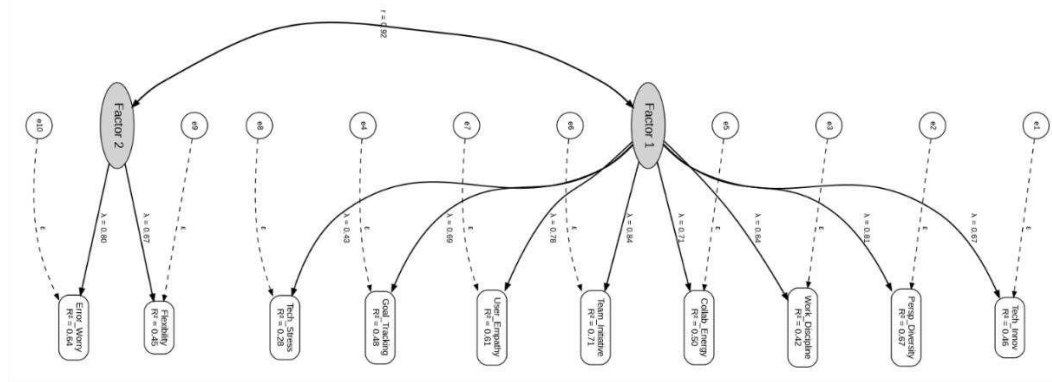


Fig. 2 EFA measurement path diagram showing the two latent constructs, their observed indicators, standardized factor loadings, measurement-error terms, and indicator communality (R^2) values.

Table 3 Indicator-level measurement model assessment showing factor assignment, selected standardized loadings, communalities, and residual variances for the observed developer stress–resilience indicators.

Variable	Assigned_Factor	Selected_Loading	Communality	Residual_Variance
Team_Initiative	Factor_1	0.836688	0.710327	0.289673
Persp_Diversity	Factor_1	0.814875	0.665432	0.334568
User_Empathy	Factor_1	0.783452	0.613896	0.386104
Collab_Energy	Factor_1	0.705819	0.500344	0.499656
Goal_Tracking	Factor_1	0.686555	0.481886	0.518114
Tech_Innov	Factor_1	0.673736	0.45546	0.54454
Work_Discipline	Factor_1	0.644608	0.420224	0.579776
Tech_Stress	Factor_1	0.434478	0.276014	0.723986
Error_Worry	Factor_2	0.801008	0.641624	0.358376
Flexibility	Factor_2	0.668382	0.451103	0.548897

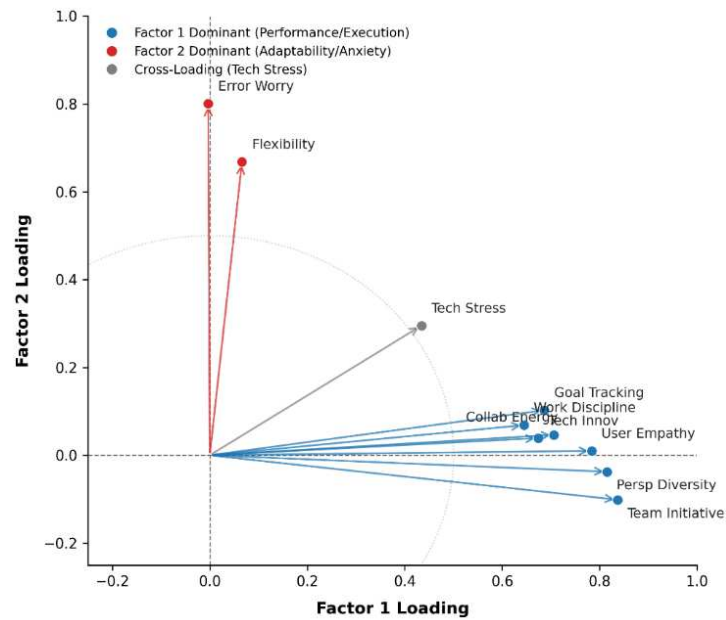


Fig. 3 Visualization of the two dominant factors and the cross-loading incident

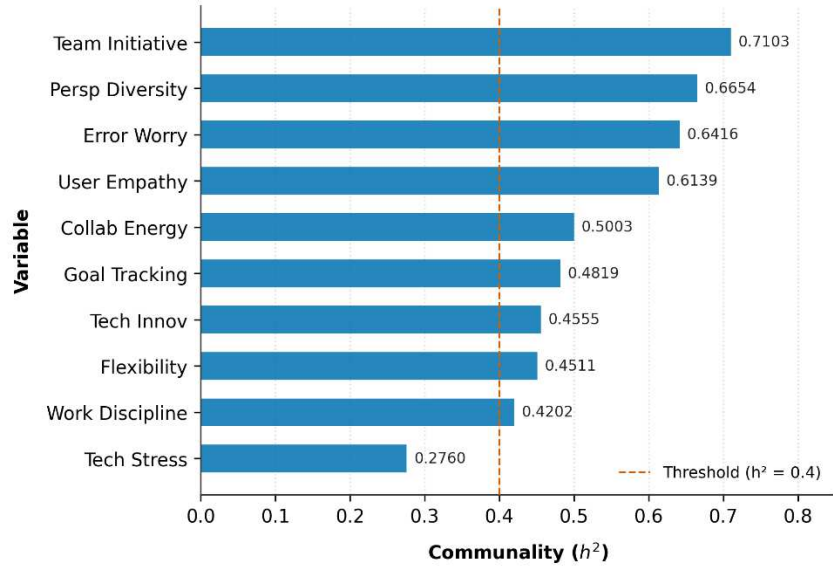


Fig. 4 Degree of communality by retained factors, explaining considerable amounts of variance for several indicators

Table 4 Variance explanation per factor loadings

	Factor	SS	Loadings	Proportion Variance	Cumulative Variance
0	Factor_1	4.0100		0.4010	0.4010
1	Factor_2	1.2063		0.1206	0.5216

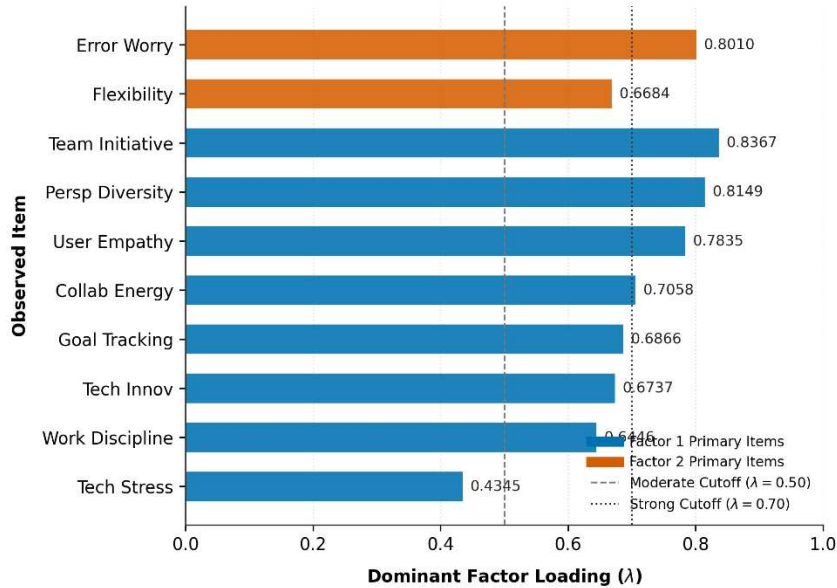


Fig. 5 Observed Item–Factor Mapping Based on Dominant Factor Loadings. The figure presents the empirical mapping of observed measurement items to their dominant latent factors, with each item x_j assigned to the factor F_k for which the absolute standardized loading $|\lambda_{jk}|$ is maximal. The displayed

loading values quantify the strength of the item–factor association and provide an interpretable representation of the measurement structure identified through exploratory factor analysis. This mapping demonstrates the empirical correspondence between the observed stress–resilience indicators and their underlying latent constructs.

4.2 Reliability analysis and confirmatory factor analysis

Whilst the EFA pathway diagram exploratorily showed how indicators empirically group onto factors, the CFA diagram on Figure 6 is confirmatory and tests the hypothesized measurement structure using standardized loadings, residuals, R^2 , and latent-factor covariances. The model demonstrates that the five latent constructs are empirically represented by their observed indicators with standardized loadings ranging from 0.632 to 0.825, with Persp_Diversity ($\lambda=0.825$) and Goal_Tracking ($\lambda=0.804$) providing the strongest measurement contributions. Flexibility ($\lambda=0.632$) is comparatively weaker and the corresponding R^2 values (0.400–0.681) indicate that the constructs explain substantial portions of indicator variance, although residual variance remains, particularly for Flexibility (0.600). The model also shows strong inter-construct associations, with correlations ranging from 0.864 to 1.099, suggesting substantial overlap among dimensions of developers' stress, adaptability, conscientiousness, collaboration, and human-centred orientation. As can be observed on Figure 7, Cronbach's α ranged from 0.635 to 0.767, whereas McDonald's ω ranged from 0.846 to 0.896, indicating generally stronger reliability under the omega estimate. Openness–Adaptability demonstrated the strongest reliability profile, followed by Collaboration and Conscientiousness. Technical Stress produced but a substantially higher, while Human-Centred Orientation showed the lowest alpha but an acceptable omega of 0.846. The discrepancy between alpha and omega is particularly relevant because each construct contains only two indicators. Given the sensitivity of Cronbach's alpha to scale length, the consistently higher omega estimates provided supportive evidence of internal consistency and justify proceeding with latent-variable modelling. This is while acknowledging the limitations of two-item constructs. In the CFA parameter estimates presented on Table 8, the results provided strong empirical support for the proposed construct measurement structure. All freely estimated factor loadings were positive and statistically significant ($p < .001$), with particularly strong relationships observed for Persp_Diversity with Openness–Adaptability (1.117; $z = 19.37$), Goal_Tracking with Conscientiousness (1.102; $z = 18.67$), Team_Initiative with Collaboration (1.004; $z = 19.38$), and Error_Worry with Technical Stress (0.976; $z = 16.80$); Flexibility also demonstrated a significant loading on Human-Centred Orientation (0.910; $z = 16.47$). The latent constructs therefore exhibited substantial and statistically significant interrelationships ($p < .001$), with the strongest association between Human-Centred Orientation and Technical Stress (0.811). Collectively, these results indicated that the observed indicators expressively represent their corresponding latent constructs, while the significant construct-level associations suggested that the dimensions are related components of developers' broader behavioural and stress–resilience profile. The overall CFA fit was strong, as indicated on Figure 8 (CFI=0.983, TLI=0.969, RMSEA=0.060, $\chi^2(25)=79.88$), supporting the adequacy of the measurement structure for representing the underlying dimensions investigated in the developer stress–resilience DT framework.

