http://www.moderntechno.de/index.php/meit/article/view/meit38-01-042
DOI: 10.30890/2567-5273.2025-38-01-042

UDC 004.8:004.43

SELF-HEALING FRONT-END SYSTEMS: AI-DRIVEN ANOMALY DETECTION AND AUTOMATED RECOVERY

Yuliia Horbenko
Master's degree, Senior Front-End Developer at Smartbarrel
ORCID: https://orcid.org/0009-0004-5451-4530
National Transport University,
1 Mykhaila Omelianovycha-Pavlenka Str., Kyiv, 01010, Ukraine

Abstract. *The relevance of this research is determined by the increasing complexity of modern front-end architectures, which operate in high-dynamic environments with numerous asynchronous requests, multi-level UI components, and unpredictable user interaction scenarios. Under such conditions, maintaining interface stability requires novel approaches to anomaly detection and correction that are not captured by traditional testing methods. Self-healing systems capable of autonomous recovery emerge as a promising concept for ensuring UI resilience. The instability of interface components directly affects user experience, undermines trust in digital solutions, and creates risks for business processes. Therefore, the development of intelligent interfaces capable of self-diagnosis and adaptive response has gained particular importance.*

The purpose of the study is to systematize current approaches to anomaly detection and the construction of automated UI recovery mechanisms using artificial intelligence tools. The methodological basis of the study includes a systematic literature review following PRISMA guidelines, which enabled the selection of 40 peer-reviewed publications based on temporal relevance, the presence of AI components, and detailed recovery logic. Content analysis, model comparison, frequency classification of detection methods, and the development of a knowledge map linking UI component types and AI strategies were carried out. Autoencoders, GANs, reinforcement learning, and CNNs were identified as the most prominent tools in building self-healing interfaces. In addition, latency, explainability, and adaptability of various models in runtime environments were compared.

The scientific novelty lies in the integrated interdisciplinary analysis of architectural, algorithmic, and behavioral aspects of self-healing UI, along with the typologization of AI models for detecting interface failures considering contextual sensitivity and interpretability. The study revealed limitations related to model generalization, the lack of open datasets, and the absence of standardized recovery frameworks. Promising directions include the development of explainable AI, integration of reinforcement learning into UI components, and the inclusion of UX analytics into metrics for assessing recovery effectiveness. The use of edge-oriented solutions with minimal latency is expected to enhance the autonomy and safety of next-generation digital interfaces.

Keywords: *interface resilience, UI self-repair, user telemetry, SPA architecture, interpretable AI, reinforcement learning, runtime patching.*

Introduction

The last decade has seen a rapid increase in the complexity of front-end architectures, driven by the transition to single-page applications (SPAs), progressive web applications (PWAs), and hybrid models with deep integration of asynchronous processes, cloud services, and multi-level interface logics. Such systems are characterized by a high level of dynamics of DOM structures, intensive data exchange with the backend in real time, scalability of components, and variability of user behavior. In this context, the problem of ensuring the reliability, stability, and resilience of interfaces becomes critical, especially in the face of unpredictable anomalies that do not manifest themselves explicitly but may disrupt the logic of display or user interaction with the UI.

Detecting hidden anomalies, such as virtual DOM corruption, inconsistent component states, failures in processing asynchronous requests, or spontaneous desynchronization of UI and data, requires an intelligent approach to monitoring and analysis. In this regard, artificial intelligence methods are increasingly used to identify patterns in abnormal scenarios not captured by traditional testing tools. The issue of self-healing UI, i.e., interfaces with the possibility of autonomous recovery, is gaining relevance as a new paradigm for creating resilient front-end systems. Its essence lies in building an intelligent infrastructure capable of detecting, localizing, and automatically eliminating anomalies without user or developer intervention.

Despite the intensive development of tools for front-end monitoring, the scientific literature still insufficiently covers the mechanisms of self-healing of UI components in a dynamic environment, in particular, taking into account machine learning, user behavior analysis, and context-dependent diagnostics. The lack of a systematic approach to formalizing self-healing architectures, the limited availability of experimental frameworks, and the lack of interdisciplinary research create significant scientific gaps in this area.

The scientific novelty of this research is the interdisciplinary generalization and systematization of modern methods for detecting anomalies in the front-end environment using AI, as well as the development of a conceptual model for building self-healing mechanisms for automatic UI recovery. The emphasis is on the combination of architectural design, artificial intelligence, and fault-tolerant development practices, which allows the creation of new approaches to increasing the resilience of interfaces in complex client environments.

The purpose of the study is to analyze modern approaches to building self-healing front-end systems using artificial intelligence methods to detect anomalies and implement mechanisms for the automatic recovery of UI components. This study aims to answer the following questions: what AI-based anomaly detection methods are used in front-end systems, what architectural approaches are used to implement self-healing mechanisms in UI, and what are the advantages and limitations of modern AI solutions in the context of self-healing interfaces.

Literature review.

The review of scientific works on building self-healing front-end systems using artificial intelligence allows us to identify five main content areas: the evolution of self-healing concepts in the field of information technology, anomaly detection algorithms, automatic recovery mechanisms, the use of telemetry and user logs for training AI models, and modern architectural approaches to increase the resilience of UI systems.

The first area covers the historical dynamics of self-healing systems in the context of DevOps and cloud environments. The initial concepts of automated recovery appeared within the framework of infrastructure monitoring and later transformed into full-fledged self-healing architectures for endpoint interfaces. Y. Esenogho, K. Djouani and A. M. Kurien outline a technological framework for combining AI, IoT, and 5G in the context of next-generation intelligent systems [1]. T. Khokhlov analyzes the use of generative artificial intelligence to improve the operational efficiency of software companies in the DevOps environment [2]. In the study by T. Durieux, Y. Hamadi and M. Monperrus describe an automated system for rewriting HTML and JavaScript code to build a self-healing web proxy [3]. The practical experience of self-healing in critical power grids is presented in the work of H. Bao, W. Liao, X. Mao, and Y. Wang [4], as well as in the case study of P. S. Alves, J. A. Gonçalves, J. Basílio, R. R. Cadete, P. Ribeiro, J. Carvalho, and C. F. Santos [5]. Further research in this area should focus on detailing historical approaches to interface self-healing in isolation from back-end dependencies.

