

CURRENT DEVELOPMENTS AND THEMATIC REVIEWS IN ENGINEERING SCIENCES

Editor
Prof. Selmin Ener Ruşen, PhD.

Current Developments and Thematic Reviews in Engineering Sciences

Editor

Prof. Selmin Ener Ruşen, PhD.

Publisher
Platanus Publishing®

Editor in Chief
Prof. Selmin Ener Ruşen, PhD.

Cover & Interior Design
Platanus Publishing®

The First Edition
October, 2026

ISBN
978-625-91203-5-5

©copyright
All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, or any information storage or retrieval system, without permission from the publisher.

Platanus Publishing®
Address: Natoyolu Cad. Fahri Korutürk Mah. 157/B, 06480, Mamak, Ankara, Turkey.

Phone: +90 312 390 1 118
web: www.platanuspublishing.com
e-mail: platanuskita@gmail.com



Platanus Publishing®

CONTENTS

CHAPTER 1	5
Transformers: Approximation Power and Explaina The Kp index is particularly used for the global-scale assessment of solar ility	
Ramazan Katırcı Hilal Çelik Nusret Gön	
CHAPTER 2	25
Sliding Mode Based Self-Optimization: Accuracy Analysis and Application to Switching Frequency Control of Three-Phase DC/AC Converters	
Ahmet Çakanel	
CHAPTER 3	39
Analysis of The Interaction Between IMF Parameters and Geomagnetic Indices	
Burhaneddin Bilgen Sercan Bülbül Fuat Başçıftçı	
CHAPTER 4	53
Electro-Fenton Process	
Elif Özkul Müftü	
CHAPTER 5	67
Quantum k-Nearest Neighbors for Photocatalytic Hydrogen Production Prediction: Encoding Strategies and Robustness Analysis	
Rezan Bakır	
CHAPTER 6	83
Bell-Shaped Transfer	
Function-Based Binary	
Starfish Optimization Algorithm (SAO_{Bin})	
Emine Baş	
CHAPTER 7	99
Wearable Biosensors for Pesticide Analysis: Design Principles, Current Applications and Future Perspectives	
Esra Armağan	

CHAPTER 8 123

Spiking Neuron Models: Dynamics, Firing Patterns, and Hardware Considerations

Mert Opoz

Tuna Goksu

CHAPTER 9 163

Android Malware Detection via Generative AI

Bekir Berkcan

CHAPTER 10..... 181

Deep Learning-Based Damage Detection and Classification in Historic Masonry Structures Implications for Post-Earthquake Assessment

Bekir Berkcan

CHAPTER 11..... 207

Comparative Evaluation of Polarization and Magnification in Petrographic Thin Section Classification

Rukiye Uzun Arslan

Mualla Toksan

CHAPTER 1

Transformers: Approximation Power and Explaina The Kp index is particularly used for the global-scale assessment of solar ility

Ramazan Katırcı¹

Hilal Çelik²

Nusret Gön³

¹ Prof. Dr., Sivas University of Science and Technology University
ORCID: 0000-0003-2448-011X

² Res. Assist., Sivas University of Science and Technology University
ORCID: 0000-0001-5428-3411

³ Res. Assist. , Sivas University of Science and Technology University
ORCID: 0009-0004-4074-3391

Introduction

Transformers have become a fundamental paradigm in modern machine learning, achieving state-of-the-art performance across a wide variety of domains, such as natural language processing, computer vision, and scientific data analysis. Their success is largely due to the attention mechanism that allows models to learn complex dependencies between elements of an input sequence without using recurrence or convolution. This architecture allows the Transformers to efficiently model long-range interactions and parallelize the processing of information, making them highly scalable and expressive [1].

But these empirical successes do not come with a comprehensive theoretical understanding of Transformers. In particular two fundamental questions continue to drive current research: (i) what is the expressive power of Transformer models, (ii) how can their decision making processes be explained? These questions are important as Transformer-based systems are increasingly used in high-stakes domains such as healthcare, finance, and autonomous systems, where strong predictive performance and reliable interpretability is required [2].

Theoretically, recent progress has started to shed light on the approximation power of Transformers. Several works rely on classical results in neural network theory and demonstrate that Transformer architectures enjoy the universal approximation property (UAP), i.e., they can approximate a large class of functions with arbitrary precision under suitable conditions. Remarkably, it has been demonstrated that a single-layer Transformer with a self-attention mechanism and a feed-forward network is a universal function approximator [3]. Furthermore, recent research indicates that the attention mechanism itself, especially when combined with softmax operations, could be enough to approximate continuous sequence-to-sequence mappings, emphasizing the importance of attention in enhancing model expressiveness [4].

A key insight underlying these results is the interpretation of attention as a mechanism to model relations between inputs. In particular, attention can be seen as computing weighted interactions based on inner products between learned feature representations, which captures similarity or relevance between elements in a sequence well. We relate Transformers to a broader family of relation-based models in which inner products of neural feature maps are universal approximators of relational functions. Their expressive power can therefore be interpreted in the context of Transformers as a consequence of the ability of attention to approximate complex relational structures [5].

These theoretical developments show that the Transformers are very flexible and powerful, but also show a big challenge: interpretability. Transformer models are black-box systems, high-dimensional and with complex internal calculations, and it is hard to understand the contribution of specific inputs to their outputs. Such opacity poses challenges to reliability, fairness, and trust especially in

sensitive applications where erroneous or biased decisions can have serious ramifications [6].

To answer these concerns, the field of Explainable Artificial Intelligence (XAI) has developed a range of methods to interpret Transformer models. They include attention-based visualization techniques, gradient-based attribution methods, perturbation-based analyses and, more recently, causality-driven frameworks. However, the effectiveness of these methods remains a topic of discussion. For example, attention weights are often used as a proxy for explanations, but several studies have shown that they are not always representative of the true importance of the input features and can lead to misleading interpretations [7].

This expressiveness vs interpretability tension hints at a deeper, more fundamental tradeoff. On the one hand, Transformers have the universal approximation ability, which enables them to model highly complex and nonlinear relationships. But this very complexity also makes it inherently difficult for us to extract meaningful and faithful explanations for their predictions. As models become more expressive, with deeper architectures, more attention heads, and richer parameterizations, their internal representations are becoming more abstract and less amenable to human interpretation.

In this chapter, we try to bridge these two views by giving a unified view of approximation theory and explainability in Transformer models. We first provide an overview of the mathematical background of the expressiveness of Transformers, with a special focus on the role of attention mechanisms as universal approximators of relational functions. We then review existing explainability methods, discussing their strengths and weaknesses in the context of transformer architectures. Finally, we discuss the emerging synergy of the approximation power and interpretability, pointing out open challenges and future research directions towards more transparent and trustworthy Transformer-based systems.

Transformer Architecture and Attention Mechanism

The Transformer replaces the recurrent or convolutional structures used in sequence-to-sequence models with a fully attention-based architecture, which allows for parallelization and direct modeling of relationships between tokens in the sequence. The architecture consists of an encoder on the left and a decoder on the right as shown in Figure 1. The encoder maps an input sequence to a sequence of continuous representations through stacked layers of multi-head self-attention and position-wise feed-forward networks. The decoder generates the output sequence autoregressively by attending to the previously generated outputs through masked self-attention, and by attending to the encoder's outputs through cross-attention. Each sub-layer is wrapped with a residual connection followed by layer normalization (Add & Norm), which stabilizes training across depth [1].

The attention mechanism learns the contextual dependencies among the tokens while the feed-forward network applies nonlinear transformations to the token representations independently. This combines the extraction of global context with the transformation of local features [8].

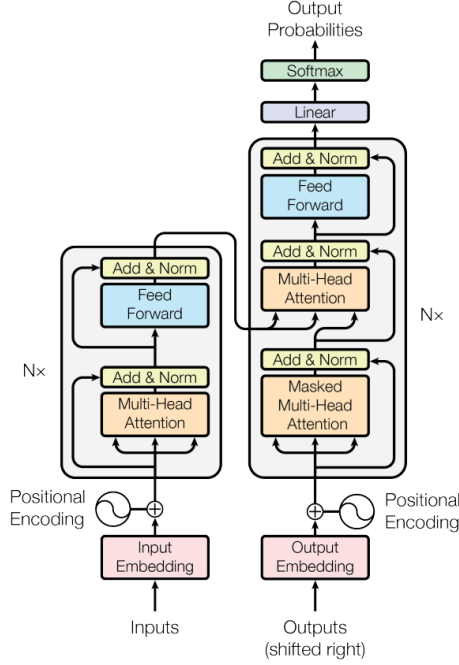


Figure 1. Transformer Architecture [1]

In self-attention, the input tokens are projected into queries, keys and values. Attention weights are computed by scaled dot-product similarities between queries and keys and used to aggregate the value representations. Thus each token can dynamically attend to any other token in the sequence. The decoder in Figure 1 has two attention regimes: masked multi-head attention, where each position can only attend to previous positions (this is needed to preserve the autoregressive generation); and encoder-decoder attention, where the queries come from the decoder side and the keys and values come from the encoder output. Multi-head attention parallelizes this operation across multiple independent heads, each learning to attend to different relational patterns, whose outputs are concatenated and linearly projected. As the mechanism is permutation-equivariant, positional encodings are explicitly added to the input embeddings visible at the bottom of both stacks in Figure 1 to preserve the sequential order [5].

The architecture can be thought of, in theory, as a composition of token-wise transformations (feed-forward layers) and token-mixing operations (attention layers). The division of labor in which feed-forward layers approximate nonlinear memory functions and attention layers retrieve relevant context is the basis for

the model's expressive power and its ability to approximate complex sequence-to-sequence mappings [8]. The final decoder output is projected linearly and passed through a softmax layer, as shown in Figure 1, to obtain a probability distribution over the target vocabulary.

Approximation Power of Transformers

A growing body of work shows that Transformer architectures satisfy the universal approximation property (UAP). Early results proved that standard Transformers with softmax attention and feedforward layers can approximate permutation-equivariant sequence functions, with later studies extending this to sparse, low-rank, and linear attention variants [9], [10]. As summarized in Figure 2, this line of research progressively relaxes the architectural requirements needed to achieve universality.