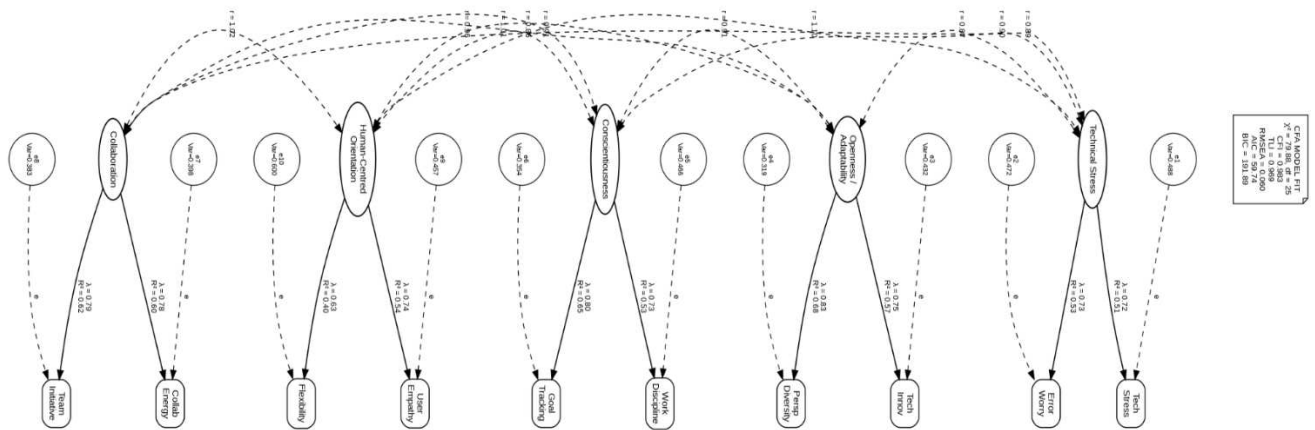


Fig. 6 Confirmatory factor analysis measurement model for the developer stress-resilience constructs indicating the latent constructs as ellipses, observed indicators as rectangles, standardized loadings (λ) on every loading arrow, residual/error terms (e) with derived residual variances, R^2 /communality for each indicator, latent-factor correlations (r) from the supplied covariance estimates, standardized using the latent variances and the CFA fit indices

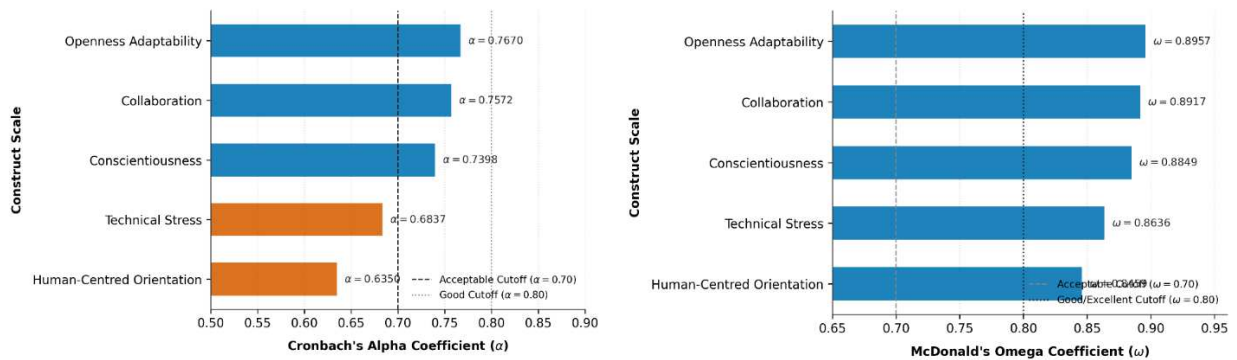


Fig. 7 Stress-resilience state in relation to internal-consistency reliability. The figure examines the relationship between the counterfactual developer state and the internal-consistency estimates obtained from Cronbach's α and McDonald's ω . Whereas α represents the conventional tau-equivalent reliability estimate, ω provides a factor-loading-sensitive estimate of composite reliability, thereby enabling a more robust assessment of measurement consistency.

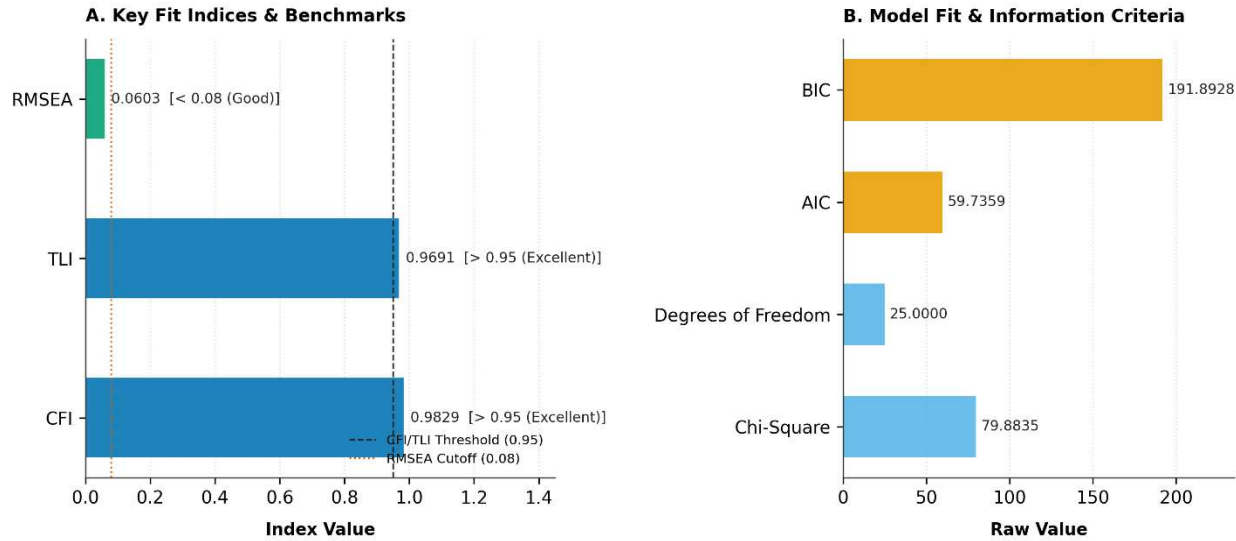


Fig. 8 Comprehensive Confirmatory Factor Model Fit and Information-Criterion Diagnostics. The figure presents two complementary sets of diagnostics for evaluating the adequacy and parsimony of the hypothesized measurement model. The first panel reports the principal goodness-of-fit indices, Root Mean Square Error of Approximation (RMSEA), Tucker–Lewis Index (TLI), and Comparative Fit Index (CFI). This together their established benchmark criteria, assessing approximate residual error and incremental model fit. The second panel presents the chi-square statistic (χ^2), degrees of freedom (df), Akaike Information Criterion (AIC), and Bayesian Information Criterion (BIC), characterizing model–data discrepancy and the balance between statistical fit and model complexity. Together, the diagnostics provide complementary evidence regarding the model's absolute/approximate fit, incremental fit, and parsimony.

4.3 Discussion of factor structure, reliability and construct validity

The psychometric analysis proceeded in two complementary stages. EFA examined the empirical dimensionality of the observed developer stress–resilience indicators without imposing the subsequent theoretical construct structure. As earlier reported in Table 4, the EFA yielded a two-factor solution, with Factor 1 accounting for 40.10% of the variance and Factor 2 accounting for 12.06%. The corresponding reliability and convergent-validity assessment indicated acceptable internal consistency and extracted variance for the two empirically derived factors. This supports the adequacy of the observed indicators for further measurement-model development. The two-factor EFA structure was not, however, treated as the final theoretical construct structure. Rather, it served as an empirical diagnostic for identifying patterns of association among the indicators. Based on the conceptual distinctions established in the study, the indicators were subsequently specified into five theoretically meaningful latent constructs: Technical Stress, Openness–Adaptability, Conscientiousness, Collaboration, and Human-Centred Orientation. This theory-driven specification was evaluated using CFA. As shown earlier on Figure 8, the five-construct measurement model demonstrated strong overall fit, indicating that the hypothesised measurement structure provides a close representation of the observed covariance structure. The construct-level CR and AVE results (Table 5) further provided evidence of acceptable internal consistency and

convergent validity for most of the specified constructs. Thus, the two-factor and the five-construct solution reported in Table 6 represent different analytical purposes rather than competing results. This sequential EFA–CFA strategy therefore provides basis for progressing from observed survey indicators to theoretically interpretable construct scores for subsequent DT state modelling and predictive analysis.

Table 5 Construct reliability and convergent validity assessment based on standardized confirmatory factor loadings, reporting CR and AVE.

Construct / Factor	CR	AVE
Factor 1	0.871	0.467
Factor 2	0.710	0.546

Table 6 Construct reliability and convergent validity of the developer stress–resilience measurement model.

Construct	Standardized loadings	CR	AVE	Convergent validity
Technical Stress	.715, .727	.684	.520	Marginal
Openness & Adaptability	.754, .826	.769	.625	Supported
Conscientiousness	.730, .804	.742	.590	Supported
Collaboration	.776, .786	.757	.609	Supported
Human-Centred Orientation	.737, .632	.639	.471	Marginal

4.4 Derivation of the developer Stress–Resilience states by unsupervised K-Means

Unsupervised learning by K-Means identified the most appropriate number of latent stress–resilience states from the continuous developer feature space. Two-state solution produced the highest silhouette coefficient (0.5529) as against 3(0.4693), 4(0.4771) and 5(0.4333), returning a two-cluster solution as the strongest combination of within-cluster cohesion and between-cluster separation among the evaluated alternatives. Accordingly, $K = 2$ was retained as the optimal clustering configuration, with the resulting cluster centroids presented on Table 7 where the two clusters exhibited a pronounced and directionally consistent separation across all five validated latent constructs. Cluster 0 was characterized by negative standardized centroid values for Technical Stress, Openness–Adaptability, Conscientiousness, Collaboration, and Human-Centred Orientation. In contrast, Cluster 1 exhibited positive centroid values across the same dimensions. Thus, the clustering is not driven by a single dimension but reflects a coherent multivariate separation between developers with comparatively lower levels across the measured stress/resilience-related constructs. The labels 0 and 1 have no intrinsic substantive meaning as they are arbitrary cluster identifiers. Their interpretation as Low-Stress Moderate-Resilience (LSMR) and High Stress-Resilience (HSR), respectively, was based on the empirical centroid profiles. Cluster 0 was designated LSMR because it occupies the lower end of the Technical Stress dimension (−1.0019) and simultaneously exhibits comparatively lower scores on the four resilience-related constructs (Table 8). Conversely, Cluster 1 (HSR) combines a substantially elevated Technical Stress centroid with positive scores across Openness–Adaptability, Conscientiousness, Collaboration, and Human-Centred Orientation. The labels therefore describe

the empirical psychological profiles represented by the clusters, rather than implying that K-Means itself generated the labels.