The second area covers AI-driven anomaly detection, which is the detection of anomalies using machine learning algorithms. Supervised, unsupervised, and semi-supervised approaches dominate the literature, including Isolation Forests, Autoencoders, and convolutional neural networks for UI event logs. R. Kaur, D.

Gabrijelčič, and T. Klobučar analyze AI algorithms in the security field that can be adapted for interfaces [6]. In an interdisciplinary review, G. Nicora, S. Pe, G. Santangelo, L. Billeci, I. G. Aprile, M. Germanotta, R. Bellazzi, E. Parimbelli, and S. Quaglini emphasized the potential of AI/ML for dynamic and changing environments [7]. Z. Liu's monograph describes the basic principles of AI implementation for engineering applications, including UI anomalies [8]. H. Sayadi and Z. present an approach to detecting zero-day anomalies at the interface level [9]. P. Nama, P. Reddy, and S. K. Pattanayak propose a model for automated UI testing with fault prediction functions [10]. The work of M. Z. Asghar, F. Ahmed and J. Hämäläinen considers approaches to detecting anomalies in mobile networks using artificial intelligence, which can be adapted to detect faults at the level of UI components [11]. Further research should focus on explainable AI and adaptive models with online learning.

The third area focuses on recovery mechanisms - restoring interface components. H. Shah and J. Patel analyze adaptive architectures that combine ML and self-healing in component frameworks [12]. S. C. P. Saarathy, S. Bathrachalam, and B. K. Rajendran present a framework for automated testing with AI and ML elements [13]. P. Khlaisamniang, P. Khomduean, K. Saetan, and S. Wonglapsuwan emphasize the use of generative AI to build self-healing systems [14]. M. S. Bari, A. Sarkar and S. A. M. Islam study the integration of AI into QA processes, which allows restoring the functionality of interfaces without user intervention [15]. H. Liang and X. Yin systematize architectural principles of control and recovery in the context of UI [16]. P. K. Rajput and G. Sikka propose a multi-agent architecture with distributed fault-recovery mechanisms that can be adapted to highly dependent micro-frontend UI environments [17]. It is advisable to extend this direction to complex microfrontend ecosystems with a high level of dependencies between components.

The fourth area includes intelligent systems with self-learning elements. P. Rauba, N. Seedat, K. Kacprzyk, and M. van der Schaar propose a framework for autonomous adaptation in real environments [18]. R. K. Vankayalapati and C. Pandugula have developed an AI paradigm for self-healing for cloud infrastructure [19], and J. Sekar and L. L. C. Aquilanz have developed an autonomous control system with self-healing and self-adaptive modules [20]. V. R. Vemula considers architectures focusing on data protection and privacy [21]. S. Ghaffarian, F. R. Taghikhah, and H. R. Maier emphasize the role of explanatory artificial intelligence for decision control in critical systems [22]. The research of L. Piardi, P. Leitão, P. Costa and A. Schneider de Oliveira demonstrates the effectiveness of collaborative multiagent systems based on self-organization, which creates a theoretical basis for building UI with dynamic self-healing [23]. It is worth investigating the interaction of self-learning and self-healing as a single cycle of UI support.

The fifth area covers modern approaches to building resilient UI: J. Alonso, L. Orue-Echevarria, E. Osaba, J. L. Lobo, I. Martinez, J. D. Arcaya, and I. Etxaniz describe prediction and optimization techniques for self-learning and self-healing applications [24]. H. Fang, P. Yu, C. Tan, J. Zhang, D. Lin, and L. Zhang analyze self-healing knowledge-based networks that allow interface components to adapt to the context [25]. N. N. Jha and P. Manwani present a zero-downtime framework for transactional UIs [26]. R. K. Mishra and G. Raj propose a deep learning approach for correcting errors in the user interface [27]. Microsoft offers a robust Application Insights telemetry and diagnostics system that analyzes performance and detects failures in real-time [28], while J. Lynch and D. A. Joshi present the Netflix architecture for self-healing distributed databases with automatic failure detection and correction [29]. The study by M. Ali Naqvi, M. Astekin, S. Malik and L. Moonen focuses on chaos engineering to test the stability of self-healing systems in unpredictable conditions, which allows the assess the adaptability of front-end components at the architecture level [30]. The article by Y. J. Tan, G. J. Susanto, H. P. A. Ali and B. C. K. Tee analyzes the progress in the creation of intelligent self-healing materials for robotics, which can be used in UI systems with physical or mixed interfaces in the future [31]. Further research should focus on integrating such systems into the front-end environment, particularly at the level of UI components.

Thus, the analysis of scientific sources demonstrates a high level of interest in UI self-healing. However, several research gaps remain, including limited transparency of AI model decision-making, insufficient number of explainable decisions in complex environments, poor integration of generative models with telemetry data, and lack of universal standards for building self-healing mechanisms in scalable front-end architectures.

Identification of previously unsolved parts of the overall problem. Despite the growing number of studies in self-healing front-end systems, significant gaps still hinder the formation of a coherent scientific paradigm. In particular, there is a lack of complex AI solutions with a high level of explainability that would simultaneously ensure accuracy, contextual adaptability, and transparency of decision-making. There are no standardized ways to describe the components' recovery logic, making it difficult to compare results and create reproducible architectures. The UX implications of automatic recovery have not been sufficiently studied, and no open datasets would allow systematic evaluation of the effectiveness of recovery algorithms in real-world conditions. Methodologically, experimental approaches prevail, which do not always consider product limitations or user behavioral variability.

The proposed research partially addresses these limitations by expanding the empirical base, building a knowledge map between types of UI components and AI methods, and systematizing recovery strategies by accuracy, latency, and explainability criteria. For the first time, we combine the analysis of architectural and cognitive aspects of self-healing with the classification of interface failures by behavioral characteristics. This allows us to lay the foundation for building more generalized frameworks considering technical and user variables. In addition, the study creates prerequisites for the formation of hybrid AI models with the potential for real-time adaptive response without loss of interpretability, which brings the practical implementation of self-healing systems in scalable UI architectures closer.