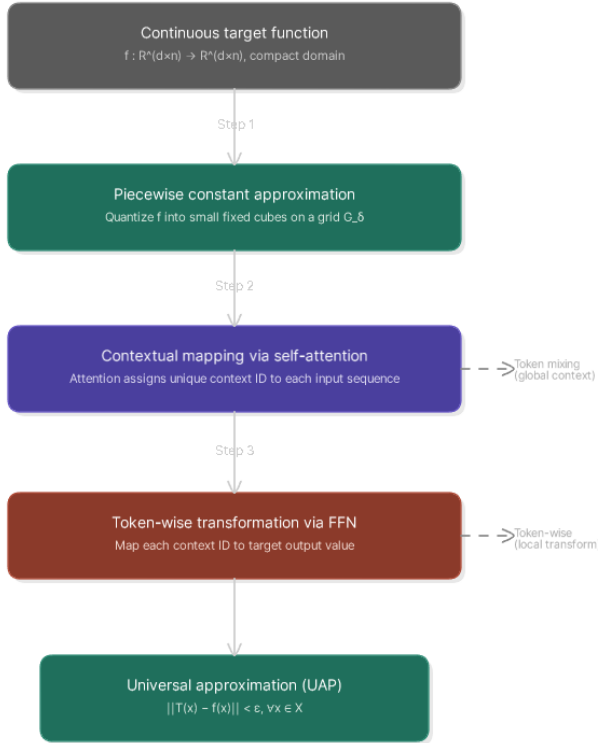


Figure 2. Approximation hierarchy of Transformer variants

More recent work simplifies the picture further: even a single sufficiently wide Transformer layer can achieve universality, though this does not guarantee learnability in practice [3]. Further, two layers of multi-head attention with basic linear transformations are sufficient without feedforward blocks, as softmax attention behaves like a piecewise linear approximator similar to ReLU networks [4]. These minimal architecture results correspond to the lower tiers of Figure 2,

highlighting how each study narrows the architectural footprint required for UAP. Despite these advances, a gap remains between theoretical expressivity and practical efficiency: the existence of approximating parameters does not imply that gradient-based training will find them, nor that the required model size is computationally feasible.

Classical Universal Approximation Theorem

A one hidden layer network with enough neurons can approximate any continuous function on a compact domain to any accuracy, according to the Universal Approximation Theorem (UAT). First, it was shown for sigmoidal activations alone and then for larger classes of activations. This showed that the representational bottleneck of early neural architectures was not an intrinsic property of the function class, but rather due to finite parameter budgets and optimization challenges. Then came depth-based extensions showing that networks of fixed width but increasing depth can also achieve universality, and later work proved that a certain minimum width is necessary and sufficient — results that are relevant to Transformers, but whose parameter-sharing and permutation-equivariance constraints require different proof techniques [3], [9]. Crucially, the UAT is purely existential: it guarantees that approximating parameters exist, but tells us nothing about whether gradient-based optimization can find them, how many parameters are needed, or sample efficiency. This gap between existence and practical learnability motivates the study of approximation rates and architectural constraints underlying the theoretical analysis of Transformer expressiveness [10].

Universal Approximation in Transformers

Extending universal approximation theory to Transformers requires overcoming structural constraints absent in standard feedforward networks: weight sharing across positions and the inherent permutation equivariance of self-attention restrict the representable function classes and necessitate tailored proof strategies. The first rigorous result proved that a Transformer with two attention layers and four-wide feedforward sublayers can approximate any continuous, permutation-equivariant sequence-to-sequence function on a compact domain. The proof proceeds in three steps: approximating the target by piecewise constant functions, using stacked self-attention to distinguish input tokens, and mapping contextual representations to output values via token-wise feedforward layers [9].

Subsequent work confirmed that universality is robust to architectural modifications. Sparse Transformers where each token attends to only $O(n)$ others retain the UAP provided each token attends to itself, the attention graph contains a Hamiltonian path, and global information can propagate across all tokens; low-rank projection variants likewise preserve universality while reducing parameter counts. A minimality result further shows that a single-layer Transformer of sufficient width is already a universal function approximator, confirming that

depth is not a fundamental requirement for expressive power though deeper models achieve the same approximation far more parameter-efficiently [3].

Role of Attention in Function Approximation

Prior universality results were mostly based on feedforward sublayers for tokenwise nonlinear transformations, where attention served mostly as a routing mechanism for context. More recent work challenges this: [4] show that two layers of multi-head softmax attention with simple linear transformations are sufficient for sequence-to-sequence universal approximation, without feedforward sublayers. The key insight is that the softmax operation can be made work as an approximate argmax selector over anchor points in the key-query-value space, thus making attention a piecewise linear approximator of arbitrary continuous functions.

In a complementary view, attention and feedforward layers have different computational roles: the feedforward sublayers approximate the nonlinear memory functions and the readout mapping, while the attention layers retrieve relevant tokens from specific memory locations. This division of labor suggests that the expressivity of Transformers is due to the cooperation of both components, not just attention alone, even though attention-only architectures can be universal in principle. Understanding this interplay is important for both efficient architecture design, and for interpreting the computational mechanisms behind Transformer predictions [8].

Approximation Rates and Model Capacity

Universality results confirm that Transformers can approximate arbitrary continuous functions but do not quantify how efficiently. Explicit approximation rates across three task classes of increasing difficulty show that simple fixed-sparse memory tasks require only a single-layer Transformer, while adaptive or nested memory structures require additional depth, with each layer resolving a distinct level of interaction [8]. However, a gap exists between theoretical universality and practical efficiency Transformers may exhibit poor approximation rates for smooth function classes, requiring excessively large models. For multi-head softmax attention, the approximation error decreases predictably as both the number of attention heads and the sequence length increase, quantifying the width-accuracy trade-off and formally justifying the empirical practice of jointly scaling attention heads and feedforward width [4].

Relation Functions and Attention Mechanisms

A particularly illuminating perspective frames self-attention as a device for computing relation functions mappings that capture how each input element influences the representation of others. For each token pair, the model computes a similarity score via inner products of learned query and key representations and uses these scores to aggregate value vectors, implicitly defining a relational

structure over the sequence. Inner products of neural feature maps have been shown to universally approximate symmetric, positive-definite relation functions, directly connecting Transformer expressiveness to the relational domain and implying that much of the model's representational capacity derives from the relational structure encoded through attention rather than from the feedforward components alone.

Definition of Relation Functions

A relation function assigns a scalar to each pair of inputs (x, y) , capturing similarity, relevance, or dependency a framework that subsumes kernels, metric learning similarity functions, and co-occurrence statistics. Symmetric, positive-definite relation functions are of particular theoretical interest: by Mercer's theorem, any such function admits a spectral decomposition as a weighted sum of eigenfunction products, providing the mathematical foundation for approximating relation functions via inner products of learned neural feature maps.

Inner Product Representation

The inner product of two neural feature maps $\phi(x)$ and $\phi(y)$ — where ϕ is a learnable neural network — defines a class of relation functions parameterized by the network's weights. A central result establishes that this class of inner product relational networks constitutes a universal approximator for symmetric, positive-definite relation functions: for any continuous symmetric kernel and any desired precision, there exists a neural network whose inner product approximates that kernel to arbitrary accuracy. The complexity of the required network is governed by the effective dimensionality of the kernel, defined as the number of eigenvalues needed to represent it up to the target precision [5].

Attention as Relation Approximation

Self-attention implements inner product-based relation approximation directly: the attention weight between each token pair is the softmax of the inner product between learned query and key feature maps, with the resulting weighted aggregation of value vectors integrating relational information globally. Since query and key transformations are learnable neural networks, they can in principle represent any symmetric, positive-definite relation function, giving self-attention an essentially unrestricted capacity for modeling pairwise dependencies.

Foundations of Explainable AI (XAI)

The ubiquitous use of Transformers in high-stakes domains has escalated the need for interpretable AI. Explainable Artificial Intelligence (XAI) refers to the techniques and frameworks developed to render model reasoning interpretable to human understanding a particularly challenging task for Transformers, whose

multi-head attention, residual connections and feedforward networks interact in ways that do not lend themselves to decomposition [2].

XAI methods can be broadly classified into two types: post-hoc methods applied on pre-trained models such as attention visualization, gradient attribution, perturbation analysis, and layer-wise relevance propagation and inherently explainable models designed to generate interpretable representations from the beginning. There are trade-offs between fidelity, computational cost, and generality in each category.

A major open debate is whether attention weights are faithful explanations. Its advocates say they are an architecture-driven reflection of token importance, while its detractors say that attention distributions can be altered without affecting predictions [11]. Work since then suggests the answer is task- and architecture-dependent [7], motivating the development of more rigorous, mathematically grounded explainability frameworks.

What is Explainability?

Explainability refers to the capacity of a model or auxiliary method to provide human-understandable accounts of how inputs are transformed into outputs. It is often distinguished from interpretability: interpretability denotes inherent comprehensibility of a model's internal mechanisms, while explainability refers more broadly to the availability of post-hoc descriptions that approximate those mechanisms. In both cases the goal is to enable domain experts, auditors, and affected stakeholders to understand, verify, and trust model predictions. As illustrated in Figure 3, these three notions transparency, explainability, and interpretability sit under the broader XAI umbrella, each capturing a distinct aspect of model comprehensibility.

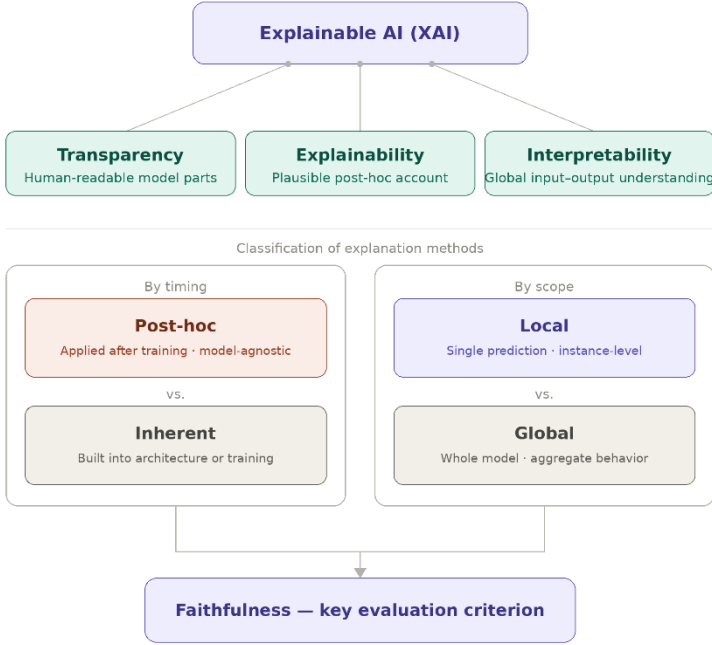


Figure 3. An overview of the XAI framework

Explainability methods are further classified along two key dimensions, shown in Figure 3. The first is timing: ante-hoc (inherent) methods design the model to be intrinsically interpretable from the outset, while post-hoc methods apply explanation procedures after training. The second is scope: local methods explain individual predictions, while global methods characterize model behavior across the full input domain. A third axis model dependence distinguishes model-agnostic methods, which apply to any predictor, from model-specific methods that exploit structural properties such as attention weights or gradients [7]. Across all three axes, faithfulness serves as the central evaluation criterion: a method is faithful to the extent that its explanations accurately reflect the model's true internal reasoning rather than merely producing a plausible narrative. For Transformers, all three axes are relevant, as the architecture offers unique explanatory opportunities through attention while presenting distinctive challenges from its depth and nonlinearity.

Challenges in Deep Learning Models

There are fundamental explainability challenges with deep learning models that are not present with simpler models such as linear classifiers or decision trees. Internal representations are high-dimensional, and it is challenging to identify the directions in which the model encodes semantically meaningful information. The model consists of multiple nonlinear transformations, which

lead to complex and counterintuitive propagation of small changes in the input, thus making gradient-based explanations local and possibly misleading [2].