This interpretation is further supported by the feature-level summary on Table 9. The HSR group comprises 425 (70.25%) developers, whereas LSMR comprises 180 (29.75%), indicating that the high-stress/high-resilience profile is the predominant cluster in the sample. Age distributions are highly similar between the groups (mean = 22.55 for HSR versus 22.64 for LSMR; median = 21 in both), suggesting that age does not substantially distinguish the two states. In contrast, technical experience shows a clearer separation, with HSR developers reporting a higher mean experience (3.31) than LSMR developers (2.38), although both groups have a median of two years. Most importantly, the five latent constructs reproduce the centroid distinction such that HSR developers have positive mean scores for Technical Stress, Openness–Adaptability, Conscientiousness, Collaboration, and Human-Centred Orientation, whereas the corresponding LSMR means are consistently negative. The median, minimum and maximum values additionally demonstrated substantial within-state variability, indicating that the clusters represent multivariate tendencies rather than deterministic thresholds.

Table 7 K-Means Cluster Centroids Across the Validated Stress–Resilience Construct Space. Centroid values represent standardized mean positions of the two clusters across Technical Stress, Openness–Adaptability, Conscientiousness, Collaboration, and Human-Centred Orientation.

Stress_Resilience _Cluster	Technical_ Stress	Openness_Ada ptability	Conscientio usness	Collabor ation	Human_Centred_O rientation
0	-1.0019	-0.9560	-0.9906	-1.0104	-1.0395
1	0.4243	0.4049	0.4196	0.4280	0.4403

Table 8 Empirical interpretation of the two K-Means Stress–Resilience states. The table maps the numerical K-Means cluster identifiers to their substantive state interpretations using the corresponding standardized centroid profiles across Technical Stress and the four resilience-related constructs.

Cluster	State interpretation	Technical_ Stress	Openness_ Adaptability	Conscientiousness	Collaboration	Human_ Centred_ Orientation
0	LSMR	-1.2151	-1.2220	-1.2310	-1.2128	-1.1962
1	HSR	0.5146	0.5176	0.5214	0.5137	0.5066

Table 9 Descriptive characteristics of developers stratified by the derived HSR–LSMR states. The table reports the mean, standard deviation (D), median, minimum (Min) and maximum (Max) for demographic, experiential, and latent stress–resilience features derived state.

Feature	Stress_ Resilience_S tate	Mean	SD	Median	Min	Max
Age	LSMR	22.6389	6.9914	21	18	67
	HSR	22.5459	5.5464	21	18	63

Tech_Exp	LSMR	2.3778	2.2625	2	0	20
	HSR	3.3106	3.0231	2	0	21
Technical_Stress	LSMR	-1.0019	0.625	-0.7718	-2.4818	0.0421
	HSR	0.4243	0.4457	0.339	-0.6023	1.1815
Openness_Adaptability	LSMR	-0.956	0.5704	-0.7857	-2.3305	0.1282
	HSR	0.4049	0.4285	0.322	-0.7035	1.1308
Conscientiousness	LSMR	-0.9906	0.5857	-0.7801	-2.4079	-0.1381
	HSR	0.4196	0.4315	0.3177	-0.5835	1.1671
Collaboration	LSMR	-1.0104	0.6337	-0.785	-2.4858	0.1149
	HSR	0.428	0.4515	0.3341	-0.6114	1.2118
Human_Centred_Orientation	LSMR	-1.0395	0.6578	-0.8374	-2.5885	-0.065
	HSR	0.4403	0.4917	0.3642	-0.8061	1.424

4.5 Ensemble state prediction and assessment by XAI SHAP

Based on the classification of developers into the stress-resilience state, their digital twin are subsequently constructed with a unique identification each, and DT_State_Vector providing a compact numerical representation of the developer's psychological state for subsequent simulations. XGBoost was implemented to drive the intelligent layer of the HDT engine during subsequent simulations. The model was benchmarked against Random Forest, Support Vector Machine, and Logistic Regression. Using 484 training and 121 testing instances, the comparative modelling results indicated strong predictive separability of the empirically derived stress-resilience states. XGBoost and Random Forest achieved the perfect training performance with XGBoost confusion matrix indicating zero false positives and false negatives (85 high-stress resilient and 36 low-stress moderate-resilience cases correctly classified), demonstrating complete class separation on the evaluated test set. Importantly, 5-fold cross-validation yielded slightly more conservative but consistently high performance (Accuracy = 0.9835 ± 0.0074 ; Precision = 0.9820 ± 0.0046 ; Recall = 0.9786 ± 0.0145 ; F1 = 0.9801 ± 0.0091 ; ROC-AUC = 0.9988 ± 0.0009), indicating that the predictive capability was not solely dependent on a single train-test partition. The final XGBoost model further produced highly confident state predictions, with the illustrated HDT instances correctly classified and prediction probabilities approaching 1.0, thereby demonstrating that the model can reliably infer the empirically defined stress-resilience state from the developer's observable and latent psychological profile. This provided a strong predictive layer for subsequent DT simulation and counterfactual analysis. Extracted decision tree on Figure 9 indicate that Collaboration and Openness/Adaptability constitute the dominant decision dimensions, with Collaboration providing the largest initial split gains (273.90), followed by Openness/Adaptability (271.47), while Human-Centred Orientation and Technical Stress provide important secondary refinements. The rules suggest that lower collaboration and lower openness/adaptability levels substantially alter the predicted stress-resilience state. Technical Stress further differentiate developers within low-openness profiles, whereas Human-Centred Orientation provides additional separation among collaboration-defined subgroups. Device appears only as a minor downstream discriminator, indicating that the model's principal predictive structure is psychological rather than demographic or technological.

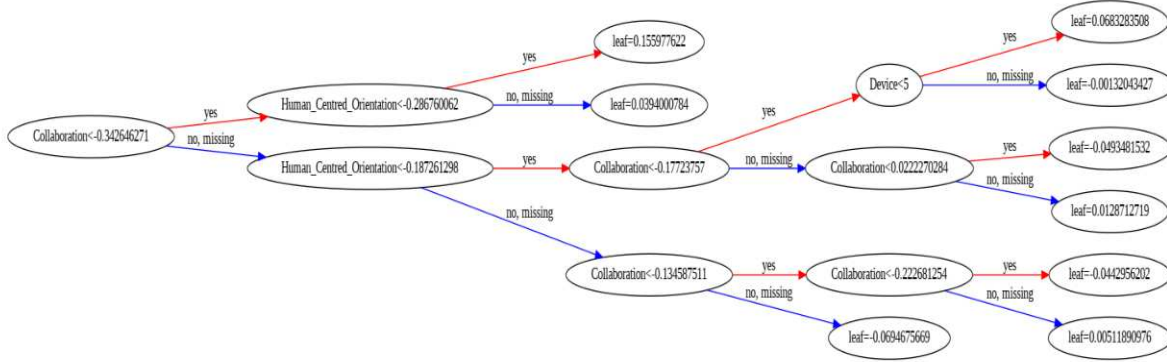


Fig. 9 XGBoost-derived decision tree for HSR-LSMR state discrimination. The figure presents the learned decision structure underlying the XGBoost classifier, illustrating how the predictor space is partitioned to distinguish HSR from LSMR developers.

The XGBoost and SHAP analyses converge on the dominant role of latent psychological characteristics in determining stress-resilience state, although their importance rankings differ because XGBoost gain and SHAP quantify contribution differently. Observed on Figure 10, XGBoost identified Collaboration (0.3052) and Conscientiousness (0.3047) as the strongest contributors, followed by Openness-Adaptability (0.2569), whereas SHAP ranks Conscientiousness (2.6004) highest, followed by Openness-Adaptability (1.5231), Technical Stress (1.3460), Collaboration (1.3248) and Human-Centred Orientation (0.7504). In contrast, demographic and contextual variables, including Age, Education Level, Technical Experience, Technical Role, Device and Social Use, exhibited comparatively negligible influence, with Social Use contributing essentially nothing in both analyses. Findings indicate that the model's discrimination is driven predominantly by psychological and behavioural latent constructs rather than demographic or technology-use characteristics, supporting their centrality as state-defining variables within the DT. SHAP projected the explainability further through the summary plot on Figure 11 showing the drivers of Stress-Resilience dynamics. The direction analysis indicates that the latent psychological constructs predominantly drive predictions toward the HSR state, with Conscientiousness exerting the strongest directional contribution (mean SHAP = -0.727), followed by Technical Stress (-0.629), Openness/Adaptability (-0.485), Collaboration (-0.444), and Human-Centred Orientation (-0.289). In contrast, contextual attributes such as Android-device use, software-development role, and occasional social-media use showed only marginal positive contributions toward the LSMR state.

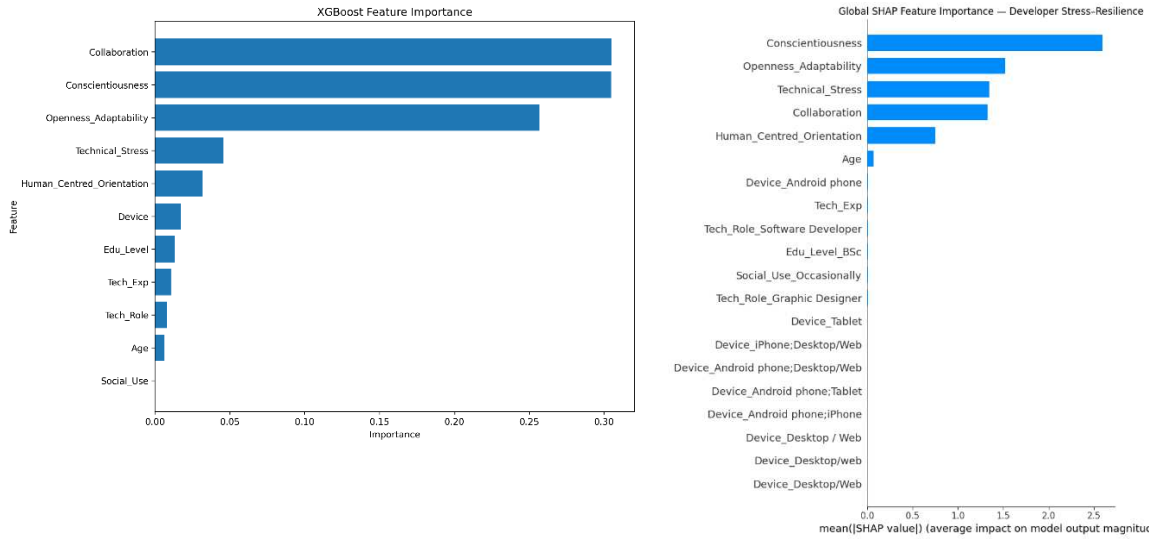


Fig. 10 Concordance Between XGBoost Feature Importance and SHAP-Based Predictor Attribution. The figure juxtaposes the global feature rankings obtained from the XGBoost model and SHAP analysis, enabling comparison of model-derived predictive importance with Shapley-based attribution. Consistent prominence across the two rankings identifies predictors that contribute substantially to discrimination between HSR and LSMR states.