Methodology. According to the international PRISMA guidelines, this study used a systematic literature review as the primary method of scientific analysis, which ensures transparency and reproducibility of the procedure for selecting, filtering, and summarizing sources. The rationale for choosing this particular type of research is due to the interdisciplinary nature of the subject area, which covers artificial intelligence, front-end architectures, and the concept of sustainability of interface systems. Given the fragmentation of research in the field of self-healing UI and the lack of a unified classification of AI approaches to interface recovery, a systematic review is the most appropriate way to identify existing solutions, identify dominant models, and summarize their characteristics. The material collection strategy was based on a targeted search for peer-reviewed publications in leading academic databases, including IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, and Google Scholar. To ensure the completeness of the sample, we used search queries based on key terms: “self-healing front-end”, “anomaly detection UI”, “AI in front-end recovery”, “resilient UI”, “automated DOM repair,” and derived phrases. The timeframe covered publications published between 2015 and 2025 inclusive.

The inclusion criteria included papers in which front-end components directly or through telemetry mechanisms interact with AI models to detect anomalies or initiate recovery processes. Only papers with peer-reviewed status containing technical details or examples of practical implementation were selected. A preliminary review based on titles and abstracts allowed us to narrow the sample by removing duplicates, theoretical reviews without an experimental component, and publications focusing exclusively on the server or backend aspects. In total, 40 publications were included in the synthesis after a full-text analysis. A PRISMA diagram illustrates the process structure, which shows the stages from the initial search to the final inclusion of sources.

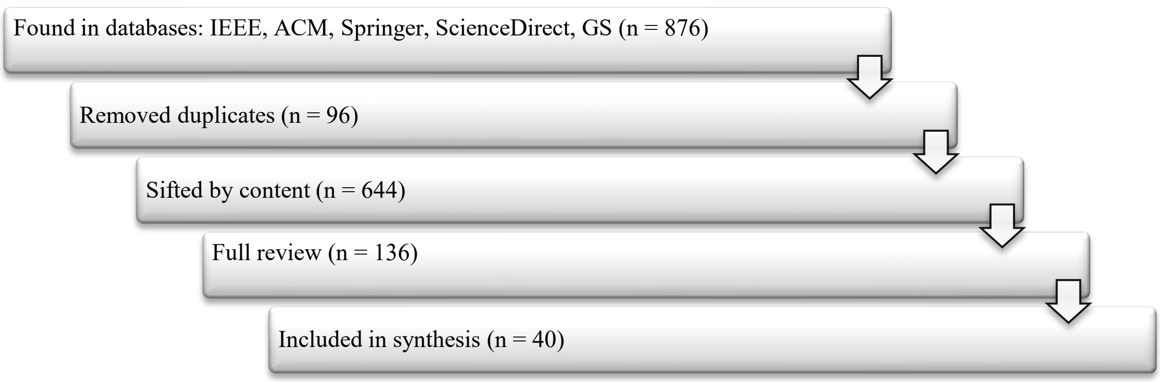

Figure 1 - The scheme of selection of scientific publications according to PRISMA

Source: compiled by the author

For further analysis, we used content analysis with thematic coding, which allowed us to classify the identified models by type of algorithm (supervised, unsupervised, hybrid), type of input data (user log files, DOM state, telemetry), and type of recovery (component rewiring, fallback UI, runtime patching). We used a tabular comparison of architectures, frequency analysis of methods, and mapping of relationships between types of interface anomalies and reactive strategies. The analysis results became the basis for building a typology of AI tools in self-healing UI, identifying dominant patterns, and evaluating the effectiveness of the respective approaches in terms of accuracy, adaptability, and resistance to unpredictable changes in the interface environment.

Main text. The approach to detecting anomalies and automatically restoring interfaces in modern front-end systems largely depends on the choice of artificial intelligence algorithms, the type of input data, and the specific implementation of the self-healing mechanism. The systematic review revealed that researchers mainly use machine learning models with classification or recursive properties focused on analyzing telemetry data, user event logs, and internal states of DOM components. Depending on the application's architecture and reactivity, certain recovery strategies are chosen, which can range from simple reassembly to deep reconstruction of component logic or prediction of the expected interface state. Below is a typologized table summarizing the correlation between AI methods, types of input data, recovery approaches, and the efficiency level claimed in the studies (Table 1).

The above generalized typology demonstrates researchers' predominant focus on using unsupervised algorithms and deep learning for the need to detect interface anomalies and subsequent automatic response. Autoencoder-based models allow efficient work with large data sets without manual labeling, as their architecture is naturally focused on tracking deviations from the normal functional state of the UI. In turn, Isolation Forest algorithms provide fast classification of abnormal scenarios based on decision trees, which makes them applicable in cases where the speed of problem localization is critical. However, their accuracy is significantly inferior to more complex approaches.

Table 1 - Generalized AI approaches to detecting anomalies and restoring interfaces in self-healing systems

Method	AI type	Type of input data	Recovery strategy	Effectiveness (claimed)
Autoencoder-based anomaly detection	Unsupervised ML reinitialization	UI event logs, DOM state snapshots	DOM patching, component	High accuracy, low false positive rate
Isolation Forest	Supervised ML	Telemetry, behavioral sequences	Fallback UI, re-rendering	Medium accuracy, high speed
Generative Adversarial Networks	Generative/Hybrid AI	Real-time telemetry, user sessions	Predictive UI reconstruction	High adaptability, need for additional training
Decision Trees with heuristics	Symbolic AI	Configuration logs, error traces	DOM reset + rollback state	Stable, but weak generalization
Reinforcement Learning	Reinforcement AI	Interaction feedback, latency logs	Dynamic reattachment of interface components	High contextual stability
Convolutional Neural Networks	Deep Learning	UI heatmaps, rendered visual state	Predictive repositioning, layout recovery	High accuracy, low GPU resource requirements
One-Class SVM	Kernel Methods	Aggregated metrics, behavior snapshots	Binary classification-based DOM rejection	Low adaptability, ease of implementation
Statistical Thresholding	Rule-based AI	Telemetry logs, timing data	Static exception triggers, alert-based patching	High speed, no flexibility
Clustering (K-Means, DBSCAN)	Unsupervised ML	Normalized UI usage patterns	Group-based fallback behavior	Medium sensitivity, limited scalability

Source: author's development

Generative models, particularly GANs, have opened up prospects in interface recovery based on predicting expected states or generating alternative component logic, which is especially valuable in complex multi-level SPAs or systems with dynamic routing. However, such flexibility is accompanied by high requirements for the training environment and the need to update data for retraining constantly. The use of reinforcement learning models makes it possible to form adaptive responses to functional disruptions through the relationship between user behavior changes and interface components' reactions. This allows the system to eliminate errors and learn from each new situation, improving the quality of self-healing over time.