Transformers add further complications. Each head and layer in multi-head attention leads to a different set of weights, resulting in partially overlapping explanatory signals that should be aggregated in a principled manner. Residual connections make attribution more difficult by propagating information from early layers to later ones, making it hard to attribute the contribution of individual attention operations. This results in partial or skewed explanations when naive strategies such as reading off last-layer attention weights are employed, hence the need for more advanced attribution techniques [12].

Explainability Methods for Transformers

Attention-Based Explanations

Attention-based explanations are the most architecturally natural approach to interpreting Transformer predictions, as token-to-token relevance scores are computed during the forward pass and require no additional computation. The most direct approach visualizes raw attention weights, treating high-weight positions as the tokens most responsible for a given prediction a technique widely applied across machine translation, question answering, and sentiment classification.

More sophisticated variants propagate attention signals across layers rather than reading them off at a single point. Attention Rollout recursively multiplies attention matrices across layers, adding the identity matrix at each step to account for residual connections; Attention Flow models propagation as a maximum-flow problem on the attention graph, identifying the highest-information paths from input to output [12]. Both methods improve on raw visualization but share a recognized limitation: they assume information flows exclusively through attention matrices and ignore the contribution of the feedforward sublayers [2].

Gradient-Based Methods

Gradient-based methods attribute predictions by computing the sensitivity of the output with respect to each input feature. Saliency mapping uses raw gradient magnitudes; Gradient $\times$ Input scales these by the activation value to account for the baseline level; Guided Backpropagation restricts flow to positive activations to suppress noise. Integrated Gradients integrates the gradient along a straight path from a baseline input to the actual input, satisfying the completeness axiom that attributions sum to the output difference between the two.

Applied to Transformers, gradients can be computed with respect to any differentiable model component, making these methods architecturally general. Their key limitation is locality: gradients capture only first-order sensitivity at a single input point and may fail to reflect global model behavior when the loss

landscape is flat or activations saturate. [7] show mathematically that gradient methods capture aspects of model structure inaccessible from attention weights alone, though the gap between the two grows with model complexity.

Perturbation-Based Methods

Perturbation-based methods compute feature importance by iteratively perturbing the input and measuring the change in model output. These methods treat the model as a black box and do not need access to internal quantities such as gradients or attention weights. LIME fits a locally linear surrogate to the predictions on perturbed inputs from the neighborhood of the input of interest, thus approximating the local decision boundary of the model. This extends SHAP to a global setting, by calculating Shapley values from cooperative game theory, where each feature is assigned an expected marginal contribution averaged over all possible feature coalitions. Both have been used on Transformers for NLP as well as computer vision tasks [12].

Transformer Input Sampling (TIS). TIS for vision Transformers aggregates the model output over random subsets of input patches to generate patch-level importance scores aligned with the causal contribution to the prediction, regardless of the gradient or attention signals. TIS explanations are more faithful to the model than attention-based methods, but suffer from heavier computational costs due to repeated forward passes.

Layer-wise Relevance Propagation (LRP)

LRP decomposes the model output into per-neuron relevance contributions by propagating relevance backwards through the network under a conservation constraint: the total relevance at each layer equals that of the layer above, so the full output score is redistributed to the input without loss. Each neuron's relevance is computed proportionally to its weighted activation relative to the total input received by the neuron above. Originally developed for convolutional networks, LRP required non-trivial adaptation to handle Transformers matrix multiplications, skip connections, and softmax operations [2].

The Transformer-specific LRP-Rollout variant addresses this by computing per-block relevance matrices as the Hadamard product of each attention matrix and its output gradient, averaged across heads and augmented with the identity matrix for the residual connection. Multiplying these matrices across all layers yields a global, class-discriminative attribution map analogous to Attention Rollout but with gradient weighting. This combination outperforms both pure rollout and gradient-only methods in faithfulness evaluations on image classification and natural language inference; a conservative variant that enforces the conservation property also at LayerNorm further improves attribution accuracy [2], [6], [12], [13].

Causality-Based Approaches

Causality-based methods move beyond statistical association by identifying input features whose manipulation would actually change the model's prediction, drawing on potential outcomes theory and interventional do-calculus. CAT-XPLAIN integrates this principle directly into Transformer training: a learned selector head outputs a distribution over input patches and is jointly optimized to identify the minimal causally sufficient patch subset for the classification decision, with a selection loss defined as cross-entropy over the unselected patches. The resulting model simultaneously classifies and explains, eliminating the need for a separately trained explainer and improving both faithfulness and computational efficiency [14].

Limitations of Explainability in Transformers

Attention is Not Explanation

The idea that attention weights provide valid explanations has been widely debated. Empirical work showed that attention often correlates weakly with gradient-based importance measures, and that very different attention patterns can produce the same prediction, questioning their reliability as explanations [11].

This view was challenged by later studies arguing that such alternative attention patterns may be unrealistic under the model's training dynamics, suggesting that attention still carries meaningful signal. Further theoretical analysis indicates that attention alone is insufficient, while post-hoc methods capture additional aspects of the decision process. Overall, the validity of attention as explanation remains context-dependent and unresolved [15].

Local vs Global Explanations

Most Transformer explanation methods are local: they identify which input tokens matter for a single prediction, but do not describe the model's overall behavior. While useful for debugging and individual decision auditing, local explanations can miss systematic patterns, including cases where models rely on spurious but locally predictive features "explanation gaming".

Global methods aim to address this by approximating model behavior across the full input space, either through surrogate models or by aggregating local attributions over datasets. However, surrogates introduce approximation error, and aggregation can hide important variability across inputs. As a result, connecting local explanations to faithful global understanding remains an open challenge, especially for high-dimensional sequence models like Transformers.

Instability of Explanations

A key requirement for explanation methods is stability: small input changes should not lead to large shifts in explanations. In practice, many gradient- and attention-based methods are sensitive to input perturbations, initialization, and training variations. SmoothGrad reduces this by averaging gradients over noisy inputs, but cannot fully resolve instability in flat or saturated regions of the model.

In attention-based explanations, instability appears across heads, layers, and aggregation choices. Different strategies (e.g., averaging vs. max) can yield conflicting importance scores, reflecting how multi-head attention distributes information across parallel, weakly coupled components. Improving robustness requires more principled aggregation methods and stronger theoretical guarantees [7].

Trade-off Between Approximation and Explainability

Expressiveness vs Interpretability

A core challenge in machine learning is balancing expressiveness and interpretability, a trade-off visualized in Figure 4. As shown along the curve, simpler models such as linear regression and decision trees sit at the high-interpretability, low-expressiveness end of the spectrum: their internal mechanisms are directly readable, but they lack the capacity to capture nonlinear and long-range patterns. Moving rightward along the curve, shallow MLPs, BERT/GPT-style models, and finally deep Transformers occupy progressively lower positions on the interpretability axis. Theoretically, as model expressiveness increases, so does the number of possible functions consistent with the data, making it harder to determine what the model has actually learned [10].

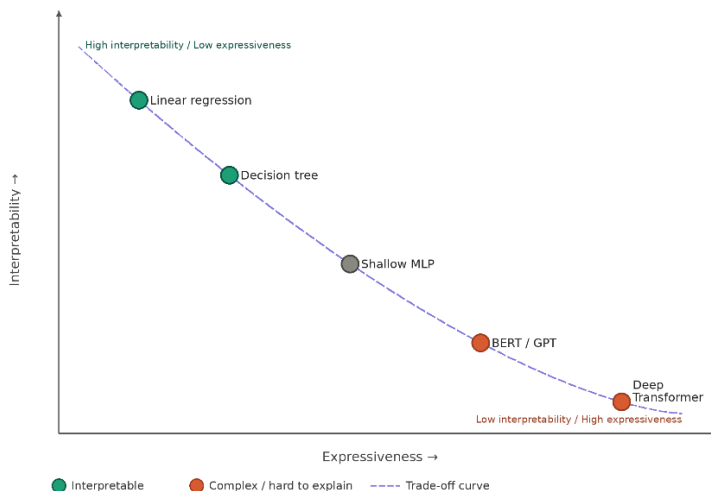


Figure 4. The expressiveness–interpretability trade-off across model families

For Transformers, this issue is amplified by their universal approximation capability: a model that can represent almost any function is inherently harder to interpret, placing deep Transformers at the far right of Figure 4. Thus, their strength in flexibility also creates a fundamental limitation in explainability. As indicated by the mitigation zone in Figure 4, recent approaches aim to partially recover interpretability by constraining the model through architecture, regularization, or domain knowledge, while preserving performance.

Complexity vs Transparency

Model complexity is generally inversely related to transparency, as illustrated in Figure 5. While a single attention head produces a relatively readable query-key similarity map, stacking multiple heads and layers progressively entangles parameter contributions through nonlinear compositions, obscuring individual roles. As shown in Figure 5, this progression from single-head attention to multi-head attention, stacked layers, and finally deep Transformer stacks systematically reduces transparency: shared query, key, and value projections operate across all positions, making attention behavior input-dependent and emergent rather than directly interpretable [8].

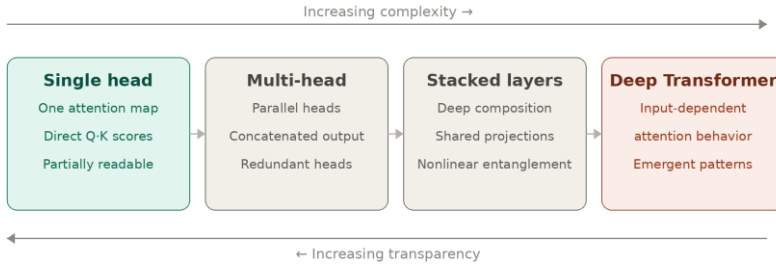


Figure 5. The inverse relationship between model complexity and transparency in Transformer architectures

Efforts to improve transparency, summarized in the mitigation zone of Figure 5, typically rely on post-hoc simplification or structured design. Techniques such as head pruning show that many attention heads are redundant, yet the remaining heads do not necessarily become more interpretable, as their behavior may reflect compensatory dynamics rather than clear functional roles [2].

Theoretical Perspective

In theory, the trade-off between approximation power and explainability can be captured by the complexity of the function class. VC dimension or Rademacher complexity: The higher the capacity, the larger the range of functions the model can represent. This increases flexibility but also leads to an identifiability problem, where many different functions can fit the same finite dataset, and it is inherently difficult to determine which one the model has learned.

In the case of Transformers, the attention mechanism introduces some structure by its dependence on inner products. Attention can learn arbitrary pairwise dependencies, because these inner products are universal approximators for relational functions. But this same flexibility results in non-identifiability: different underlying parametrizations of queries and keys can lead to the same attention weights. Therefore, attention weights alone cannot explain the model behavior and faithful interpretation requires understanding the full computational structure of the model [5].