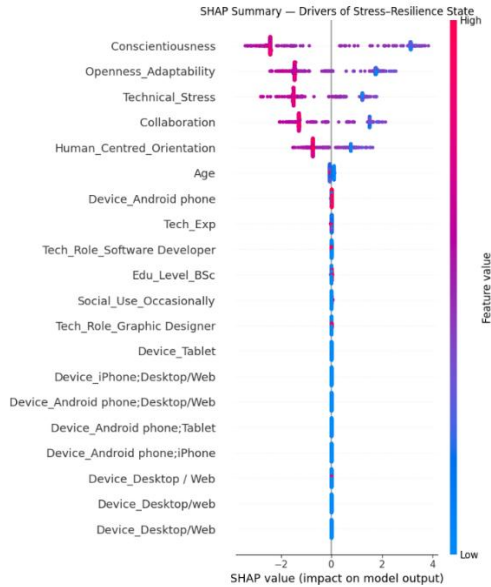


Fig. 11 SHAP Summary of the Principal Drivers of Developer Stress–Resilience State. The SHAP summary plot characterizes the distribution, magnitude, and direction of feature contributions across developers, with feature position reflecting global importance and SHAP-value polarity indicating whether individual observations increase or decrease the model output associated with the predicted stress–resilience state.

4.6 Stress-Resilience DT profiling of surveyed developers

The DT profile of developer DT_DEV_0001 is presented on Table 10 and Figure 12. The developer exhibits HSR state, driven predominantly by the developer's latent psychological profile rather than demographic or technology-related attributes. Her strongest individual contributions arise from Conscientiousness (SHAP = -2.494), Openness/Adaptability (-1.475), Technical Stress (-1.356), and Collaboration (-1.283), with Human-Centred Orientation (-0.752) providing a further substantial contribution. The twin characterises a developer whose observed stress condition is accompanied by substantial psychological and behavioural resources that support resilience. For developer DT_DEV_0026 on Table 11 and Figure 13 respectively, her twin indicates that her predicted LSMR state is driven predominantly by the psychological constructs, with Conscientiousness exerting the strongest contribution (SHAP = -2.753), followed by Openness-Adaptability (-1.439), Collaboration (-1.136), Technical Stress (-0.988), and Human-Centred Orientation (-0.531). Her negative SHapley contributions collectively indicate a strong movement toward the LSMR state, while demographic and contextual predictors contribute negligibly. The two instance-level explanations reveal a remarkably consistent construct-level hierarchy despite differences in their feature values. For both developers, Conscientiousness is the dominant contributor, followed by Openness-Adaptability, Collaboration, Technical Stress, and Human-Centred Orientation, whereas demographic and categorical variables contribute negligibly. For DT_DEV_0001, these five constructs accounted for the overwhelming majority of the absolute SHAP attribution, with Conscientiousness producing the largest contribution ($|\text{SHAP}| = 2.4943$), followed by Openness-Adaptability, Technical Stress, Collaboration, and Human-Centred Orientation. DT_DEV_0026 exhibits the same ordering except for the relative positions of Technical Stress and Collaboration, with Conscientiousness again dominant ($|\text{SHAP}| = 2.7528$), followed by Openness-Adaptability, Collaboration, Technical Stress, and Human-Centred Orientation. Importantly, the feature values differ substantially between the two developers, particularly for the five latent constructs, demonstrating that the SHAP explanation is responding to individual-level positioning within the learned feature space rather than merely reproducing a fixed ranking. The near-zero contributions of Age, Tech_Exp, device, social-use and educational/role indicators further suggest that, for these two instances, the classification is principally determined by the validated latent stress-resilience constructs.

Table 10 Instance-level SHAP attribution for DT_DEV_0001HSR developer

Feature	SHAP_Value	Feature_Value	Absolute_SHAP
Conscientiousness	-2.4943	0.40797	2.49429
Openness_Adaptability	-1.4747	0.17955	1.47471
Technical_Stress	-1.3558	0.01716	1.35582
Collaboration	-1.2834	0.40098	1.28337
Human_Centred_Orientation	-0.752	0.6583	0.75195
Age	-0.0691	0.57089	0.06909
Tech_Exp	-0.0106	1.04239	0.01059
Device_Android_phone	-0.0055	0	0.00549
Tech_Role_Software_Developer	-0.0054	1	0.00543
Edu_Level_BSc	0.0021	1	0.0021

Tech_Role_Graphic Designer	-0.0009	0	0.00089
Social_Use_Occasionally	-0.0008	0	0.00077
Edu_Level_100 level	0	0	0
Edu_Level_Bachelor in view	0	0	0
Edu_Level_HS	0	0	0

Table 11 Instance-level SHAP attribution for DT_DEV_0026 LSMR developer

Feature	SHAP_Value	Feature_Value	Absolute_SHAP
Conscientiousness	-2.75277	-0.54513	2.752773
Openness_Adaptability	-1.43881	-0.59245	1.438809
Collaboration	-1.13572	-0.73972	1.135722
Technical_Stress	-0.98758	-0.645	0.987584
Human_Centred_Orientation	-0.53113	-0.96401	0.53113
Tech_Role_Data Scientist	-0.002	0	0.001996
Age	-0.00069	18	0.00069
Device Iphone	0.000634	0	0.000634
Social_Use_Occasionally	0.000245	0	0.000245
Tech_Exp	-0.00012	6	0.000121
Edu_Level_Business	0	0	0

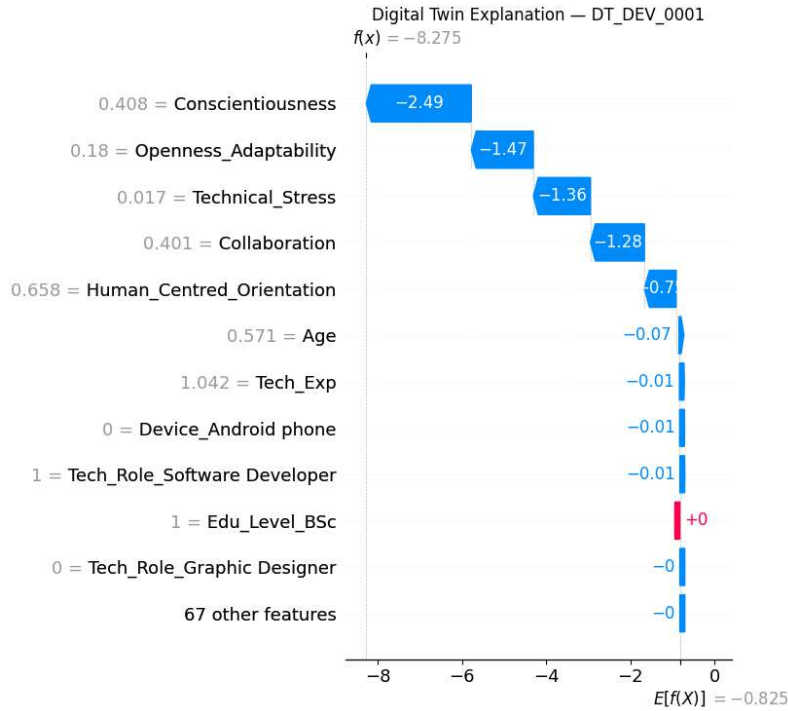


Fig. 12 Instance-level SHAP eXplanation for Developer DT_DEV_0001. The waterfall plot decomposes the XGBoost prediction for DT_DEV_0001 from the expected model output to the

final HSR prediction, showing the sequential additive contributions of individual predictors and identifying the features that most strongly shift the prediction from the baseline.

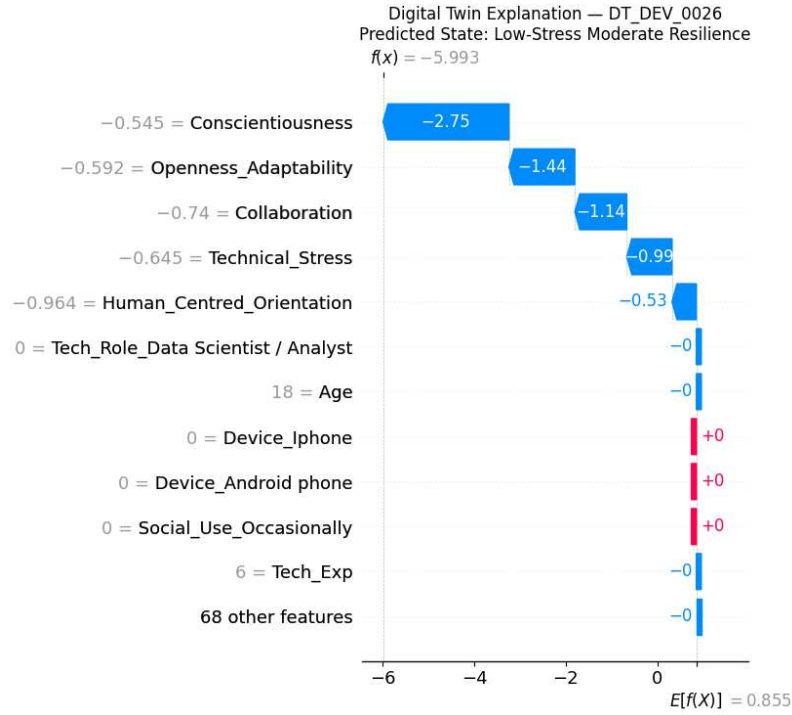


Fig. 13 Instance-level SHAP explanation for developer DT_DEV_0026.

The waterfall plot decomposes the XGBoost prediction for DT_DEV_0026 from the expected model output to the final LSMR prediction, quantifying the individual feature contributions that collectively determine the instance-level classification.