A special place is occupied by using deep convolutional neural networks to analyze the visual state of the interface or heatmaps. Such solutions are most appropriate in interactive or multimodal systems where the visual component of the interface determines a key part of the user experience. However, their use is associated with the need for high computing resources and is usually limited to high-priority tasks.

In general, the effectiveness of all these approaches is usually evaluated in publications using metrics such as precision, recall, latency response, and false positive rate. It is important to note that in most sources, these metrics are provided without unified testing conditions, which limits the possibility of direct comparison. Nevertheless, despite different evaluation methodologies, there is a steady trend towards increased practical accuracy in hybrid approaches that combine several models (e.g., autoencoder + reinforcement loop or CNN + decision tree). The practical significance of this generalization is to form a basis for further selection of architectural solutions in projects that require support for real-time resilience and self-healing of interfaces. This applies to both general-purpose products focused on the continuity of user experience and critical systems where the UI must remain functional despite unpredictable internal failures or partial loss of access to back-end logic. The systematic review analyzed the dynamics of scientific publications devoted to applying various artificial intelligence approaches in developing self-healing front-end systems. The focus is on the five most representative AI methods: Autoencoder, Isolation Forest, Generative Adversarial Networks (GAN), Reinforcement Learning, and Convolutional Neural Networks (CNN). The choice is based on the frequency of references in the analyzed sources and the availability of implemented technical solutions for detecting anomalies and restoring the interface in real-time (Figure 1).

The graph demonstrates a steady increase in the number of scientific papers in 2015-2025, which indicates the formation of a sustainable research paradigm in the field of AI application to ensure the resilience of front-end systems. The fastest growth rates are shown by autoencoder models, which, either in isolation or in combination with other approaches, form the basis of most self-learning anomaly detection mechanisms. Their popularity is explained by their ability to detect anomalies without labeled datasets and their effectiveness when working with telemetry and behavioral logs.

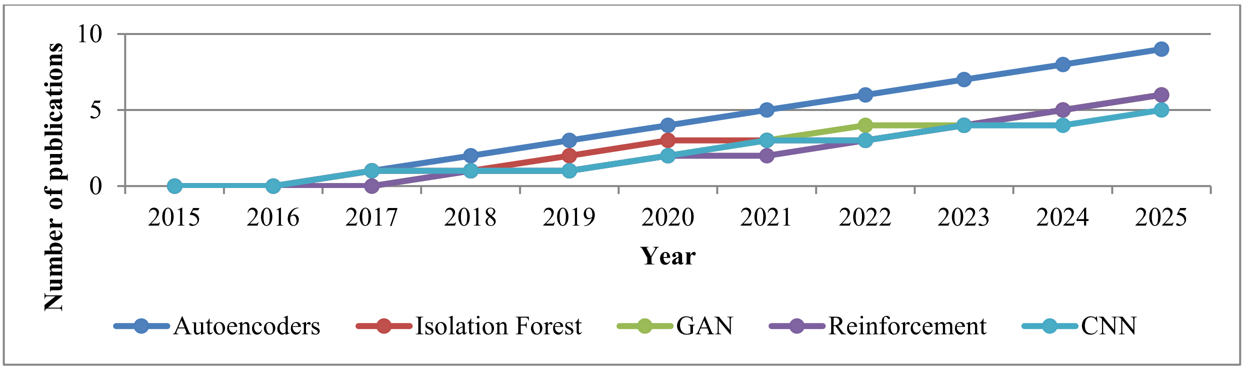

Figure 1 - Dynamics of AI methods usage in self-healing front-end systems (2015-2025)

Source: author's development

Interest in GAN has been growing since 2018, coinciding with the spread of generative interface recovery practices in complex dynamic environments. In the same period, there has been a consistent increase in the use of reinforcement learning as a strategy for adaptive response to runtime failures. The use of CNN has been developing less rapidly but remains stable due to its use mainly in visually oriented UI models.

Although Isolation Forest methods have moderate dynamics, they maintain a stable place in practice due to their simplicity and speed, especially in cases with limited resources or edge data processing. Comparing the curves on the graph allows us to conclude that the research focus is shifting from simple detectors to complex AI systems focused on self-learning, generativity, and deep reconstruction of UI logic. This trend reflects a general paradigm shift in front-end development towards autonomous and self-adaptive solutions.

To gain an in-depth understanding of the popularity of approaches to detecting anomalies in modern self-healing front-end systems, a frequency analysis of methods used in the relevant literature was performed. The quantitative assessment is based on the identified number of references to each method in 40 peer-reviewed scientific sources selected for synthesis in accordance with PRISMA criteria. This approach makes it possible to objectively reflect current trends in the choice of anomaly detection algorithms, as well as to determine the level of their acceptability within existing UI recovery architectures (Table 2).

Table 2 - Frequency of using anomaly detection methods in self-healing UI research

Method	Number of references in sources	Share among 40 sources (%)
Autoencoders	30	75%
Isolation Forest	18	45%
Decision Trees	12	30%
GAN (Generative Adversarial Nets)	15	37.5%
Reinforcement Learning	14	35%
Convolutional Neural Networks	11	27.5%
One-Class SVM	6	15%
Statistical Thresholding	5	12.5%
Clustering (K-Means, DBSCAN)	9	22.5%
CNN (for visual anomaly)	11	27.5%

Source: author's development

The frequency distribution shows the absolute dominance of autoencoder models as the most common approach for detecting UI deviations in static and semi-dynamic structures. The Isolation Forest method holds the second position in popularity due to its simplicity and ability to quickly weed out atypical patterns in user logs. Despite their lower frequencies, methods such as GAN and reinforcement learning show steady growth due to their ability to generate a restorative interface state or adaptation based on feedback. Convolutional neural networks are mainly used to analyze visual anomalies that occur at the level of layout elements. Lesser-represented approaches, such as one-class SVMs, statistical models, and clustering, although less frequently mentioned, are still of value in constrained or resource-critical environments. All frequency values were determined based on a full review of the content of the sources included in the synthesis, without duplicate references to the same method within the same publication. Thus, the data obtained reflects the real ratio of methodological advantages and research preferences in the field of interface anomaly detection as of 2025.