Emerging Directions and Research Opportunities

Inherently Explainable Transformers

A key approach to resolving the tension between expressiveness and interpretability is to design Transformers that are explainable by construction, rather than relying on post-hoc methods. Such models generate explanations as part of their forward pass, ensuring faithfulness to their actual decision process. For example, CAT-XPLAIN integrates causal feature selection into training, enabling the model to identify the minimal set of input elements responsible for its predictions [14].

More generally, interpretability can be improved by imposing structural constraints on attention. Sparse attention limits each token's focus, making patterns easier to interpret, while relational architectures explicitly model objects and their interactions, offering a more modular and inspectable structure. The main challenge is maintaining strong approximation power while introducing these constraints, as preserving both expressiveness and interpretability remains theoretically difficult [5], [10].

Efficient Attention Mechanisms

The quadratic cost of the standard softmax attention has motivated the development of more efficient mechanisms that attempt to preserve expressiveness. For example, the Performer architecture approximates the softmax attention with the random feature methods (FAVOR+), which allows a linear time complexity without explicitly computing the full attention matrix. Importantly, it approximates standard attention in appropriate regimes, i.e., efficiency does not come at the expense of core expressiveness [16].

Other variants, such as sparse and linear attention, could improve interpretability. Sparse attention attends to a small subset of tokens, resulting in more interpretable attention patterns, while kernel-based linear attention can be more amenable to theoretical analysis. In general, work in progress is a trade-off between efficiency, approximation power, and interpretability in a single framework [10].

Hybrid Explainability Frameworks

Given the limits of any single method, hybrid explainability frameworks combine complementary techniques for more reliable insights. For example, LRP-Rollout merges Attention Rollout with gradient-based relevance, producing explanations that reflect both the model's structure and class-specific signals. Similarly, combining perturbation-based methods like SHAP with gradient-based approaches such as Integrated Gradients can improve coverage and attribution quality.

A promising direction is linking theoretical insights with practical methods. Viewing attention as relational function approximation suggests focusing on pairwise dependencies rather than individual tokens. When combined with causal approaches, this could yield explanations that are mathematically grounded, causally faithful, and more interpretable. Advancing this area will require closer integration between theory and applied explainability research [5], [7], [14].

Conclusion

This chapter has explored Transformer architectures from two complementary theoretical angles: approximation power and explainability. We have demonstrated that there is a fundamental tension at the heart of modern deep

learning. On the side of expressiveness a clear picture has emerged: Transformers are universal approximators of functions. This was shown for permutation-equivariant sequence-to-sequence functions by the foundational universality result, and subsequent works have incrementally improved the result. Softmax attention alone, without feedforward sublayers, is sufficient for continuous sequence-to-sequence approximation, and approximation error decreases predictably with the number of attention heads and sequence length. Universality follows from a sufficiently wide single-layer Transformer. Grounding this expressiveness in the theory of kernels, the relational perspective shows that inner products of learned feature maps universally approximate symmetric, positive-definite relation functions, which is precisely what self-attention computes [3], [4], [5], [9].

The very expressivity that gives Transformers their power also makes them difficult to interpret. Universal approximation implies that there is no finite human-understandable description that can fully characterize what a trained model has learned. The non-identifiability of the underlying relation function from attention weights implies that faithful explanations cannot be read off from the most architecturally accessible quantity. The XAI methods reviewed in this chapter attention visualization, gradient attribution, perturbation analysis, LRP, and causality-based approaches, each provide insight into a different aspect of model behavior, but none achieve universal faithfulness. However, whether attention is an explanation remains an open question, and the instability reported across attribution methods should prompt caution against over-reliance on a single approach [2], [5], [7], [11].

The most promising directions going forward are at the intersection of the two perspectives. Inherently explainable architectures show that faithfulness is achievable and that performance doesn't have to be sacrificed for explainability in the training objective. Efficient attention mechanisms show that the universality of softmax attention can be preserved under linear-complexity approximations, possibly also mitigating the opacity of dense attention patterns. Hybrid explainability frameworks that combine gradient information, relevance propagation and causal reasoning provide a route to explanations that are mathematically grounded, computationally tractable and human-comprehensible at the same time. The intersection of approximation theory and interpretability research remains an open and important challenge, and progress on this frontier will be necessary to deploy Transformer-based systems responsibly in the high-stakes applications that increasingly depend on them [7], [14], [16].

References

- [1] A. Vaswani et al., "Attention Is All You Need," Aug. 2017, [Online]. Available: <http://arxiv.org/abs/1706.03762>
- [2] P. Fantozzi and M. Naldi, "The Explainability of Transformers: Current Status and Directions," *Computers*, vol. 13, no. 4, Apr. 2024, doi: 10.3390/computers13040092.
- [3] E. Gumaan, "Universal Approximation Theorem for a Single-Layer Transformer," Jul. 2025, [Online]. Available: <http://arxiv.org/abs/2507.10581>
- [4] J. Y.-C. Hu, H. Liu, H.-Y. Chen, W. Wu, and H. Liu, "Universal Approximation with Softmax Attention," Dec. 2025, [Online]. Available: <http://arxiv.org/abs/2504.15956>
- [5] A. Altabaa and J. Lafferty, "Approximation of relation functions and attention mechanisms," Jun. 2024, [Online]. Available: <http://arxiv.org/abs/2402.08856>
- [6] A. Ali, T. Schnake, O. Eberle, G. Montavon, K.-R. Müller, and L. Wolf, "XAI for Transformers: Better Explanations through Conservative Propagation," Jun. 2022, [Online]. Available: <http://arxiv.org/abs/2202.07304>
- [7] G. Lopardo, F. Precioso, and D. Garreau, "Attention Meets Post-hoc Interpretability: A Mathematical Perspective," Jun. 2024, [Online]. Available: <http://arxiv.org/abs/2402.03485>
- [8] M. Wang and W. E, "Understanding the Expressive Power and Mechanisms of Transformer for Sequence Modeling," Oct. 2024, [Online]. Available: <http://arxiv.org/abs/2402.00522>
- [9] C. Yun, S. Bhojanapalli, A. Singh Rawat, S. J. Reddi, and S. Kumar, "Are Transformers Universal Approximators of Sequence-to-Sequence Functions?," 2020.
- [10] A. Abbasi and M. Hooshmand, "On the Universality of Transformer Architectures; How Much Attention Is Enough?," Dec. 2025, [Online]. Available: <http://arxiv.org/abs/2512.18445>
- [11] S. Jain and B. C. Wallace, "Attention is not Explanation," May 2019, [Online]. Available: <http://arxiv.org/abs/1902.10186>
- [12] R. Kashefi, L. Barekatin, M. Sabokrou, and F. Aghaeipoor, "Explainability of Vision Transformers: A Comprehensive Review and New Perspectives," Nov. 2026, doi: 10.1007/s11042-026-21313-7.

- [13] H. Chefer, S. Gur, and L. Wolf, "Generic Attention-model Explainability for Interpreting Bi-Modal and Encoder-Decoder Transformers." [Online]. Available: <https://github.com/hila-chefer/>
- [14] S. Khanal, B. Brodie, X. Xing, A.-L. Lin, and N. Jacobs, "Causality for Inherently Explainable Transformers: CAT-XPLAIN," Jun. 2022, [Online]. Available: <http://arxiv.org/abs/2206.14841>
- [15] S. Wiegrefe and Y. Pinter, "Attention is not not Explanation," Sep. 2019, [Online]. Available: <http://arxiv.org/abs/1908.04626>
- [16] K. Choromanski et al., "Rethinking Attention with Performers," Nov. 2022, [Online]. Available: <http://arxiv.org/abs/2009.14794>

CHAPTER 2

**Sliding Mode Based Self-Optimization:
Accuracy Analysis and Application to
Switching Frequency Control of
Three-Phase DC/AC Converters**

Ahmet Çakanel¹

¹ Dr. Öğr. Üyesi, Kırklareli Üniversitesi, ORCID: 0000-0003-2988-325X

1. Introduction

The objective of a control system is to maintain the desired output of a plant with the best attainable efficiency. In many modern technologies this performance is enhanced by minimizing or maximizing selected state variables, and such optimization problems must frequently be handled under conditions in which the extremum value, the local plant gain, and the required inputs are unknown. Static optimization consists in finding a combination of input parameters such that the scalar output of the plant attains an extremal value (Utkin, 1992). Problems of this class arise in a wide range of engineering applications, including power converters, combustion and gasification processes, and optical tracking systems (Cakanel, 2017).

The conventional approach to self-optimization is based on the calculation or evaluation of the gradient of the plant characteristic $y = f(x)$. In gradient-based schemes, the derivative of the state is chosen proportional to the gradient of the function,

$$\dot{y} = -\gamma f', \quad y = f(x), \quad (1)$$

where γ determines the convergence rate. Numerous methods have been introduced to acquire the gradient or information about it: direct differentiation of input and output signals, incrementation by one step, or injection of probe signals. In the Russian literature, probe signals are used within the so-called synchronous detection scheme, in which the plant input is perturbed by $\sin(\omega t)$, the output is multiplied by the same harmonic, and low-pass filtering recovers a signal proportional to $\gamma f'$ (Korovin & Utkin, 1974). Sinusoidal perturbation is likewise the core of classical extremum-seeking control, valued for its fast adaptation (Krstić & Wang, 2000; Ariyur & Krstić, 2003), where the plant is locally represented as

$$f(x) = f(x^*) + \frac{f''}{2} (x - x^*)^2. \quad (2)$$

In all such schemes the gradient must be evaluated, directly or indirectly, before the control can be formed. The additional components required for this evaluation slow down the search process, and the oscillations they induce around the extremum cause power losses and diluted accuracy.

Sliding mode control (SMC) offers an alternative that avoids gradient evaluation altogether. SMC theory was developed for plants operating under uncertainty and exhibits high accuracy, robustness, and low sensitivity to parameter variations and disturbances (Utkin, 1977; Utkin, 1992). However, SMC cannot be applied to the self-optimization problem directly, because the gradient, which takes part in the local gain, changes its sign at the extremum, making conventional tracking impossible. The sliding mode self-optimization procedure therefore implies that the plant output tracks a monotonously

decreasing (or increasing) reference input; as a result, the output reaches the extremum without any gradient information (Utkin & Cakanel, 2016).

This chapter presents the design principle of sliding mode based self-optimization together with the analysis of system behavior in the vicinity of the extremum, where the local gain tends to zero, the sliding mode disappears, and high frequency oscillations known as chattering arise. The main theoretical result is that the accuracy of the search process depends neither on the rate of the reference input nor on the sliding mode existence conditions, but is determined only by the width of the hysteresis loop of the switching element. The methodology is then applied to a practical power electronics problem: minimization of the total switching frequency of a three-phase DC/AC inverter feeding an induction motor, which directly reduces the heat losses of the converter (Cakanel & Utkin, 2016). All simulation results presented in the chapter were regenerated numerically by the author.

The chapter is organized as follows. Section 2 introduces the search algorithm, analyses its stability and accuracy after the sliding mode disappears, and compares the scheme with the time-varying search speed modification. Section 3 formulates the inverter switching frequency control problem, derives the sliding mode control algorithm, and presents simulation results. Section 4 concludes the chapter.