4.6 Initial HDT scenario simulation outcome

The What-If simulations on the existing HDT were evaluated using the 20 best-ranked scenarios from each outcome category. The results demonstrated how targeted, construct-level interventions altered XGBoost predicted stress–resilience states and provided a basis for personalised intervention prioritisation. Across the two outcome categories, the simulations revealed a marked asymmetry in intervention response. Among the 20 highest-ranked LSMR scenarios, all cases achieved a simulated transition to HSR, with Openness–Adaptability constituting the most frequent primary target (9/20), followed by Human-Centred Orientation (6/20), Collaboration (3/20), and Conscientiousness (2/20) [Table 12]. Conversely, the 20 HSR scenarios remained in the HSR state, with Openness–Adaptability again representing the most frequent primary target and Human-Centred Orientation, Collaboration, and Conscientiousness forming the principal complementary targets [Table 13]. Collectively, these results suggested the ability of the HDT in distinguishing between state-preserving interventions for HSR developers and state-transition initiatives for LSMR clusters, while consistently identifying Openness–Adaptability as a prominent intervention lever. However, these findings represent simulated responses rather than

observed intervention effects. DT_DEV_0280 is particularly useful as a case study because it was identified as the strongest transition scenario in the LSMR simulations, with the predicted probability changing from 0.999356 to 0.483763 and the resulting state changing to HSR (Table 14). The personalised recommendation therefore links the modelled transition to actionable construct-level interventions rather than merely reporting a probability change. Key finding therefore is that Openness_Adaptability, Collaboration, Conscientiousness, and Human_Centred_Orientation are the most important commonality since they recur as intervention targets in both states, closely aligning with the XGBoost/SHAP findings. However, their functional role differs by current state; for HSR developers they predominantly reinforce an already stable classification, whereas for LSMR developers their personalised combinations can produce substantial simulated movement toward HSR. This provides evidence that the DT is not merely recommending generic interventions, rather, it exhibits state-contingent intervention behaviour, where the same underlying factors can serve different purposes, i.e, conservation of HSR resilience versus transition toward higher resilience for LSMR developers. Observed comparative patterns are therefore synthesized on Table 15.

Table 12 Aggregated What-If Digital Twin simulation findings for Low-Stress Moderate Resilience (LSMR) developers across the 20 highest-ranked intervention scenarios

Finding	Outcome
Developers captured	20
Initial state	20/20 LSMR
State transitions	20/20 (100%) transitioned to HSR
Most frequent primary target	Openness_Adaptability — 9/20 (45%)
Other primary targets	Human_Centred_Orientation — 6/20 (30%); Collaboration — 3/20 (15%); Conscientiousness — 2/20 (10%)
Major secondary targets	Conscientiousness, Collaboration, Human_Centred_Orientation, Openness_Adaptability
Strongest transition	DT_DEV_0280 — probability changed from 0.999356 to 0.483763
Other notable transitions	DT_DEV_0416, DT_DEV_0276, DT_DEV_0444 and DT_DEV_0247 also showed substantial probability shifts
Positive state transition	20/20 (100%)
Probability improvement toward target state	All 20 cases showed a reduction in Class-0 probability, corresponding to movement toward High-Stress Resilient
Critical priority	3/20 — DT_DEV_0280, DT_DEV_0444, DT_DEV_0432
High priority	17/20
Overall intervention outcome	100% simulated transition success

Table 13 Aggregated What-If simulation findings for HSR developers across the 20 highest-ranked intervention scenarios

Finding	Outcome
Developers captured	20
Current state	20 HSR
Most frequent primary target	Openness_Adaptability
Other major targets	Human_Centred_Orientation, Collaboration, Conscientiousness

Secondary targets	Conscientiousness, Human_Centred_Orientation, Collaboration, Openness_Adaptability
State transitions	0/20
Positive probability improvement	Mainly DT_DEV_0217, DT_DEV_0374 and DT_DEV_0591
Negative probability change	DT_DEV_0522, DT_DEV_0521, DT_DEV_0516, DT_DEV_0527
Recommended priority	20/20 Moderate

Table 14 Personalised What-If intervention recommendation for DT_DEV_0280. Result shows the developer's baseline state and probability, primary and secondary intervention targets and actions, simulated post-intervention probability and state, probability change, transition outcome, and derived intervention priority.

Parameter	Value
Current_State	LSMR
Primary_Intervention_Target	Collaboration
Primary_Intervention	Team collaboration/peer-support
Primary_Action	Increase pair programming, peer review, mentor...
Secondary_Intervention_Target	Conscientiousness
Secondary_Intervention	Structured workflow and goal-management support
Secondary_Action	Introduce clearer milestones, coding checklist..
Baseline_Probability	0.999356
Combined_Probability	0.483763
Combined_State	HSR
Combined_State_Changed	True
Intervention_Probability_Improvement	-0.515593
Positive_State_Transition	True
Intervention_Priority_Score	7.0
Recommended_Priority	Critical
Personalised_Recommendation	For DT_DEV_0280, prioritise team collaboration...

Table 15 Comparative patterns and state-specific intervention characteristics across HSR and LSMR developers

Dimension	HSR intervention pattern	LSMR intervention pattern	Interpretation
Initial state	HSR	LSMR	Opposite intervention directions
Intervention objective	Maintain/support existing resilience	Promote transition to HSR	State-dependent logic
State transition	0/20 observed transitions	20/20 observed transitions	Strong asymmetry for topmost observations
Probability behaviour	Remained extremely high (~0.9997–0.9998)	Class-0 probability decreased substantially	LSMR developers showed greater simulated transition
Dominant intervention emphasis	Openness/Adaptability, Human-Centred Orientation, Collaboration, Conscientiousness		Same core psychosocial factors
Openness/Adaptability	Most frequent primary target		Shared dominant factor

Collaboration	Primary or secondary target; sometimes with negligible/negative probability changes	Associated with successful transitions	State-dependent effect
Conscientiousness	Mostly secondary support	Primary/secondary reinforcement	Largely complementary
Human-Centred Orientation	Repeatedly identified as an intervention target		Consistent cross-state intervention dimension
Multi-factor intervention	Retained inertial	Produced 100% transitions	Consequential for LSMR→HSR only
Negative/negligible effects	Mostly zero or negative probability changes	Complete transition cases with varying magnitude	Responsiveness strongly state-dependent
Priority profile	Mostly moderate priority	17 High + 3 Critical among captured cases	Transition-focused cases attract higher intervention urgency
Strongest transition	+0.004586 prob. Change	-0.515593 prob. change	Larger simulated movement in LSMR
Overall implication	Interventions as state-maintenance mechanism		Directional, state-dependent DT intervention

4.7 Counterfactual Simulation for HDT update

The counterfactual simulation was implemented on 19 LSMR developers, taking cue from the earlier scenario simulation phase with a 100% positive transition and 0.5591 and 0.8144 mean and median probability improvements respectively. The influence of the respective feature variable on the recorded improvements is presented on Table 16, providing a strong evidence base for feature-level intervention, showing that not all indicators exerted the same influence on the LSMR state. User_Empathy exhibited strongest hope for stress resilience, increasing mean predicted probability by 0.2027 and producing positive state transitions in 89.47% of cases where it was counterfactually modified. Persp_Diversity follows (+16.91 pp; 31.58% transitions), while Goal_Tracking, Tech_Innov, and Work_Discipline produce moderate probability gains. Interestingly, Flexibility produces a relatively large mean absolute probability change (+7.79 pp) and affects 73.68% of cases, but its positive state-transition rate is only 5.26%, indicating that probability movement does not necessarily translate into state reclassification. Conversely, Error_Worry (-2.63 pp) and Tech_Stress (-0.83 pp) generally move predictions in an unfavourable direction, while Team_Initiative has virtually no predictive effect in this counterfactual setting. Therefore, the results identified User_Empathy, Persp_Diversity, Goal_Tracking, Tech_Innov, and Work_Discipline as the most consequential intervention levers. Feature-change summary shows a complementary but important pattern. Observation indicated that whenever intervention magnitude applied to each feature is considered, User_Empathy generates the largest mean absolute change, followed by Flexibility and Goal_Tracking. However, every feature produced a 100% transition rate after its feature-level intervention among the evaluated LSMR developers. This signals a sufficiently strong intervention design, capable of moving all cases to the target state. This is when each feature was independently subjected to the prescribed counterfactual modification. Importantly, the distinction between probability improvement and actual state transition shows why intervention effectiveness should not be judged from probability changes alone.

The developers' transition dynamics provided strong evidence that the framework can recognize individualised, feature-driven opportunities for favourable transitions, like the case during scenario simulation. Of the 19 developers, all achieved a positive state transition to HSR, although the magnitude of improvement varied substantially. This indicates heterogeneous intervention responsiveness, as can be observed on Table 17. The probability improvement ranged from only 0.0004 (0.04%) for DT_DEV_0529 to 0.9176 (91.76%) for DT_DEV_0133. A particularly important observation is that developers with initially low HSR probabilities experienced the largest gains: DT_DEV_0038 increased by 96.79%, DT_DEV_0155 by 91.57%, and DT_DEV_0133 by 91.77%. Conversely, developers already highly probable to occupy the HSR state showed only marginal gains, e.g., DT_DEV_0529 (0.04%) and DT_DEV_0259 (1.61%). This pattern suggests a ceiling effect and, more importantly, supports the value of the HDT as an adaptive system, implying that intervention intensity can be directed toward developers for whom counterfactual modification is predicted to yield the greatest state improvement rather than applying uniform interventions. The model-derived counterfactual influence returned features strongly associated with improved predicted states under the auspices of the ensemble XGBoost model. Dumbbell plot on Figure 14 visualizes the feature-based transitions of topmost developers during the counterfactual simulation. The plot collectively demonstrates heterogeneous developer trajectories and identifies the feature modifications associated with transitions from LSMR to HSR states. Most important finding is that directionality is feature-specific across the evaluated developers. Features that generally increased values for positive transitions included Tech_Innov, Persp_Diversity, Work_Discipline, Goal_Tracking, Collab_Energy, Team_Initiative, User_Empathy and Flexibility while moderated features included Tech_Stress and Error_Worry. In an instance, Tech_Stress witnessed a 5 $\rightarrow$ 4 moderation for optimal result, just like a 4 $\rightarrow$ 3.5 decline on Error_Worry in another instance. Conversely, Goal_Tracking (2 $\rightarrow$ 3.5), User_Empathy (1 $\rightarrow$ 3), Flexibility (2 $\rightarrow$ 3.5) and Team_Initiative (1 $\rightarrow$ 2.5) increased metrics at relevant scenarios.

An overview of observations returned actionable insights from the counterfactual simulation. It was observed that a developer-specific intervention profiling played out and simulated counterfactual configurations were strongly aligned with the XGBoost's learned HSR region, thereby making new HSR probabilities very high. Large improvements likewise occurred mainly among initially low-HSR-probability developers, suggesting that the framework has greatest managerial value for developers whose baseline profiles were genuinely vulnerable. It is also noted that improvements frequently involved several simultaneously features, indicating resilience as a multidimensional construct, rather than attributable to one isolated feature. The simulation also retained some feature metrics, reasoning that a developer does not necessarily need improvement in every dimension to reach HSR. Observation also shows that some changes were in the negative direction as the model does not simply prescribe upward increase. Observed empirical counterfactual target ranges exhibited by features are presented on Table 18. Therefore, based on the observed patterns, we formulate an empirical HSR-developer behavioural pattern as one exhibiting low-to-moderate Tech_Stress and Error_Worry, combined with high technological innovation, perspective diversity, work discipline, goal tracking, collaborative energy, team initiative, user empathy and flexibility. Whereas, consequential risk outlook is as outlined on Table 19. Follow-on policy principle suggests a strategy that shifts organisations stress management approach from '*reduce developer stress*' to *optimise multidimensional developer resilience profile*.