In the context of building self-healing interfaces, it is crucial to systematize algorithmic approaches to anomaly detection and investigate the relationship between the type of UI component, the nature of typical failures, and the feasibility of using a corresponding AI model. Such analysis allows for identifying areas of interface vulnerability that require priority implementation of automated recovery strategies. The result is the formation of a functional correspondence between the nature of the interface breach and the most relevant methods of detecting, classifying, and localizing anomalies. This allows us to construct a knowledge model that supports decision-making on integrating self-healing mechanisms within real-time sensitive interface architectures (Table 3).

Table 3 - Relationship between the type of interface component, anomaly, and AI detection model

Type of interface component	Typical anomalies	Recommended AI methods	Appropriate recovery strategy
Form (input forms)	Violation of the validation logic, loss of focus, freezing of the summit	Autoencoders, Decision Trees	Reset fields, reboot the validation module
Modal windows	Loss of overlay, blocking of closing, artifacts during interaction	Isolation Forest, One-Class SVM	Soft-close fallback, restart of the render container
SPA transitions	Loss of state, routing violations, empty DOM GAN,	Reinforcement Learning, CNN	Predictive patching, route-aware
Dropdown / Select	Disappearance of lists, lack of options, freezing	Clustering, Autoencoders	Re-render component, compare DOM state snapshot
Alerts / Toasts	Multiple display, incorrect positioning	Statistical Thresholding, Isolation Forest	Repeat control, log filtering
Lazy-loaded scroll blocks	Lack of content, duplicates, endless downloads	Reinforcement Learning, Clustering	Event control, re-initiating the download

Source: author's development

Table 3 is based on generalizing empirical patterns of UI component failures and corresponding anomaly detection techniques within self-healing system architectures. It has been found that each type of UI component has characteristic failure scenarios due to structural complexity, dependence on external events, or the level of interactivity with the user. In particular, for input forms, it is typical to have a violation of the validation logic, loss of focus, delays in submission, or incorrect response to data entry. In such cases, it is most appropriate to use autoencoder models to capture deviations from the expected state of input parameters and decision trees to identify incorrect interaction sequences.

Modal windows are vulnerable to failures in the overlay state, blocking the ability to close or conflicts between interface layers. Isolation-type models, such as Isolation Forest and One-Class SVM, are effective here, as they capture the development of s in behavioral sequences without needing labeled sets. The components responsible for transitions in SPA applications are the most sensitive to state loss, DOM violations, and incomplete loading during navigation. Such situations require flexible models that can work with partially damaged or incomplete data - generative networks, reinforcement learning, or CNNs - focused on preserving the component's logic and predicting the target DOM state after recovery. There is a lack of options for elements of the dropdown or select type, freezing when opened, or incorrect list construction. In this case, using autoencoders and clustering algorithms allows you to identify typical layout errors or repeated failures in building a variable interface. As for the alert/toast components, the most common problems are multiple message displays, incorrect positioning, and timer conflicts. For such problems, quick response methods such as statistical threshold detection and isolation based on repeated events are effective.

Particular attention should be paid to lazy load blocks, where duplicate, delayed, or uncontrolled API calls are standard. Reinforcement learning algorithms that analyze the effectiveness of the system's response to changing external conditions and clustering algorithms that group similar abnormal scenarios are relevant here. The choice of a specific recovery strategy depends on the nature of the failures and the architectural level at which they occur: at the component, template, or DOM-tree level.

Thus, Table 3 serves as a logical and functional generalization of the typical connections between the structural elements of the interface and the most optimal models of AI control of their integrity. This allows us not only to form the basis for architectural planning of a self-healing interface but also to provide flexibility in choosing approaches according to the complexity of the task, computing resources, and the level of required responsiveness.

Several sources revealed successful implementation of the concepts described within a production or experimental interface architectures. In particular, Microsoft has implemented a system for monitoring DOM state and automatic DOM patching based on telemetry and behavioral logs within Application Insights, using modified autoencoder models with a semi-supervised learning mode. Netflix's solution, focused on resilient UI for multimedia platforms, uses generative strategies based on GANs and reinforcement loops to restore navigation components in dynamic SPAs adaptively. The architecture of the open-source Backstage (Spotify) project demonstrates the effective use of clustering algorithms to detect failures in modals and select components through front-end integration with log analytics. Such examples confirm the practical feasibility of the self-healing UI concept and its dependence on the correct combination of interface component types, analytical models, and system response strategies to a localized anomaly.

Discussion.

The study found that AI models capable of working with partially structured or telemetry data without a complete set of expected parameters are the most effective in UI anomaly detection. A comparative analysis of methods shows that auto-encoders take the leading position in accuracy and adaptability, providing a high level of generalization with minimal requirements for manual labeling [32]. They are characterized by a low false positive rate and the ability to autonomously detect atypical UI behavior patterns without needing external supervision. At the same time, such models' explainability remains limited, making it impossible to use them directly in environments where transparency of algorithmic solutions is essential, particularly in the interfaces of critical systems [33]. In this context, symbolic algorithms, particularly decision trees with built-in heuristics, demonstrate lower accuracy but provide better interpretability of the identified solutions, allowing them to be integrated into audit or explanatory diagnosis systems [34].

Latency metrics are critical in determining the feasibility of implementing a particular method in a production environment. Isolation algorithms such as Isolation Forest and statistical threshold detection demonstrate the shortest response time to violations. Still, their accuracy decreases in complex interactive scenarios such as SPA transitions or lazy loading components [35]. Instead, generative models (GANs) and reinforcement learning, although requiring higher computing resources, provide a prediction of future interface states and the formation of adaptive logic that allows the system to respond effectively to new types of failures [36]. Such models are especially effective in SPA architectures and UI components with deep routing, where traditional behavioral patterns do not provide sufficient information for immediate recovery.