2. Sliding Mode Based Self-Optimization

2.1. Search Algorithm

Self-optimization means that the input x should be found such that the function $y = f(x)$ reaches its minimum value. The tracking system is designed such that $\varepsilon(t)$, the difference between the monotonously decreasing reference input $g(t)$ and the output of the plant $y(t)$, vanishes:

$$\varepsilon = g(t) - y(t), \quad \dot{g} = -\rho, \quad \dot{x} = u, \quad \rho > 0. \quad (3)$$

The output tracks the reference input if a sliding mode exists, which requires that $\dot{\varepsilon}$ and ε have opposite signs. For a relay control of magnitude A this leads to the sliding mode existence condition

$$A |f'| > \rho. \quad (4)$$

If the sign of f' were known, the sliding mode could be enforced with $u = A \text{sign}(\varepsilon f')$. Since f' is unknown and changes sign at the extremum, a conventional relay results in positive feedback on one branch of the extremal curve. The conventional relay is therefore replaced by a discontinuous control with two points of discontinuity,

$$u = A \text{sign}(s_1 s_2), \quad s_1 = \varepsilon, \quad s_2 = \varepsilon + \delta, \quad (5)$$

where δ and A are positive constants. Let the initial conditions be

$$s_1(t) s_2(t) < 0, \quad A \left| \frac{df}{dx} \right| > \rho. \quad (6)$$

Under these initial conditions, either s_1 or s_2 satisfies the inequality $s\dot{s} < 0$ in the vicinity of $s = 0$; hence a sliding mode is bound to start on one of the two surfaces, and with the decaying reference input the minimum point is eventually reached (Utkin, 1992). Which of the two surfaces carries the sliding mode is decided by the sign of the unknown gradient, and this is precisely why the algorithm requires no gradient evaluation.

To create and restore the initial conditions in Equation 6, an additional discontinuous function v with two hysteresis loops is added to the reference input dynamics. Since, by virtue of Equations 3–5,

$$\dot{s}_1 = \dot{s}_2 = \dot{\varepsilon} = -\rho + \frac{df}{dx} A \operatorname{sign}(s_1 s_2) \mp v, \quad (7)$$

a sufficiently large magnitude M makes s_1 and s_2 acquire opposite signs after a finite time interval whenever they coincide in sign. The function v is defined as

$$v = \begin{cases} -M, & (s_1 - \Delta) > 0, \quad s_2 > 0, \\ 0, & (s_1 + \Delta)(s_2 - \Delta) < 0, \\ +M, & s_1 < 0, \quad (s_2 + \Delta) < 0, \end{cases} \quad (8)$$

where 2Δ is the width of the hysteresis loop and $M \gg \rho$ is a positive constant. In a real sliding mode the functions s_1 and s_2 oscillate with small but finite amplitude; the width of the hysteresis loop must exceed this amplitude to avoid spurious activation of v . The signal v both ensures the initial conditions and stabilizes the motion of the system in the vicinity of the extremum: whenever the describing point leaves the hysteresis band, it is tossed back into the region $(s_1 + \Delta)(s_2 - \Delta) < 0$, and the sliding mode takes the plant output into a certain domain containing the extremum point.

Figure 1 shows the simulation of the search process for a quadratic plant $f(x) = (x - 2)^2 + 1$ with the parameters listed in Table 1. The output tracks the linearly decreasing reference input in the sliding mode and reaches the minimum value $y_{min} = 1$; afterwards, the motion is confined to a small vicinity of the extremum by the hysteresis function v .

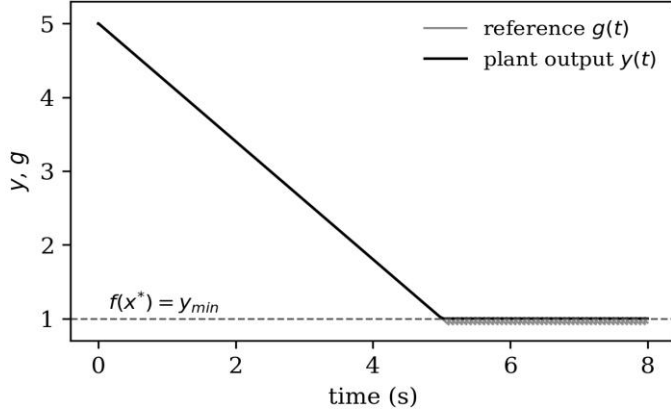


Figure 1. Sliding mode self-optimization: plant output $y(t)$ tracking the monotonously decreasing reference $g(t)$ until the minimum $f(x^*)$ is reached.

Table 1. Parameters of the self-optimization simulation

Parameter	Symbol	Value
Relay amplitude	A	2
Rate of reference input	ρ	0.8
Distance between switching surfaces	δ	0.05
Half-width of hysteresis loop	Δ	0.03
Restoring signal magnitude	M	200

2.2. Accuracy after the Sliding Mode Disappears

In the course of approaching the minimum point, f' tends to zero for differentiable functions and the condition in Equation 4 ceases to hold; the sliding mode disappears. In the majority of the published work, the point where the sliding mode disappears was accepted as the measure of accuracy, and if this level was too low, the rate of the reference input was decreased, slowing down the search. The analysis below shows that the output continues to decay in average after the sliding mode disappears, and that the accuracy is determined only by the width of the hysteresis loop (Utkin & Cakanel, 2016).

When the sliding mode disappears, the term ρ dominates the control and the describing point moves toward $(s_2 - \Delta)$, activating the function v , which instantaneously restores the condition $s_2 = +\Delta$; oscillations then repeat. During each oscillation the describing point travels $0 \leq s_2 \leq \Delta$ in a time τ_1 and $-\Delta \leq s_2 \leq 0$ in a time τ_2 , where

$$\Delta = \int_0^{\tau_1} \left(\rho - A \frac{df}{dx} \right) dt, \quad \Delta = \int_0^{\tau_2} \left(\rho + A \frac{df}{dx} \right) dt. \quad (9)$$

For a locally constant gradient the interval durations follow as

$$\tau_1 = \frac{\Delta}{\rho - A \frac{df}{dx}}, \quad \tau_2 = \frac{\Delta}{\rho + A \frac{df}{dx}}, \quad (10)$$

and the corresponding output changes are $\Delta y_1 = \Delta - \rho \tau_1$ and $\Delta y_2 = \Delta - \rho \tau_2$. Denote by $f'_R > 0$ and $f'_L < 0$ the gradient values on the right and left sides of the minimum. Considering one oscillation that starts at $s_2 = +\Delta$, the input excursions on the two sides of the extremum are

$$\Delta x_1 = \frac{\Delta A}{|\rho - f'_R A|}, \quad \Delta x_2 = \frac{\Delta A}{|\rho + f'_L A|}. \quad (11)$$

Since $\Delta x_2 < \Delta x_1$, the input moves toward the extremum, and tracing the subsequent partial excursions shows that they form a decreasing sequence; consequently the average value of the output continues decaying after the sliding mode disappears. In the area of the extremum point, the difference between the minimum value of the minimized function and the real output is of order Δ . Thus Δ , the hysteresis width of the switching element, is the measure of the search process accuracy: the accuracy depends neither on the rate ρ of the reference input nor on the sliding mode existence condition. Figure 2 presents the steady-state behavior in the vicinity of the extremum; the output oscillates within a band of order Δ above the true minimum, confirming the analysis.

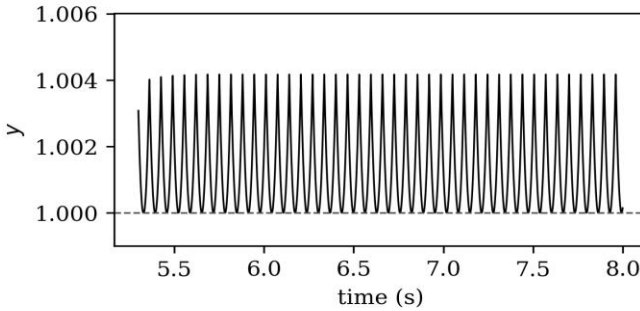


Figure 2. Oscillations of the output in the vicinity of the extremum; the amplitude is of order Δ .

2.3. Comparison with the Time-Varying Search Speed

The sliding mode can alternatively be preserved down to the extremum if the rate of the reference input is gradually decreased using the equivalent control method,

$$\rho = -\rho_0 \left(1 - \frac{|u_{eq}|}{u_0}\right), \quad \tau \dot{u}_{eq} + u_{eq} = \frac{u}{u_0}, \quad (12)$$

where ρ_0 is a positive constant and u_{eq} is the equivalent control value obtained by a first-order filter with a sufficiently small time constant τ ; the sliding mode is maintained under the condition $2\rho_0\tau\ln 2 < \Delta$ (Utkin, 1992). The convergence rate of the modified model is

$$|\dot{y}_{av}| = \rho_0 \left(1 - \frac{\rho_0}{\rho_0 + u_0|f'|}\right), \quad (13)$$

whereas the convergence rate of the unmodified model, calculated from the stability analysis above, is

$$|\dot{y}_{av}| = \frac{A^2 f'^2}{\rho_0}. \quad (14)$$

Comparison of Equations 13 and 14 shows that the unmodified (constant-rate) method is faster at the beginning of the search, while close to the extremum, where $|f'| \rightarrow 0$, the equivalent control method becomes faster. Importantly, however, the accuracy of both systems is of order Δ ; preserving the sliding mode by decreasing the search speed brings no accuracy advantage.

3. Application: Switching Frequency Control of a Three-Phase DC/AC Inverter

3.1. Model and Problem Statement

DC/AC inverters are commonly used for controlling three-phase loads such as AC motors and uninterruptible power supplies. Pulse width modulation (PWM) techniques—space vector PWM, sine-triangle PWM, and hybrid PWM—provide different efficiencies in terms of robustness, harmonic losses, and tracking ability (Holtz, 1992). Sliding mode control is a natural preference for power converter control, since the switching between two discrete values can be used directly as the gating signals of the semiconductor devices (Utkin, Guldner, & Shi, 2009). Typical switching power supplies operate at 75–90% efficiency, and the lost energy is dissipated as heat; elevated temperature reduces the life of electronic components and increases the failure rate. In hysteresis-based control, decreasing the width of the hysteresis loop improves accuracy but increases the switching frequency, and hence the heat losses. The design problem is therefore to minimize the total switching frequency for a given hysteresis width or, equivalently, to decrease the hysteresis width while keeping the frequency within an admissible range.

Figure 3 illustrates the circuit diagram of a three-phase full-bridge inverter implemented on an induction motor. The converter control is three-dimensional, with each leg switching between two values. In this design, two phase currents track preset references, while the third degree of freedom—unused in conventional hysteresis current control—is employed to control the switching frequency through the neutral point voltage V_N .