Table 16 Feature influence on counterfactual stress-resilience intervention

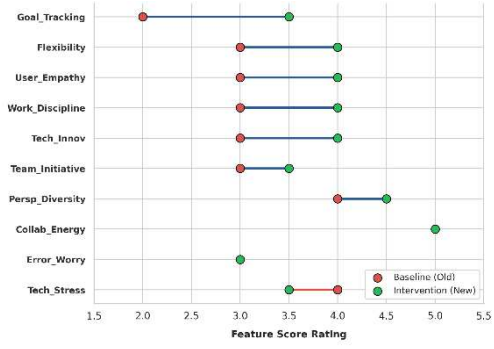
Feature	Mean _Baseline_Prob.	Mean_Cou nterfactual_Prob.	Mean_Prob._ Change	Mean_A bsolute_Prob._C hange	Positive_St ate_Transi tions	Positive_Prob._ Improvement_ %	Positive_St ate_Transi tion_%	Influe nce_R ank
User_Empathy	0.4221	0.6248	0.2027	0.2027	89.4737	4	21.0526	1
Persp_Diversity	0.4221	0.5913	0.1691	0.1691	47.3684	6	31.5789	2
Goal_Tracking	0.4221	0.549	0.1268	0.1268	42.1053	3	15.7895	3
Tech_Innov	0.4221	0.5422	0.12	0.12	42.1053	3	15.7895	4
Work_Discipline	0.4221	0.5402	0.1181	0.1181	47.3684	3	15.7895	5
Flexibility	0.4221	0.5	0.0779	0.0779	73.6842	1	5.2632	6
Error_Worry	0.4221	0.3958	-0.0263	0.0271	15.7895	0	0	7
Collab_Energy	0.4221	0.4334	0.0113	0.0113	31.5789	0	0	8
Tech_Stress	0.4221	0.4138	-0.0083	0.0083	0	0	0	9
Team_Initiative	0.4221	0.4221	0	0	0	0	0	10

Table 17 Comparative developer dynamics for topmost transitions

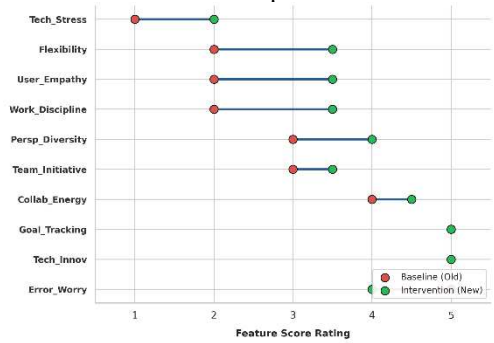
Developer_ DT_ID	Baseline_ Probability	New_ Probability	Probability_ Improvement	Probability_ Improvement_%	Positive_ State_ Transition	Transition_ Degree
DT_DEV_0037	0.977912	0.999519	0.021606	2.160645	TRUE	0.021606
DT_DEV_0038	0.027457	0.99543	0.967972	96.79723	TRUE	0.967972
DT_DEV_0103	0.135206	0.932789	0.797583	79.7583	TRUE	0.797583
DT_DEV_0129	0.187866	0.991805	0.803938	80.39384	TRUE	0.803938
DT_DEV_0130	0.090278	0.964976	0.874698	87.46982	TRUE	0.874698
DT_DEV_0133	0.075048	0.992661	0.917613	91.76131	TRUE	0.917613
DT_DEV_0136	0.094694	0.999106	0.904412	90.44122	TRUE	0.904412
DT_DEV_0142	0.071248	0.895789	0.824541	82.45409	TRUE	0.824541
DT_DEV_0152	0.12146	0.959131	0.837671	83.76705	TRUE	0.837671
DT_DEV_0155	0.019491	0.93521	0.915719	91.57192	TRUE	0.915719
DT_DEV_0241	0.860346	0.998832	0.138486	13.84862	TRUE	0.138486
DT_DEV_0259	0.982529	0.998598	0.016069	1.606864	TRUE	0.016069
DT_DEV_0277	0.94858	0.998726	0.050146	5.014598	TRUE	0.050146
DT_DEV_0341	0.173826	0.98818	0.814354	81.43541	TRUE	0.814354
DT_DEV_0379	0.114235	0.999106	0.884871	88.48713	TRUE	0.884871
DT_DEV_0441	0.175285	0.997488	0.822204	82.22038	TRUE	0.822204
DT_DEV_0455	0.983542	0.998799	0.015257	1.525724	TRUE	0.015257
DT_DEV_0456	0.983542	0.998799	0.015257	1.525724	TRUE	0.015257
DT_DEV_0529	0.998146	0.998545	0.0004	0.039983	TRUE	0.0004



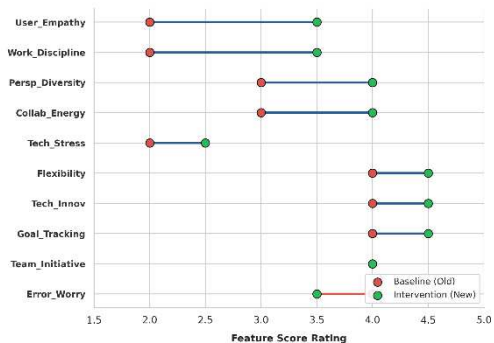
Pre-post feature score trajectory for DT_DEV_0037 developer



Pre-post feature score trajectory for DT_DEV_0038 developer



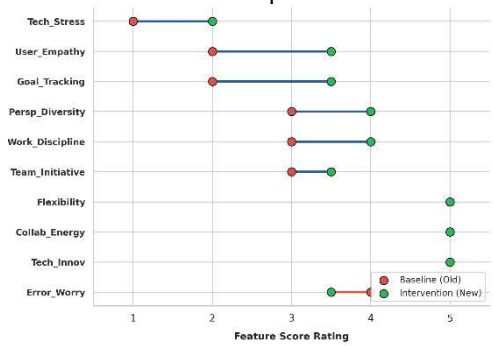
Pre-post feature score trajectory for DT_DEV_0103 developer



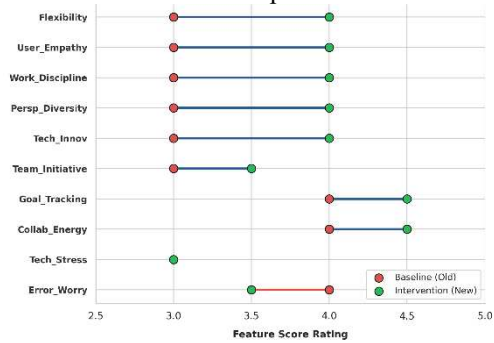
Pre-post feature score trajectory for DT_DEV_0129 developer



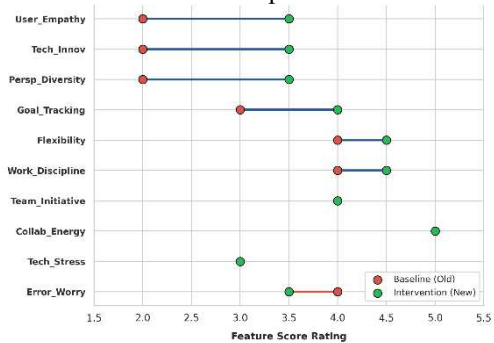
Pre-post feature score trajectory for DT_DEV_0130 developer



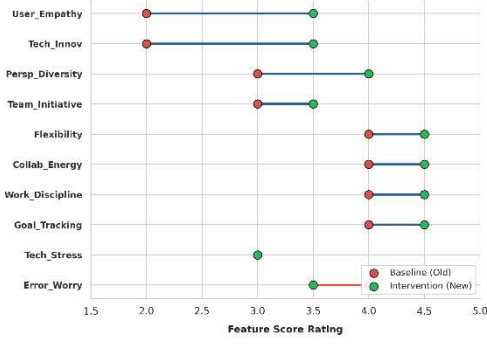
Pre-post feature score trajectory for DT_DEV_0133 developer



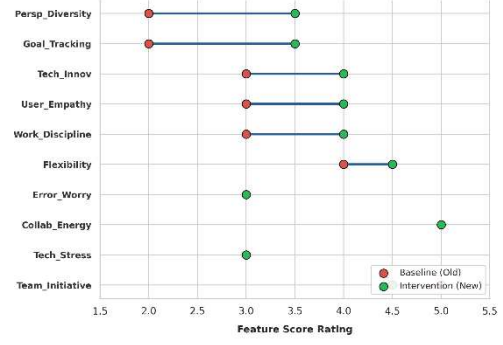
Pre-post feature score trajectory for DT_DEV_0136 developer



Pre-post feature score trajectory for DT_DEV_0142 developer



Pre-post feature score trajectory for DT_DEV_0152 developer



Pre-post feature score trajectory for DT_DEV_0155 developer

Fig. 14 Individual-Level Counterfactual Transitions Across Stress–Resilience Features. Dumbbell plots quantify the counterfactual displacement of each stress–resilience feature for individual developers, with red and green endpoints denoting the observed baseline value x_{ij} and intervention-induced counterfactual value x'_{ij} , respectively. For developer i and feature j , the connecting displacement represents $\Delta x_{ij} = x'_{ij} - x_{ij}$, thereby capturing both the direction and magnitude of the prescribed feature modification. The resulting trajectories characterize heterogeneous counterfactual pathways through feature space and show how coordinated changes in stress-, cognitive-, behavioural-, and interpersonal dimensions are associated with the model-predicted transition from LSMR to HSR.

Table 18 Counterfactual feature-level intervention effects on the predicted probability of transition toward the High-Stress Resilient state.