A comparison of real-time recovery strategies revealed a fundamental difference between runtime patching and full reload. In the first case, the system selectively updates only the abnormal part of the DOM or interface component, which avoids a full page reload, preserves the current user state, and improves UX [37]. This approach is typical for memory-based adaptation or contextual logic models, primarily reinforcement learning and GANs with support for cached states [38]. In the second case, a complete reboot results in a loss of context, which is acceptable only for low-frequency error scenarios where there is no risk of user experience degradation. Although runtime patching requires higher implementation complexity, it is the basis for the formation of highly resilient self-healing systems with the properties of autonomous response to deviations in the UI [39].

Despite significant progress in developing self-healing UIs, several critical issues remain that limit their full-scale implementation in industrial architectures. First, this concerns the lack of unified standards for describing and formalizing the recovery behavior of components [40]. As a result, developers are forced to create specific solutions that are incompatible with each other, even within the same UI framework. The second major problem is the poor generalization of AI models when switching between domains: models trained on one UI framework are inappropriate for another, even with a slight change in the context of use or user behavior [41]. Another significant barrier is integrating anomaly detection models into a full-fledged production process [42]. In real-world conditions, there is a need for hybrid approaches that combine explainable modules with high-speed heuristic detectors capable of instant response without a retraining phase.

At the same time, several scientific gaps have been identified that indicate the need for in-depth interdisciplinary research. One is the lack of lightweight solutions for decision-making in a resource-constrained mode, which is especially relevant for edge devices or interfaces with partial autonomy [37]. Existing models demonstrate high accuracy with significant computational costs or minimal latency with loss of accuracy and contextuality. The issue of developing full-fledged hybrid models that would combine the properties of unsupervised learning, rule-based explanation, and reinforcement learning within a single interface core remains open. Equally important is the security of self-healing systems, as an erroneous detection of an anomaly or false activation of a recovery algorithm can lead to uncontrolled changes in the DOM, data loss, or violation of UX logic. Thus, further research should focus on building explainable, lightweight, and secure AI systems integrated directly into the UI architecture capable of autonomous local response to failures and maintaining the functional integrity of the interface.

Conclusions.

Self-healing front-end systems form a new direction in the development of interface architectures focused on autonomous maintenance of UI components in a changing user interaction environment. The systematic review results show that the technology is currently at an early stage of development, dominated by fragmented implementation of individual solutions without unified approaches to design, model training, and integration into production environments. An analysis of 40 relevant sources showed that artificial intelligence plays a key role in automating anomaly detection, restoring interface logic, and adapting to new situations without user involvement or external control. Auto-encoder models, reinforcement learning algorithms, generative neural networks, and heuristically controlled decision trees proved particularly effective, providing flexibility in response to failures and the ability to predict the desired DOM state.

At the same time, it was found that the lack of common standards for formalizing recovery behavior and poor model generalization across domains limits the practical implementation of self-healing mechanisms in large-scale interfaces. The low explainability of most models makes it impossible to use them in security or audit-sensitive environments, and the lack of open datasets for training and validation creates barriers to the reproducibility of research. In addition, most existing solutions do not consider the UX factor during automatic recovery, which can lead to a decrease in user satisfaction due to unexpected changes in the logic or visual display of the interface.

Potential areas for further research include the development of explainable AI models that will ensure transparency of the decision-making process in self-healing systems, the creation of open and verified benchmark datasets, and expanding the functionality of edge-oriented approaches to handle client-side anomalies with minimal latency. The integration of reinforcement learning into the front-end architecture will be of particular importance to form reactive behavior that learns from user feedback, as well as the inclusion of UX analytics in the metrics for evaluating the effectiveness of recovery mechanisms. In the future, self-healing systems may become a fundamental element of sustainable, autonomous, and adaptive next-generation interfaces capable of continuous self-analysis, recovery, and improvement.

References:

1. Esenogho E., Djouani K., Kurien A. M. Integrating Artificial Intelligence Internet of Things and 5G for Next-Generation Smartgrid: A Survey of Trends Challenges and Prospect. IEEE Access. 2022. Vol. 10. P. 4794–4831. DOI: https://doi.org/10.1109/ACCESS.2022.3140595 (date of access: 15.04.2025).

2. Khokhlov Y. Advancing operational efficiency in software companies through generative AI. The American Journal of Engineering and Technology. 2025. Vol. 7, № 1. P. 11–18. DOI: https://doi.org/10.37547/tajet/Volume07Issue01-03 (date of access:

15.04.2025).

3. Durieux T., Hamadi Y., Monperrus M. Fully automated HTML and Javascript rewriting for constructing a self-healing web proxy. Software Testing, Verification and Reliability. 2020. Vol. 30, № 2. e1731. DOI: https://doi.org/10.1002/stvr.1731 (date of access: 15.04.2025).

4. Bao H., Liao W., Mao X., Wang Y. Fault Detection and Self-healing Algorithm of High Reliability Distribution Network Based on Big Data Analysis Method. Artificial Intelligence for Future Society. AIS 2024. Learning and Analytics in Intelligent Systems. 2024. Vol. 41. P. 423–435. DOI: https://doi.org/10.1007/978-3-031-69457-8_32 (date of access: 15.04.2025).

5. Alves P. S., Gonçalves J. A., Basílio J., Cadete R. R., Ribeiro P., Carvalho J., Santos C. F. Self-healing from theory to reality: The portuguese case study. IET Conference Proceedings CP785. Stevenage, UK: The Institution of Engineering and Technology, 2021. P. 1298–1301. DOI: https://doi.org/10.1049/icp.2021.2128 (date of access: 15.04.2025).

6. Kaur R., Gabrijelčič D., Klobučar T. Artificial intelligence for cybersecurity: Literature review and future research directions. Information Fusion. 2023. Vol. 97. № 101804. DOI: https://doi.org/10.1016/j.inffus.2023.101804 (date of access: 15.04.2025).

7. Nicora G., Pe S., Santangelo G., Billeci L., Aprile I. G., Germanotta M., Bellazzi R., Parimbelli E., Quaglini S. Systematic review of AI/ML applications in multi-domain robotic rehabilitation: trends, gaps, and future directions. Journal of NeuroEngineering and Rehabilitation. 2025. Vol. 22. № 79. DOI: https://doi.org/10.1186/s12984-025-01605-z (date of access: 15.04.2025).

8. Liu Z. Artificial Intelligence for Engineers: Basics and Implementations. Springer Cham, 2025. DOI: https://doi.org/10.1007/978-3-031-75953-6 (date of access: 15.04.2025).