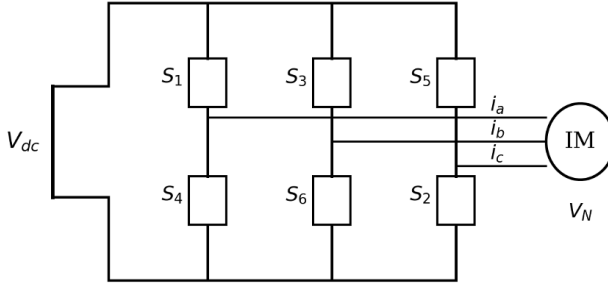


Figure 3. Circuit diagram of the three-phase full-bridge inverter and induction motor load.

Assuming an ideal three-phase full-bridge inverter, the system is governed by

$$Li_k + Ri_k + E_k = H_k V_{dc} - V_N, \quad k = a, b, c, \quad (15)$$

where H_1, H_3, H_5 are the switching functions of the inverter legs, $E_a + E_b + E_c = 0$ are the back-EMF voltages of the load, and the neutral point voltage is

$$V_N = \frac{V_{dc}}{3} (H_1 + H_3 + H_5). \quad (16)$$

By virtue of Kirchhoff's laws the phase current dynamics can be rewritten as

$$i_a = K_a + \frac{V_{dc}}{3L} (2H_1 - H_3 - H_5), \quad K_a = -\frac{R}{L} i_a - \frac{E_a}{L}, \quad (17)$$

with analogous expressions for phases b and c . The torque and speed of the machine define the desired currents; the control system must track these currents while selecting V_N such that the sum of the three switching frequencies in the legs of the converter—and with it the heat loss—is minimized.

3.2. Sliding Mode Control Algorithm

The controlled variables are the two phase currents, with references i_a^* and i_b^* , and the neutral point voltage, with reference V_N^* . Since V_N is a discontinuous function, the objective is to make its average value equal to V_N^* . Three sliding surfaces are introduced:

$$s_a = i_a^* - i_a, \quad s_b = i_b^* - i_b, \quad s_V = \int (V_N^* - V_N) d\tau \quad (18)$$

The differential equations for s_a, s_b, s_V are interconnected with respect to the control. A new set of surfaces decouples the design (Utkin, Guldner, & Shi, 2009):

$$\bar{s} = \frac{1}{3L} \begin{pmatrix} Ls_a + s_V \\ Ls_b + s_V \\ -L(s_a + s_b) + s_V \end{pmatrix} = \begin{pmatrix} \bar{s}_1 \\ \bar{s}_2 \\ \bar{s}_3 \end{pmatrix}. \quad (19)$$

Differentiation yields

$$\dot{\bar{s}} = a - \frac{V_{dc}}{3L} \begin{pmatrix} H_1 \\ H_3 \\ H_5 \end{pmatrix}, \quad a_i = g_i + \frac{1}{3L} V_N^*, \quad (20)$$

where the vector a does not depend on the control variables and $g_1 + g_2 + g_3 = 0$. The switches are triggered by

$$H_1 = \text{sign}(\bar{s}_1), \quad H_3 = \text{sign}(\bar{s}_2), \quad H_5 = \text{sign}(\bar{s}_3) \quad (21)$$

and, provided $V_{dc}/3L > |a_i|$, each component of $\bar{s}$ reaches zero after a finite time and a sliding mode occurs on the manifold $\bar{s} = 0$. With hysteresis of width Δ in the switching devices, the on and off intervals of leg i are

$$t_{ONi} = \frac{\Delta}{\frac{V_{dc}}{3L} - a_i}, \quad t_{OFFi} = \frac{\Delta}{\frac{V_{dc}}{3L} + a_i}, \quad (22)$$

so that the switching frequency of leg i and the total switching frequency are

$$f_i = \frac{3L}{2\Delta V_{dc}} \left[\left(\frac{V_{dc}}{3L} \right)^2 - a_i^2 \right], \quad (23)$$

and the total switching frequency of the converter is

$$f_\Sigma = \frac{3L}{2\Delta V_{dc}} \left[3 \left(\frac{V_{dc}}{3L} \right)^2 - \sum_{i=1}^3 (g_i + \alpha V_N^*)^2 \right], \quad (24)$$

with $\alpha = 1/3L$. Since $\sum g_i = 0$, the cross terms vanish and

$$\frac{\partial f_\Sigma}{\partial V_N^*} = -\frac{1}{\Delta V_{dc} L} V_N^*, \quad (25)$$

so that f_Σ attains its maximum at $V_N^* = 0$, which is exactly the operating point of conventional hysteresis current control. The total switching frequency is minimized by the largest admissible magnitude of V_N^* : the points g_i are shifted up or down by V_N^* subject to $|a_i| < V_{dc}/3L$, which yields, for $|g_i| > |g_j| > |g_k|$,

$$V_N^* = \left[\frac{V_{dc}}{3L} - |g_j| \right] \text{sign}(g_j). \quad (26)$$

Figure 4 shows the dependence of the normalized total switching frequency on V_N^* computed from Equation 24: any nonzero admissible V_N^* decreases the frequency, and hence the heat losses, relative to the conventional operating point. The same mechanism can also be used in the opposite direction: when a maximum admissible frequency is prescribed, a proper V_N^* equalizes the leg frequencies below that limit, which allows the hysteresis width to be decreased and the tracking accuracy to be improved.

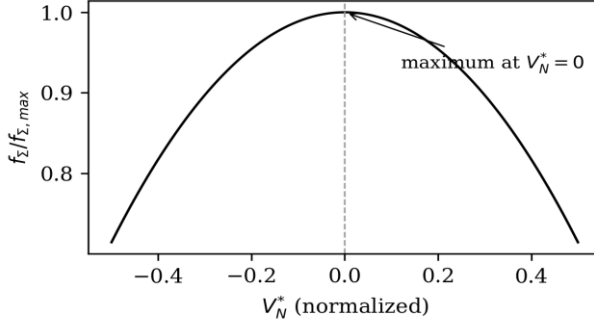


Figure 4. Dependence of the normalized total switching frequency f_{Σ} on the neutral voltage reference V_N^* for $g_1 = 0.5$, $g_2 = g_3 = -0.25$, $V_{dc}/3L = 1$.

3.3. Simulation Results

The control system was simulated numerically. In the first set of simulations, the current references were selected as sinusoids of amplitude 10 A and frequency 60 Hz with 120° phase shift. Figure 5 demonstrates that the real currents coincide with their references in the sliding mode; the average value of the neutral voltage likewise tracks its reference after a short transient.

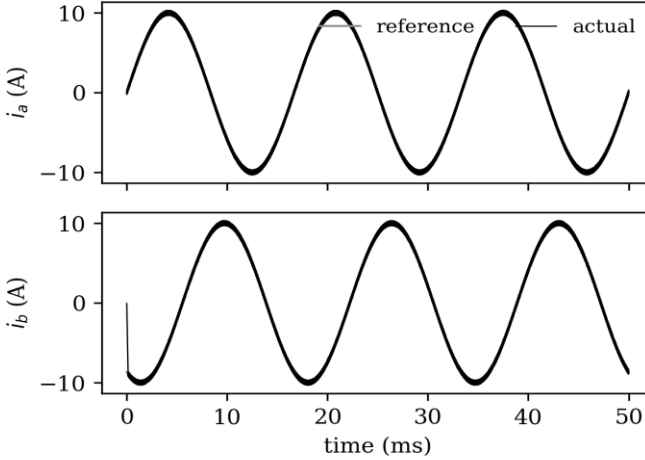


Figure 5. Phase current tracking under sliding mode control; reference and actual currents coincide.

The second set of simulations demonstrates the frequency control mechanism for constant $g_1 = 0.5$, $g_2 = g_3 = -0.25$ and $V_{dc}/3L = 1$. With $V_N^* = 0$, the switching frequency associated with $g_2 = g_3$ corresponds to the maximum admissible value, while the frequency of leg 1 is lower. Selecting $V_N^* = -0.125$ according to Equation 26 equalizes the frequencies of all three legs at a value below the maximum admissible frequency, as shown in Figure 6. Consequently

the hysteresis width can be decreased until the equalized frequency again reaches the admissible limit, improving the current tracking accuracy at no cost in switching losses.

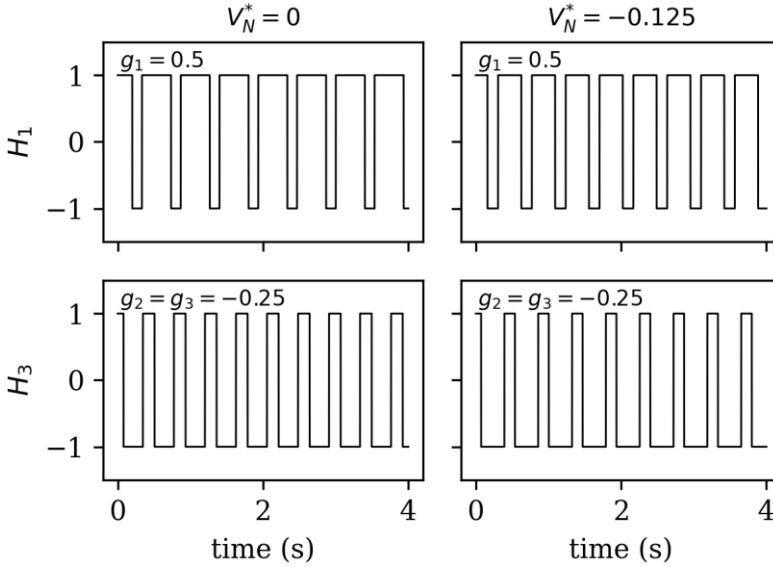


Figure 6. Switching functions of legs 1 and 3 for $V_N^* = 0$ (left) and $V_N^* = -0.125$ (right); the nonzero neutral voltage reference equalizes the leg frequencies.

4. Conclusion

This chapter presented a gradient-free self-optimization methodology based on sliding mode control. The search algorithm forces the plant output to track a monotonously decreasing reference input; a discontinuous control with two switching surfaces handles the unknown, sign-varying local gain, and an auxiliary hysteresis function restores the sliding mode initial conditions in the vicinity of the extremum. The central theoretical result is that the accuracy of the search process is determined solely by the width of the hysteresis loop of the switching element: it depends neither on the rate of the reference input nor on the sliding mode existence conditions. Comparison with the time-varying search speed modification showed that preserving the sliding mode near the extremum alters the convergence rate but not the accuracy.

The methodology was applied to the switching frequency control of a three-phase DC/AC inverter feeding an induction motor. Using the third degree of freedom of the converter—the neutral point voltage—the total switching frequency was shown to be maximal at the conventional operating point $V_N^* = 0$ and reducible by an admissible shift of the neutral voltage reference. Simulations confirmed accurate phase current tracking and demonstrated both the reduction

and the equalization of leg switching frequencies, which translates directly into lower converter heat losses or, alternatively, improved tracking accuracy at a prescribed admissible frequency. Extensions of the same self-optimization principle to multidimensional optimization, underground coal gasification output maximization, and focus-tunable lens particle tracking are developed in the source dissertation (Cakanel, 2017).