Feature	Direction toward HSR	Observed successful HSR target values	Provisional management interpretation
Tech_Stress	↓ / moderate	2 – 4	Technology-related stress must be kept relatively controlled
Error_Worry	↓ / controlled	2 – 3.5	Reduction of excessive error-related apprehension
Tech_Innov	↑	3.5 – 5	Strong technological adaptability
Persp_Diversity	↑	3.5 – 5	Inspire diverse viewpoints
Work_Discipline	↑	3.5 – 5	Robust work disciplinary ethics
Goal_Tracking	↑	3.5 – 5	Strong goal alignment
Collab_Energy	↑	4 – 5	Strong collective engagement
Team_Initiative	↑	2.5 – 4.5	Inclusive team management
User_Empathy	↑	3 – 5	Steady user-centric orientation
Flexibility	↑	3 – 5	High adaptability

Table 19 Conceptual feature behavioural risk profile inferred from the intervention directions

Dimension	Empirical HSR outlook	LSMR-risk outlook
Tech_Stress	Low–moderate	High
Error_Worry	Low–moderate	High

Tech_Innov	High	Low–moderate
Persp_Diversity	High	Low–moderate
Work_Discipline	High	Low–moderate
Goal_Tracking	High	Low
Collab_Energy	High	Low–moderate
Team_Initiative	Moderate–high	Low
User_Empathy	High	Low
Flexibility	High	Low

4.8 Discussions

The transition from LSMR to HSR represents a desirable intervention outcome since the HSR state indicate a relatively more favourable blend of stress regulation and resilience-supporting resources within the study population. The objective is therefore not to eliminate all occupational stress, which would be impractical in software development, but to enable functionally resilient and adaptive developers while managing inevitable technical and organisational demands. Results from the counterfactual simulation demonstrated this objective quantitatively since interventions that amplified selected protective features raised the predicted probability of HSR, with User_Empathy (+0.2027), Persp_Diversity (+0.1691), Goal_Tracking (+0.1268), Tech_Innov (+0.1200), Work_Discipline (+0.1181), and Flexibility (+0.0779) driving the positive changes. Consequently, recorded LSMR → HSR is a model-predicted movement toward a more adaptive and resource-rich stress–resilience profile, rather than as evidence that the intervention has causally transformed an individual's psychological state.

4.8.1 Predictors of developer stress and stress-resilient transition

The results indicate that developer stress–resilience in investigated population is not determined by a single attribute but emerges from the interaction of technology-related stress, cognitive–emotional responses, adaptive capacity, work regulation, and social–human-centred capabilities. The analysis on Table 20 sheds light on the observation that resilience is strengthened not merely by suppressing stress, but by simultaneously developing adaptive, self-regulatory, innovative and interpersonal capabilities. Accordingly, the features are interpreted as predictive/intervention-sensitive factors rather than causal determinants.

Table 20 Overview of stress and resilience predictors across functional roles

Functional role	Principal features identified	Empirical interpretation
Stress-related vulnerabilities	Tech_Stress, Error_Worry	Higher technology-related stress and excessive concern about errors are associated with a less favourable stress–resilience profile and therefore represent dimensions requiring control.
Resilience-supporting resources	Flexibility, User_Empathy, Persp_Diversity	Adaptability, user-centred orientation and openness to diverse perspectives constitute important resources for maintaining a more resilient developer profile.
Performance/self-regulation resources	Goal_Tracking, Work_Discipline, Tech_Innov	Goal orientation, disciplined work behaviour and technological innovativeness support movement toward the HSR profile.
Social/team resources	Collab_Energy, Team_Initiative	Collective engagement and proactive participation in teams contribute to a more favourable stress–resilience profile, although

4.8.2 Gender-responsive interpretation for female software developers

These findings have particular relevance for female software developers, for whom technical performance occurs within a professional environment that may simultaneously involve workload pressures, team dynamics, expectations of competence, work–life responsibilities and, for some women, biologically associated life-course conditions. The present findings do not establish that female developers possess different levels of these characteristics; rather, they identify intervention dimensions that can be incorporated into a gender-responsive DT framework. The strongest intervention-sensitive factors provide a useful basis for this gender-responsive interpretation. User Empathy and Perspective Diversity, which produced the largest positive probability changes, suggest that interpersonal and perspective-oriented capabilities should not be treated as peripheral attributes of technical work. Similarly, Goal Tracking, Technical Innovation, Work Discipline and Flexibility indicate that interventions supporting autonomy, adaptive working practices, structured goal management and continuous technological learning may strengthen the resilience profile of female developers. Conversely, Tech Stress and Error Worry identify psychological and technology-mediated pressures that should be monitored. This is particularly relevant because women's professional routines may, depending on individual circumstances, intersect with menstrual-cycle-related symptoms, pregnancy, postpartum recovery, menopause, caregiving responsibilities and other work–life demands. These conditions should not be assumed to apply uniformly to female developers, nor should they be treated as direct explanations for the observed model states. Rather, they reinforce the need for flexible, context-sensitive interventions capable of accommodating heterogeneous working circumstances. The gender-responsive contribution of the framework therefore lies less in claiming that women experience stress through fundamentally different mechanisms and more in providing a mechanism for identifying and adapting interventions to the circumstances of individual female developers. The model's feature-level counterfactual capability is particularly relevant here: rather than assigning a developer to a static “stressed” category, the DT can identify modifiable dimensions, such as flexibility, goal management, error-related worry, technological stress and interpersonal orientation, and estimate their potential effect on the individual's predicted state. This shifts the contribution from gender-neutral prediction to gender-responsive decision support. For example, where a female developer's elevated stress coincides with demanding work routines or temporary life-course constraints, an intervention may prioritise flexibility and workload adaptation. Also, whenever error-related apprehension is prominent, supportive development and psychologically safe feedback may be more appropriate. In the face of weak social or collaborative resources, team-integration mechanisms may be prioritised. Importantly, these are personalised intervention pathways, not assumptions that all female developers share the same needs.

4.8.3 Discussions on implications for software development process models and life cycle management

Empirical HSR target ranges profile suggests that developer resilience should become an operational quality attribute of the development process, thereby ensuring that development process model should evolve from measuring predominantly *Time* → *Cost* → *Quality* → *Productivity* towards *Time* → *Cost* → *Quality* → *Productivity* → *Developer Resilience*. Inferred

policy framework for development units is therefore analysed on Table 21, with its implications for development process models on Table 22. The counterfactual analysis thereby indicates that transition toward HSR is associated with a coordinated multidimensional profile characterised by controlled technology-related stress and error worry, alongside stronger innovation, perspective diversity, work discipline, goal tracking, collaboration, initiative, user empathy and flexibility. The magnitude of predicted improvement is highly developer-specific, demonstrating that resilience-oriented interventions should be personalised rather than uniformly applied. Important methodological insight suffixes such indicating that successful counterfactual transitions do not, by themselves, establish causal feature thresholds. They identify model-consistent target configurations.

Table 21 Inferred policy framework for increasing HSR across development units

Development unit	Primary HSR levers	Practical organisational intervention	Expected resilience mechanism
UI/UX	User_Empathy, Flexibility, Persp_Diversity	User research, usability testing, inclusive design reviews	Reduces uncertainty and improves adaptive problem-solving
Software Engineering	Goal_Tracking, Work_Discipline, Tech_Innov	Sprint goals, coding standards, learning time, technical experimentation	Improves control and technical confidence
QA/Testing	Error_Worry, Work_Discipline, Flexibility	Blameless defect reviews, automated testing, test automation training	Converts errors from threat signals into learning signals
DevOps/SRE	Tech_Stress, Flexibility, Team_Initiative	Incident automation, runbooks, realistic on-call rotations, post-incident learning	Reduces operational overload
Data Science/AI	Tech_Innov, Flexibility, Error_Worry	Experimentation environments, peer review, reproducibility protocols	Increases confidence under uncertainty
Product Management	Goal_Tracking, Persp_Diversity, User_Empathy	Clear objectives, stakeholder alignment, evidence-based prioritisation	Reduces ambiguity and competing demands
Cybersecurity	Tech_Stress, Error_Worry, Team_Initiative	Incident simulations, playbooks, team response drills	Builds preparedness without excessive threat anxiety
Project/Team leadership	Collab_Energy, Team_Initiative, Goal_Tracking	Psychological safety, workload balancing, shared objectives	Strengthens collective resilience

Table 22 Implications for SDPM/SDLC management

SDPM approach	Inferred Recommendations	Implications for industry standard practices
Agile	Treat resilience as a continuous team-development variable	Sprint retrospectives should examine not only velocity and defects but also Goal Tracking, Flexibility, Collaboration and Tech Stress

Scrum	Integrate developer resilience into empirical inspection and adaptation	Sprint Reviews/Retrospectives can identify developers or teams whose feature profiles are moving toward LSMR and trigger targeted interventions
DevOps	Engineer the work environment to minimise unnecessary Tech_Stress and Error_Worry	Automation, observability, incident playbooks, blameless postmortems and balanced on-call practices become resilience interventions rather than merely technical practices

5. Conclusion

In this study, we developed an empirically evaluated Stress–Resilience Digital Twin framework of surveyed female software developers to model, simulate and explain individual stress–resilience states among female developers. The study responded to observed limitations of largely unvalidated feature space within the data layer of existing Human Digital Twin frameworks. Psychometric validation, latent construct modelling, unsupervised state profiling, supervised prediction, explainable artificial intelligence, and intervention-oriented simulation were therefore integrated into a unified computational framework. The psychometric analysis established the empirical basis for the proposed stress–resilience feature space, while high overall sampling adequacy and significant factorability supported latent-structure analysis. The reliability and construct-validity assessments provided evidence that the derived dimensions modelling-ready. Although the two-indicator constructs produced some limitations, particularly for composite reliability and convergent validity in selected dimensions, the results provide a defensible measurement foundation while highlighting the need for richer indicator sets in future studies. The computational results further demonstrated the feasibility of representing developers through distinguishable stress–resilience states and learning their associated predictive patterns. Unsupervised learning by K-means model provided an empirical basis for developer-state profiling, while ensemble XGBoost enabled individual state prediction with SHAP enhancing model transparency by revealing the relative contribution of the validated constructs to predicted states. Scenario (What-If) and personalized counterfactual simulation stages examined how changes in potentially adjustable dynamics may alter an individual's predicted state. The illustrative DT transition from Low-Stress Moderate Resilience to High-Stress Resilient, accompanied by a probability improvement of 0.967972, demonstrated the potential of this approach for personalised, computationally informed intervention analysis. The study positioned the developer Stress–Resilience DT as a decision-support paradigm rather than a diagnostic system. Its principal contribution lies in connecting validated psychological and behavioural constructs with predictive and explainable machine learning and simulation, thereby providing a pathway from measurement, to profiling, prediction, then explanation, intervention simulation and finally twin updating. The framework offered a foundation for more adaptive and individualized approaches to developer well-being, while its gender-responsive orientation provides a basis for investigating how stressors, resilience resources, and intervention opportunities may differ across developer populations, including female developers. Future approach would validate the framework using larger and more varied longitudinal samples, including the increment of indicators per latent construct, and also incorporate more behavioural and longitudinal developer data. Such extensions would enable the DT to evolve from a cross-sectional analytical representation toward a continuously updated model.