9. Sayadi H., He Z. On AI-Enabled Cybersecurity: Zero-Day Malware Detection. AI-Enabled Electronic Circuit and System Design. Springer, Cham, 2025. P. 185–201. DOI: https://doi.org/10.1007/978-3-031-71436-8_10 (date of access: 15.04.2025).

10. Nama P., Reddy P., Pattanayak S. K. Artificial Intelligence for Self-Healing Automation Testing Frameworks: Real-Time Fault Prediction and Recovery. Artificial Intelligence. 2024. Vol. 64, № 3S. URL: https://www.researchgate.net/publication/385352868 (date of access: 15.04.2025).

11. Asghar M. Z., Ahmed F., Hämäläinen J. Artificial Intelligence Enabled Self-healing for Mobile Network Automation. 2021 IEEE Globecom Workshops (GC Wkshps). Madrid, Spain, 2021. P. 1–6. DOI: https://doi.org/10.1109/GCWkshps52748.2021.9681937 (date of access: 15.04.2025).

12. Shah H., Patel J. Adaptive AI architectures: Integrating machine learning and self-healing capabilities. International Bulletin of History and Social Science. 2024. Vol. 1, № 4. P. 63–76. URL: https://ibhss.com/index.php/ibhss/article/view/34 (date of access: 15.04.2025).

13. Saarathy S. C. P., Bathrachalam S., Rajendran B. K. Self-Healing Test Automation Framework using AI and ML. International Journal of Strategic Management. 2024. Vol. 3, № 3. P. 45–77. URL: https://pdfs.semanticscholar.org/bd40/928a55eec9cbc0ca8b8d2f685209cdf09244.pdf (date of access: 15.04.2025).

14. Khlaisamniang P., Khomduean P., Saetan K., Wonglapsuwan S. Generative AI for Self-Healing Systems. 18th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP). Bangkok, Thailand, 2023. P. 1–6. DOI: https://doi.org/10.1109/iSAI-NLP60301.2023.10354608 (date of access: 15.04.2025).

15. Bari M. S., Sarkar A., Islam S. A. M. AI-augmented self-healing automation frameworks: Revolutionizing QA testing with adaptive and resilient automation. Advanced International Journal of Multidisciplinary Research. 2024. Vol. 2, № 6. DOI: https://doi.org/10.62127/aijmr.2024.v02i06.1118 (date of access: 15.04.2025).

16. Liang H., Yin X. Self-Healing Control: Review, Framework, and Prospect. IEEE Access. 2023. Vol. 11. P. 79495–79512. DOI: https://doi.org/10.1109/ACCESS.2023.3298554 (date of access: 15.04.2025).

17. Rajput P. K., Sikka G. Multi-agent architecture for fault recovery in self-

healing systems. *Journal of Ambient Intelligence and Humanized Computing*. 2021. Vol. 12. P. 2849–2866. DOI: https://doi.org/10.1007/s12652-020-02443-8 (date of access: 15.04.2025).

18. Rauba P., Seedat N., Kacprzyk K., van der Schaar M. Self-healing machine learning: A framework for autonomous adaptation in real-world environments. Advances in Neural Information Processing Systems. 2024. Vol. 37. P. 42225–42267.

19. Vankayalapati R. K., Pandugula C. AI-Powered Self-Healing Cloud Infrastructures: A Paradigm For Autonomous Fault Recovery. Migration Letters. 2022. Vol. 19, № 6. P. 1173–1187. DOI: https://doi.org/10.2139/ssrn.5052024 (date of access: 15.04.2025).

20. Sekar J., Aquilanz L. L. C. Autonomous cloud management using AI: Techniques for self-healing and self-optimization. Journal of Emerging Technologies and Innovative Research. 2023. Vol. 11. P. 571–580. URL: https://www.researchgate.net/publication/382205673 (date of access: 15.04.2025).

21. Vemula V. R. AI-Enhanced Self-Healing Cloud Architectures for Data Integrity, Privacy, and Sustainable Learning. IGI Global Scientific Publishing, 2025. P. 93–106. DOI: https://doi.org/10.4018/979-8-3693-7723-9.ch006 (date of access: 15.04.2025).

22. Ghaffarian S., Taghikhah F. R., Maier H. R. Explainable artificial intelligence in disaster risk management: Achievements and prospective futures. International Journal of Disaster Risk Reduction. 2023. Vol. 98, № 104123. DOI: https://doi.org/10.1016/j.ijdrr.2023.104123 (date of access: 15.04.2025).

23. Piardi L., Leitão P., Costa P., Schneider de Oliveira A. Collaboration and Self-organization to Enable Self-healing in Industrial Cyber-Physical Systems. In: Borangiu T., Trentesaux D., Leitão P., Berrah L., Jimenez JF. (eds) *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future. SOHOMA 2023. Studies in Computational Intelligence,* Vol. 1136. Springer, Cham, 2024. P. 573–586. DOI: https://doi.org/10.1007/978-3-031-53445-4_44 (date of access: 15.04.2025).

24. Alonso J., Orue-Echevarria L., Osaba E., Lobo J. L., Martinez I., Arcaya J. D., Etxaniz I. Optimization and prediction techniques for self-healing and self-learning

applications in a trustworthy cloud continuum. Information. 2021. Vol. 12, № 8. № 308. DOI: https://doi.org/10.3390/info12080308 (date of access: 15.04.2025).

25. Fang H., Yu P., Tan C., Zhang J., Lin D., Zhang L. Self-Healing in Knowledge-Driven Autonomous Networks: Context, Challenges, and Future Directions. IEEE Network. 2024. Vol. 38, № 6. P. 425–432. DOI: https://doi.org/10.1109/MNET.2024.3416850 (date of access: 15.04.2025).

26. Jha N. N., Manwani P. Self-healing payment systems via AI-driven anomaly recovery: a zero-downtime framework for secure and reliable transactions. International Journal of Computer Engineering & Technology. 2025. Vol. 16, № 2. P. 200–212. DOI: https://doi.org/10.34218/IJCET_16_02_014 (date of access: 15.04.2025).