Acknowledgment

This chapter is derived from the author's doctoral dissertation entitled *Self-Optimization Systems Design Based on Sliding Mode Control*, completed at The Ohio State University, Department of Electrical and Computer Engineering, in 2017, under the supervision of Prof. Vadim I. Utkin (Cakanel, 2017). Parts of the material summarized here were presented in Utkin and Cakanel (2016) and Cakanel and Utkin (2016). The author expresses his sincere gratitude to Prof. Vadim I. Utkin for his guidance and support.

References

- Ariyur, K. B., & Krstić, M. (2003). *Real-time optimization by extremum-seeking control*. Hoboken, NJ: Wiley-Interscience.
- Cakanel, A. (2017). *Self-optimization systems design based on sliding mode control* (Doctoral dissertation). The Ohio State University, Columbus, OH.
- Cakanel, A., & Utkin, V. (2016). Frequency control of DC/AC inverter. In *2016 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)* (pp. 1–6). Ixtapa, Mexico: IEEE. doi:10.1109/ROPEC.2016.7830632
- Holtz, J. (1992). Pulsewidth modulation—A survey. *IEEE Transactions on Industrial Electronics*, 39(5), 410–420.
- Korovin, S. K., & Utkin, V. I. (1974). Using sliding modes in static optimization and nonlinear programming. *Automatica*, 10(5), 525–532.
- Krstić, M., & Wang, H.-H. (2000). Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, 36(4), 595–601.
- Utkin, V. I. (1977). Variable structure systems with sliding modes. *IEEE Transactions on Automatic Control*, 22(2), 212–222.
- Utkin, V. I. (1992). *Sliding modes in control and optimization*. Berlin, Germany: Springer-Verlag.
- Utkin, V., Guldner, J., & Shi, J. (2009). *Sliding mode control in electromechanical systems* (2nd ed.). Boca Raton, FL: CRC Press.
- Utkin, V. I., & Cakanel, A. (2016). Chattering analysis of sliding mode self-optimization systems. In *2016 14th International Workshop on Variable Structure Systems (VSS)* (pp. 1–6). Nanjing, China: IEEE. doi:10.1109/VSS.2016.7506899

CHAPTER 3

Analysis of The Interaction Between IMF Parameters and Geomagnetic Indices

Burhaneddin Bilgen¹

Sercan Bülbul²

Fuat Başçiftçi³

¹ Academy of Land Registry and Cadastre, Ankara Hacı Bayram Veli University, Türkiye

² Department of Geomatic Engineering, Faculty of Engineering and Natural Sciences, Konya Technical University, Türkiye

³ Map and Cadastre Programme, Vocational School of Technical Sciences, Karamanoglu Mehmetbey University, Türkiye

Introduction

Space weather events resulting from solar activity can significantly affect the interactions between the Earth's magnetosphere and the solar wind, potentially leading to geomagnetic storms. Such events can have various impacts on satellite systems, communication infrastructure, global positioning systems, and power transmission networks. Therefore, investigating the relationships between solar wind parameters and geomagnetic indices is considered an important area of research in space weather studies (Dai et al., 2025; Tulasi Ram et al., 2024).

Geomagnetic storms primarily occur as a result of the interaction of coronal mass ejections (CMEs) and high-speed solar wind streams originating from the Sun with the Earth's magnetosphere. During this process, the magnitude and direction of the interplanetary magnetic field (IMF), particularly the Bz component of the IMF, are considered among the key factors controlling energy transfer between the solar wind and the magnetosphere. A southward-directed Bz component enhances magnetic reconnection processes between the magnetosphere and the solar wind, thereby increasing energy transfer within the system (Chilingarian et al., 2024).

Recent studies have shown that prolonged periods of southward-directed IMF Bz conditions can substantially increase the intensity of geomagnetic activity. In particular, the persistence and magnitude of the southward component of the IMF have a significant influence on the development and intensity of geomagnetic storms (Pandey et al., 2025). Therefore, evaluating IMF parameters together with geomagnetic indices is of critical importance for understanding solar wind–magnetosphere interactions.

Recent intense space weather events have also clearly demonstrated these relationships. For example, during the strong geomagnetic storm that occurred in May 2024, changes in IMF orientation and solar wind parameters resulted in significant variations in geomagnetic indices (Tulasi Ram et al., 2024). Similarly, studies have shown that space weather conditions following intense solar flares and CME events can cause significant changes in the geospace environment (Omojola et al., 2025).

In this context, the joint analysis of geomagnetic indices and IMF parameters is of great importance for understanding space weather processes and assessing geomagnetic activity. In this study, the relationships between geomagnetic indices (Kp, Dst, and F10.7) and IMF magnitude and Bz components (Bz_GSE and Bz_GSM) were investigated using space weather data covering a 31-day period. The analyzed dataset contains a total of 744 hourly measurements, corresponding to a 31-day observation period. Using this dataset, the temporal variations of the parameters and the relationships among them were evaluated through graphical and comparative analysis methods.

1. Materials and Methods

In this study, various space weather parameters and indices were used to assess geomagnetic activity. The primary parameters considered in the analysis were the Kp index, Dst index, F10.7 solar radio flux, and the Bz component of the interplanetary magnetic field (IMF). These parameters are widely used in

space weather research and provide important information for evaluating the interactions between the solar wind and the Earth's magnetosphere.

The data used in the study have an hourly temporal resolution and cover a total of 744 hours (31 days) of observations during the analyzed period. The temporal variations of the parameters and the relationships among them were investigated using graphical analysis methods.

1.1. Kp Index

The Kp index is one of the most widely used indicators of geomagnetic activity and represents the intensity of global geomagnetic disturbances in the Earth's magnetic field. The Kp index is calculated based on measurements obtained from magnetometer stations located at mid-latitudes around the Earth and is expressed on a dimensionless scale ranging from 0 to 9. Low Kp values indicate quiet geomagnetic conditions, whereas high values represent geomagnetic storm conditions (Chilingarian et al., 2024).

The Kp index is particularly used for the global-scale assessment of solar wind–magnetosphere interactions. Values of $Kp \geq 5$ generally indicate the onset of geomagnetic storm conditions, which may cause auroral activity to extend toward lower latitudes The phrase '(Figure 1)' should be added to the end of this sentence. (Tulasi Ram et al., 2024).

Green	Calm or small geomagnetic disturbance	Kp-Index 0–4	No effect on devices or people
Yellow	Weak/minor geomagnetic storm	Kp-Index 5 / G1	Weak fluctuations in the electrical grid, minor effects on the operation of space satellites, as well as on the migration of animals are quite possible
Dark yellow	Moderate geomagnetic storm	Kp-Index 6 / G2	Power systems located at high latitudes can experience emergency situations. Prolonged geomagnetic storms can damage transformers. HF radio signals may weaken
Orange	Strong geomagnetic storm	Kp-Index 7 / G3	False alarms may be triggered on some protective electronic devices. Correction of satellite orientation and navigation in outer space may be required
Red	Severe geomagnetic storm	Kp-Index 8 / G4	There may be widespread problems with power grid voltages. Satellite navigation may worsen for several hours, and LF radio navigation may be disrupted
Dark red	Extreme geomagnetic storm	Kp-Index 9 / G5	Power systems may experience transformer damage and a complete collapse. HF radio communications may not be possible. Satellite navigation may be disrupted

Figure 1. Kp index threshold values

1.2. Dst Index

The Dst (Disturbance Storm Time) index is an important geomagnetic index that represents the intensity of the ring current generated in the Earth's magnetosphere during geomagnetic storms. The Dst index is calculated using measurements obtained from equatorial magnetometer stations and is expressed in nanotesla (nT).

During geomagnetic storms, the transfer of energy from the solar wind strengthens the ring current in the magnetosphere, causing the Dst index to decrease toward negative values. Therefore, the Dst index is widely used to assess the intensity of geomagnetic storms (Dai et al., 2025).

In general, geomagnetic conditions are classified as follows:

- Dst > −30 nT: quiet conditions

- $-30 \text{ nT} > \text{Dst} > -50 \text{ nT}$: weak storm
- $-50 \text{ nT} > \text{Dst} > -100 \text{ nT}$: moderate storm
- $\text{Dst} < -100 \text{ nT}$: strong geomagnetic storm (Pandey et al., 2025).

1.3. F10.7 Solar Radio Flux

The F10.7 index represents the solar radio flux at a wavelength of 10.7 cm (2800 MHz) and is widely used as an important indicator of solar activity. This index was initially calculated based on measurements made at the Penticton Observatory in Canada and is currently widely used in space weather research.

The F10.7 index is expressed in solar flux units (sfu) ($1 \text{ sfu} = 10^{-22} \text{ W m}^{-2} \text{ Hz}^{-1}$). This parameter is particularly considered an indicator of solar radiation emitted into the upper atmosphere and extreme ultraviolet (EUV) energy input (Qian & Mursula, 2025).

The F10.7 index is also used as a proxy for solar activity in ionospheric and thermospheric models and plays an important role in assessing energy input into the upper atmosphere (Tulasi Ram et al., 2024).

1.4. IMF Bz Component

The Bz component of the interplanetary magnetic field (IMF) is considered one of the most important parameters controlling energy transfer between the solar wind and the Earth's magnetosphere. A southward-directed (negative) Bz component of the IMF enhances the magnetic reconnection process between the magnetosphere and the solar wind.

As a result of this process, a significant amount of energy is transferred from the solar wind into the magnetosphere, which can lead to increased geomagnetic activity. Numerous studies have demonstrated that prolonged periods of southward-directed IMF Bz conditions play a critical role in the development of geomagnetic storms (Pandey et al., 2025; Chilingarian et al., 2024).

2. Application

In this study, the relationships between geomagnetic indices and interplanetary magnetic field parameters were investigated using space weather data covering a 31-day period. The analyzed dataset contains a total of 744 hourly measurements, corresponding to a 31-day observation period. This temporal resolution provides a suitable data structure for examining the response of geomagnetic activity to solar wind parameters over short time scales. The temporal variations of the Kp, Dst, and F10.7 indices, together with IMF magnitude, Bz_GSE, and Bz_GSM parameters, were analyzed. The relationships among these parameters were evaluated through both time-series analysis and comparative graphical assessments.

2.1. Temporal Variations of the Parameters

As a first step, the time-dependent variations of Kp, Dst, IMF, and the Bz components were analyzed over the 31-day period under investigation. This analysis was conducted to identify simultaneous variations between geomagnetic activity and solar wind parameters.