References

- Godliauskas, P., & Šmite, D. (2025). The well-being of software engineers: a systematic literature review and a theory. *Empirical Software Engineering*, 30(1), 35.
- Tulili, T. R., Capiluppi, A., & Rastogi, A. (2023). Burnout in software engineering: A systematic mapping study. *Information and Software Technology*, 155, 107116.
- Palacio, R. R., Córdova, G. M., Martínez, A. C., Borrego, G., & Vizcaíno, A. (2026). Factors to foster resilience against coding stressors of agile software developers. *Journal of Software: Evolution and Process*, 38(6), e70139.
- Zhang, Y., Liu, X., Yan, Q., & Na, M. (2025). Empowering workforces in AI-driven environments: co-skilling, organizational support, and mitigating job insecurity. *Frontiers in Psychology*, 16, 1700129.
- Rezvani, A., & Khosravi, P. (2019). Emotional intelligence: The key to mitigating stress and fostering trust among software developers working on information system projects. *International Journal of Information Management*, 48, 139-150.
- Zeuge, A., Lemmer, K., Klesel, M., Kordyaka, B., Jahn, K., & Niehaves, B. (2023). To be or not to be stressed: Designing autonomy to reduce stress at work. *Work*, 75(4), 1199-1213.
- Chen, X., & Xie, Q. (2025). The relationship between job stress, resilience, emotional intelligence, and job satisfaction among college teachers. *Scientific Reports*, 15(1), 20390.
- Suárez, J., & Vizcaíno, A. (2024). Stress, motivation, and performance in global software engineering. *Journal of Software: Evolution and Process*, 36(5), e2600.
- Perlo, V., Fernando, T., Flintoff, J.M., Perera, M.M.N., Ahamed, S., Schmidt, L., Broszczak, D., Fookes, C., Frentiu, F.D., Hunt, A.P. and Lipp, O.V. (2026). A data-informed multidimensional composite score for stress assessment. *Acta psychologica*, 267, 107168.
- Kallio, J., Vildjiounaite, E., Tervonen, J., & Bordallo López, M. (2025). A survey on sensor-based techniques for continuous stress monitoring in knowledge work environments. *ACM Transactions on Computing for Healthcare*, 6(3), 1-31.
- Wang, H., Zhang, Y., Cui, X., Liu, B., Niu, D., Yan, T., ... & Li, J. (2025). The Occupational Stress Check and Assessment Questionnaire: development, validity, and reliability. *BMC Public Health*, 26(1), 41.
- Srisuk, N., Koson, N., Rattanaprom, A., Peansungnern, N., Le Grande, M.R., Jackson, A.C., Thompson, D.R. and Ski, C.F. (2026). Psychometric Evaluation of a Thai Version of the Cardiac Distress Inventory. *Nursing in Critical Care*, 31(5), e70605.
- Luijken, K., Groenwold, R. H., Van Calster, B., Steyerberg, E. W., & van Smeden, M. (2019). Impact of predictor measurement heterogeneity across settings on the performance of prediction models: a measurement error perspective. *Statistics in medicine*, 38(18), 3444-3459.
- Samaraweera, S. I. L. D., Adeeba, S., & Ahamed, M. S. (2026, February). The Role of Developer Communication in Analyzing Burnout: A Deep Learning Approach. In 2026 6th International Conference on Advanced Research in Computing (ICARC) (pp. 1-6). IEEE.
- Fariha, F. H., Ishan, I., Zahid, M., & Akram, Z. A. (2025). Detecting developer emotions in GitHub commits using large language models: implications for project health (Doctoral dissertation, Brac University).
- Obi, I., Butler, J., Haniyur, S., Hassan, B., Storey, M. A., & Murphy, B. (2025, April). Identifying factors contributing to “bad days” for software developers: A mixed-methods study. In 2025 IEEE/ACM 47th

International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP) (pp. 1-11). IEEE.

Sajadi, A., Damevski, K., & Chatterjee, P. (2023, November). Towards understanding emotions in informal developer interactions: A gitter chat study. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (pp. 2097-2101).

Bozorgmehr, A., & Weltermann, B. (2023). Prediction of chronic stress and protective factors in adults: development of an interpretable prediction model based on XGBoost and SHAP using national cross-sectional DEGS1 data. *Jmir ai*, 2, e41868.

Arriel, J., Canuto, T., Azevedo, J., Lima, M. V., Mann, P., Garcia, A., & Pereira, J. A. (2026). Understanding developer well-being: measuring mental health and productivity in software teams. *Empirical Software Engineering*, 31(6), 153.

Montes, C. M. (2026). *From Stress to Strength: Well-Being and Resilience in Software Engineering* (Doctoral dissertation, Chalmers University of Technology).

Singh, A., Anish, P. R., & Ghaisas, S. (2024, October). SOFTMENT: Detecting mental health and wellbeing of women in the software sector. In *Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 405-411).

Guenes, P., Tomaz, R., Trinkenreich, B., Baldassarre, M. T., Storey, M. A., & Kalinowski, M. (2025, April). Impostor phenomenon among software engineers: Investigating gender differences and well-being. In *2025 IEEE/ACM Sixth Workshop on Gender Equality, Diversity, and Inclusion in Software Engineering (GEICSE)* (pp. 33-40). IEEE.

Oliveira, A., Freire, S., Canedo, E. D., Mendonça, M., & Rocha, L. (2025, April). Investigating the challenges faced by women on software engineering: A grey literature study. In *2025 IEEE/ACM Sixth Workshop on Gender Equality, Diversity, and Inclusion in Software Engineering (GEICSE)* (pp. 17-24). IEEE.

Chen, H., Chen, J., Li, Z., Liu, Y., & Wang, J. (2025). Digital twin modeling Enabling structure full field data reconstruction by Variable fidelity data fusion. *Mechanical Systems and Signal Processing*, 230, 112619.

Halimeh, M., Jackson, M., Vieluf, S., Loddenkemper, T., & Meisel, C. (2023). Explainable AI for wearable seizure logging: Impact of data quality, patient age, and antiseizure medication on performance. *Seizure: European Journal of Epilepsy*, 110, 99-108.

Hossain, M. I., Zamzmi, G., Mouton, P. R., Salekin, M. S., Sun, Y., & Goldgof, D. (2025). Explainable AI for medical data: current methods, limitations, and future directions. *ACM Computing Surveys*, 57(6), 1-46.

Pawlicki, M. (2023, October). Towards quality measures for xAI algorithms: Explanation stability. In *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 1-10). IEEE.

Kumi, S., Lomotey, R. K., Ray, M., Cunningham, E., & Deters, R. (2025, August). Explainable Data-Driven Digital Twin for Stress Management. In *International Conference on Advances in Social Networks Analysis and Mining* (pp. 43-60). Cham: Springer Nature Switzerland.

AR, S., & Mathumathi, R. (2025, October). Digital Twin based Smart Safety for Occupational Stress Management. In *2025 10th International Conference on Communication and Electronics Systems (ICCES)* (pp. 2195-2202). IEEE.

- Ezcurra, V., Margondai, A., Von Ahlefeldt, C., Willox, S., Hani, S., Diaz, A. A., ... & Mouloua, M. (2025). Engineering mental wellness: a digital twin for chronic stress modeling and real-time intervention. MODSIM World.
- Kobryn PA (2020) The digital twin concept. In *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2019 Symposium* (pp. 17–24). National Academy of Engineering. <https://doi.org/10.17226/25620>
- Boschert S, Rosen R (2016) Digital twin - the simulation aspect. In P. Hehenberger & D. Bradley (Eds.), *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and Their Designers* (pp. 59–74). Springer Nature. https://doi.org/10.1007/978-3-319-32156-1_1
- Corral-Acero J, Margara F, Marciniak M, Rodero C, Loncaric F, Feng Y, Gilbert A, Fernandes JF, Bukhari HA, Wajdan A, Martinez MV, Santos MS, Shamohammdi M, Luo H, Westphal P, Leeson P, DiAchille P, Gurev V, Mayr M, ... Lamata P (2020) The “Digital Twin” to enable the vision of precision cardiology. *European Heart Journal*, 41(48), 4556-4564B. <https://doi.org/10.1093/eurheartj/ehaa159>
- Barricelli BR, Casiraghi E, Gliozzo J, Petrini A, Valtolina S (2020) Human digital twin for fitness management. *IEEE Access* 8:26637–26664. <https://doi.org/10.1109/ACCESS.2020.2971576>
- Sharotry A, Jimenez JA, Wierschem D, Koldenhoven RM, Valles D, Koutitas G, Aslan S (2020) A digital twin framework for real-time analysis and feedback of repetitive work in the manual material handling industry. In R. Bae, K.H., Feng, B., Kim, S., Lazarova-Molnar, S., Zheng, Z., Roeder, T. and Thiesing (Ed.), *Proc. of the 2020 Winter Simulation Conference* (pp. 2637–2648)
- Demirel HO, Irshad L, Ahmed S, Tumer IY (2021) Digital twin-driven human-centered design frameworks for meeting sustainability objectives. *J Comput Inf Sci Eng* 21(3). <https://doi.org/10.1115/1.4050684>
- Olaleye, T. O., Ajayi, T., Dada, O., Aderele, T., Akintunde, S., & Arogundade, O. (2026). Multi-Perspective Ensemble Stratification-Driven Quality of Life Assessment for Transformative Planning of Coastal Communities. <https://orcid.org/0000-0001-6222-7575>
- Panfilova, A., Bolshev, V., Mozikov, M., Latynov, V., Vanin, A., Nestik, T., Kozin, M., Serohvostov, A., Tarasova, E. and Nikolenko, S., (2026). Toward Psychological Digital Twins: Evaluating Large Language Models for Simulating Assessment Responses from Interview Data. *IEEE Access*.
- Vallu, V. R., Samudrala, V. K., & Pulakhandam, W. (2025). AI-driven digital twin framework for accurate mental health stress detection and personalized management. In *Accelerating Product Development Cycles With Digital Twins and IoT Integration* (pp. 377-408). IGI Global Scientific Publishing.
- Tan, Y., & Wei, J. (2025, June). 28 Exploring intervention effects of digital twin city stress visualization systems on mental crisis early warning for occupational populations. In *Current Opinion in Psychiatry* (Vol. 38, No. e-Supplement 1, pp. e8-e9). LWW.
- Olaleye, T. O., Arogundade, O. T., Abayomi-Alli, A., Aborishade, D. A., & Adeniran, O. J. (2025). Feature engineering of sentiment polarities for bias/variance tradeoff consideration in software defect prediction modelling. *Cureus Journals*, 2(1).
- Wen, J., Gabrys, B., & Musial, K. (2022). Toward digital twin oriented modeling of complex networked systems and their dynamics: A comprehensive survey. *Ieee Access*, 10, 66886-66923.
- Henley, S. S., Golden, R. M., & Kashner, T. M. (2020). Statistical modeling methods: challenges and strategies. *Biostatistics & Epidemiology*, 4(1), 105-139.

Liu, F., Ju, Q., Zheng, Q., & Peng, Y. (2024). Artificial intelligence in mental health: Innovations brought by artificial intelligence techniques in stress detection and interventions of building resilience. *Current Opinion in Behavioral Sciences*, 60, 101452.

Lin, Y., Chen, L., Ali, A., Nugent, C., Cleland, I., Li, R., ... & Ning, H. (2024). Human digital twin: A survey. *Journal of Cloud Computing*, 13(1), 131.