27. Mishra R. K., Raj G. Self-Healing AI: An Autonomous Deep Learning Approach for Software Error Correction. International Journal of Advanced Research in Computer Science and Engineering. 2025. Vol. 1, № 1. P. 28–34. URL: https://ijarcse.org/index.php/ijarcse/article/download/20/9 (date of access: 15.04.2025).

28. Microsoft. Application Insights documentation. Microsoft Azure. (n.d.). URL: https://learn.microsoft.com/en-us/azure/azure-monitor/app/app-insights-overview (date of access: 15.04.2025).

29. Lynch J., Joshi D. A. Netflix's self-healing architecture for distributed databases: automatic failure detection and recovery. Proceedings of the ACM Symposium on Cloud Computing. 2023. P. 143–157. DOI: https://doi.org/10.1145/3573728.3573739 (date of access: 15.04.2025).

30. Naqvi M. A., Malik S., Astekin M., Moonen L. On Evaluating Self-Adaptive and Self-Healing Systems using Chaos Engineering. *2022 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS).* CA, USA, 2022. P. 1–10. DOI: https://doi.org/10.1109/ACSOS55765.2022.00018 (date of access: 15.04.2025).

31. Tan Y. J., Susanto G. J., Ali H. P. A., Tee B. C. K. Progress and roadmap for intelligent self-healing materials in autonomous robotics. *Advanced Materials.* 2021.

Vol. 33, № 19. Article ID 2002800. DOI: https://doi.org/10.1002/adma.202002800 (date of access: 15.04.2025).

32. Garousi V., Joy N., Keleş A. B., Değirmenci S., Özdemir E., Zarringhalami R. AI-powered test automation tools: A systematic review and empirical evaluation. *arXiv preprint*, 2024. arXiv:2409.00411. DOI: https://doi.org/10.48550/arXiv.2409.00411 (date of access: 15.04.2025).

33. Awad M. H., Qutqut A., Ahmed F., Al-Haj F., Almasalha F. Artificial Intelligence Role in Software Automation Testing. *2024 International Conference on Decision Aid Sciences and Applications (DASA)*. Manama, Bahrain, 2024. P. 1–6. DOI: https://doi.org/10.1109/DASA63652.2024.10836630 (date of access: 15.04.2025).

34. P. K., Vinod S. Performance Characterization Model For A Cloud Control Plane. *2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)*. Bangalore, India, 2024. P. 696–701. DOI: https://doi.org/10.1109/ICWITE59797.2024.10502895 (date of access: 15.04.2025).

35. Tkachenko O., Chechet A., Chernykh M., Bunas S., Jatkiewicz P. Scalable Front-End Architecture: Building for Growth and Sustainability. *Informatica*. 2025. Vol. 49, № 1. URL: https://informatica.si/index.php/informatica/article/view/6304 (date of access: 15.04.2025).

36. Musunuri H. Intelligent UI's: Revolutionizing Financial Transaction Systems Through AI and Event-Driven Architecture. *IJSAT – International Journal on Science and Technology*. 2024. Vol. 16, № 2. DOI: https://doi.org/10.71097/IJSAT.v16.i2.3106 (date of access: 15.04.2025).

37. Thota S. R., Arora S. Neurosymbolic AI for Explainable Recommendations in Frontend UI Design – Bridging the Gap between Data-Driven and Rule-Based Approaches. *International Research Journal of Engineering and Technology*. 2024. Vol. 11, № 5. URL: https://www.researchgate.net/profile/Sunil-Raj-Thota/publication/380598618 (date of access: 15.04.2025).

38. Chen W., Cao H., Blanc X. An Improving Approach for DOM-Based Web Test Suite Repair. In: Brambilla M., Chbeir R., Frasincar F., Manolescu I. (eds) *Web Engineering. ICWE 2021*. Lecture Notes in Computer Science, Vol. 12706. Springer,

Cham. DOI: https://doi.org/10.1007/978-3-030-74296-6_29 (date of access: 15.04.2025).

39. Imtiaz J., Iqbal M. Z., Khan M. U. An automated model-based approach to repair test suites of evolving web applications. *Journal of Systems and Software*. 2021. Vol. 171. Article 110841. DOI: https://doi.org/10.1016/j.jss.2020.110841 (date of access: 15.04.2025).

40. Brisset S., Rouvoy R., Seinturier L., Pawlak R. Leveraging flexible tree matching to repair broken locators in web automation scripts. *Information and Software Technology*. 2022. Vol. 144. Article 106754. DOI: https://doi.org/10.1016/j.infsof.2021.106754 (date of access: 15.04.2025).

41. Górski J., Kamiński M. Towards automation of IT systems repairs. *Software Quality Journal*. 2018. Vol. 26. P. 67–96. DOI: https://doi.org/10.1007/s11219-016-9335-5 (date of access: 15.04.2025).

42. Stocco A., Yandrapally R., Mesbah A. Visual web test repair. In: *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 2018. P. 503–514. DOI: https://doi.org/10.1145/3236024.3236063 (date of access: 15.04.2025)

Abstract. *This research focuses on AI-driven methods for building self-healing front-end systems that autonomously detect and recover from UI anomalies. A systematic literature review of 40 peer-reviewed publications was conducted following PRISMA guidelines. Key models such as autoencoders, GANs, reinforcement learning, and CNNs were identified. The study revealed critical limitations including lack of standardization, weak generalizability, and limited explainability. Future directions include the development of edge-capable, lightweight, and transparent recovery strategies to improve interface robustness and adaptability.*

Key words: *interface resilience, UI self-repair, user telemetry, SPA architecture, interpretable AI, reinforcement learning, runtime patching.*

The article has been submitted:

25.04.2025

CONTENTS

Industrial engineering. Management engineering

International periodic scientific journal

MODERN ENGINEERING AND INNOVATIVE TECHNOLOGIES

Heutiges Ingenieurwesen und innovative Technologien

Indexed in
INDEXCOPERNICUS
high impact factor (ICV: 70.62)

Issue №38
Part 1
April 2025

Development of the original layout - Sergeieva&Co

Signed: April 30, 2025

Sergeieva&Co
Lußstr. 13
76227 Karlsruhe
e-mail: editor@moderntechno.de
site: www.moderntechno.de

Articles published in the author's edition

ISSN 2567-5273

www.moderntechno.de

e-mail: editor@moderntechno.de