2. The time-series variations of the relevant parameters are presented in Figure

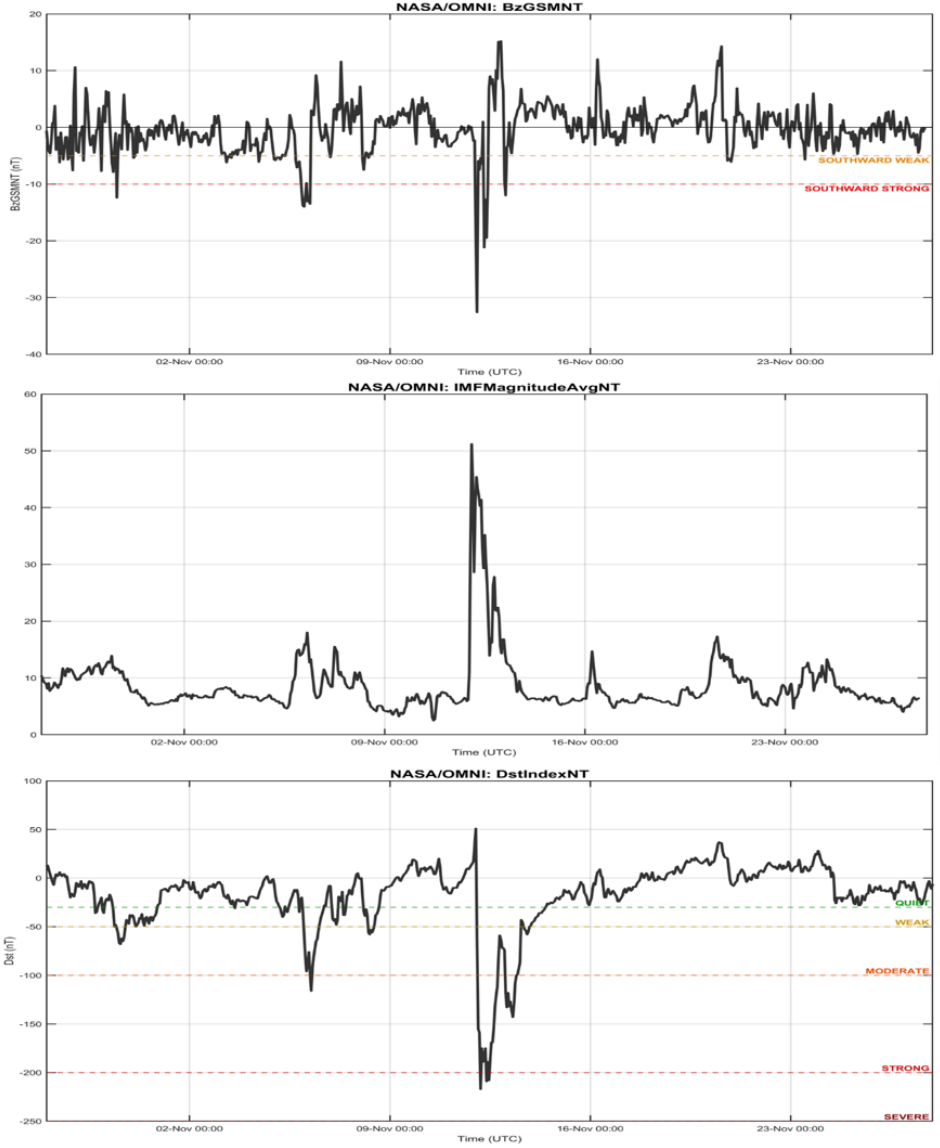


Figure 2. Temporal variations of the Dst index, IMF magnitude, and Bz components over the 31-day period

As shown in Figure 2, decreases in the Dst index, particularly, occur simultaneously with increases in IMF magnitude and variations in the Bz components. This indicates the determining role of solar wind–magnetosphere interactions in geomagnetic activity.

2.2. Relationship Between IMF Magnitude and Geomagnetic Activity

In the second stage of the analysis, the relationship between IMF magnitude and geomagnetic indices was investigated. In particular, a clear relationship was observed between the Kp index and IMF magnitude. This relationship is illustrated in the scatter plot presented in Figure 3.

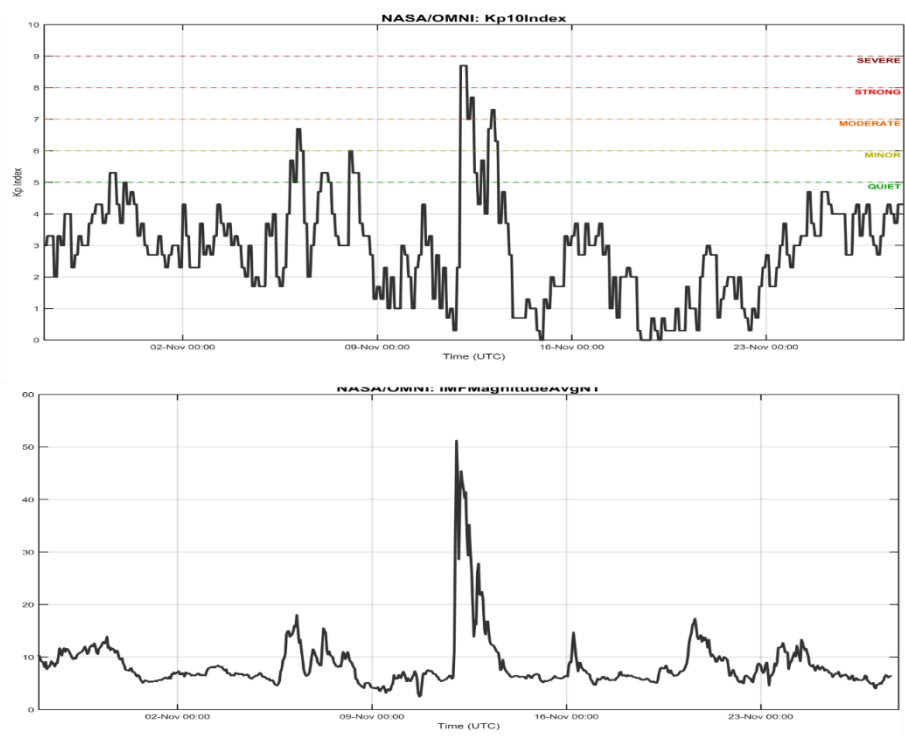


Figure 3. Scatter plot showing the relationship between the Kp index and IMF magnitude

As shown in Figure 3, the Kp index tends to increase as IMF magnitude increases. This indicates that strong magnetic field conditions in the solar wind have a significant influence on geomagnetic activity.

2.3. Variations in Dst as a Function of IMF Magnitude

The relationship between variations in the Dst index and IMF magnitude was also investigated. The analysis results indicate a negative relationship between these two parameters. This relationship is presented in Figure 4.

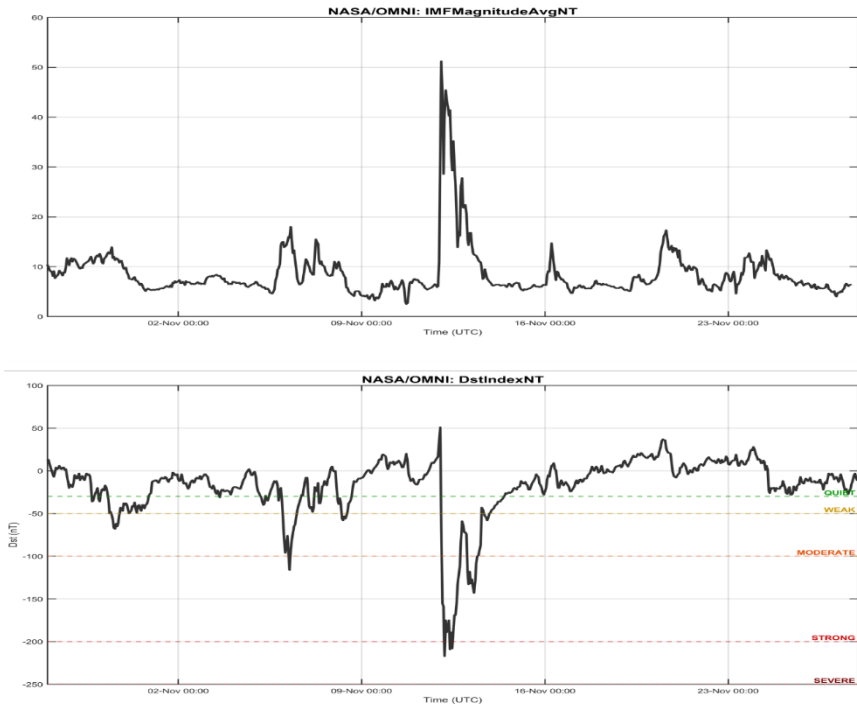


Figure 4. Relationship between the Dst index and IMF magnitude

As shown in Figure 4, the Dst index tends to shift toward more negative values as IMF magnitude increases. This indicates that energy accumulation in the magnetosphere has a significant effect on geomagnetic indices.

2.4. Effect of the Bz Component

To investigate the effect of the IMF z-component on geomagnetic activity, the relationship between the Bz_{GSM} component and the Kp index was analyzed. This relationship is presented in Figure 5.

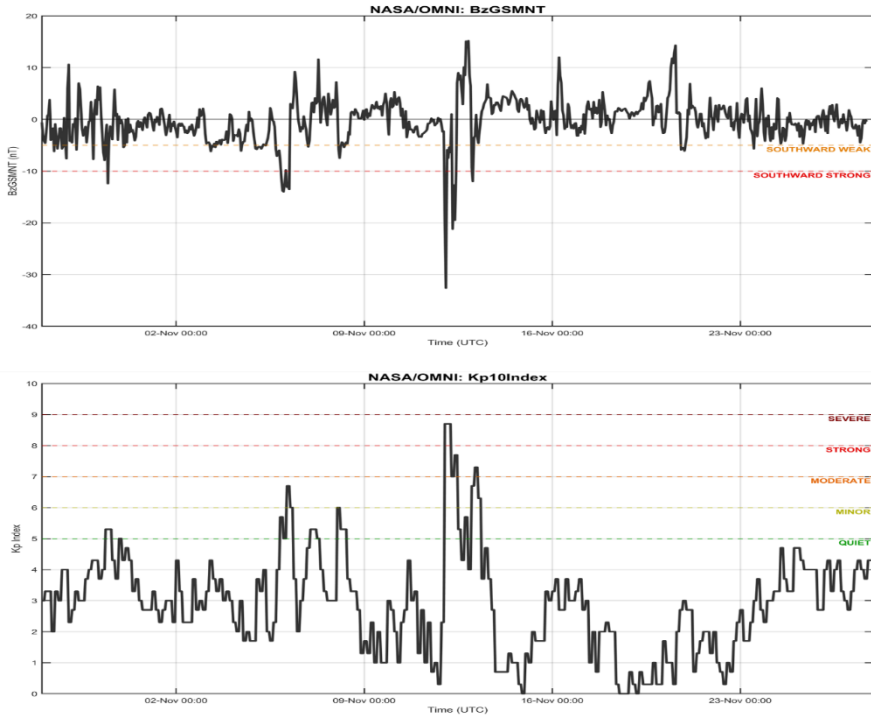


Figure 5. Relationship between the Kp index and the Bz_GSM component

As shown in Figure 5, negative Bz values are particularly associated with increased geomagnetic activity. This can be considered an important indicator of the interaction between the solar wind and the magnetosphere.

2.5. Comparison of the Bz_GSE and Bz_GSM Components

To examine the behavior of the IMF z-component in different coordinate systems, the Bz_GSE and Bz_GSM parameters were compared. This comparison is presented in Figure 6.

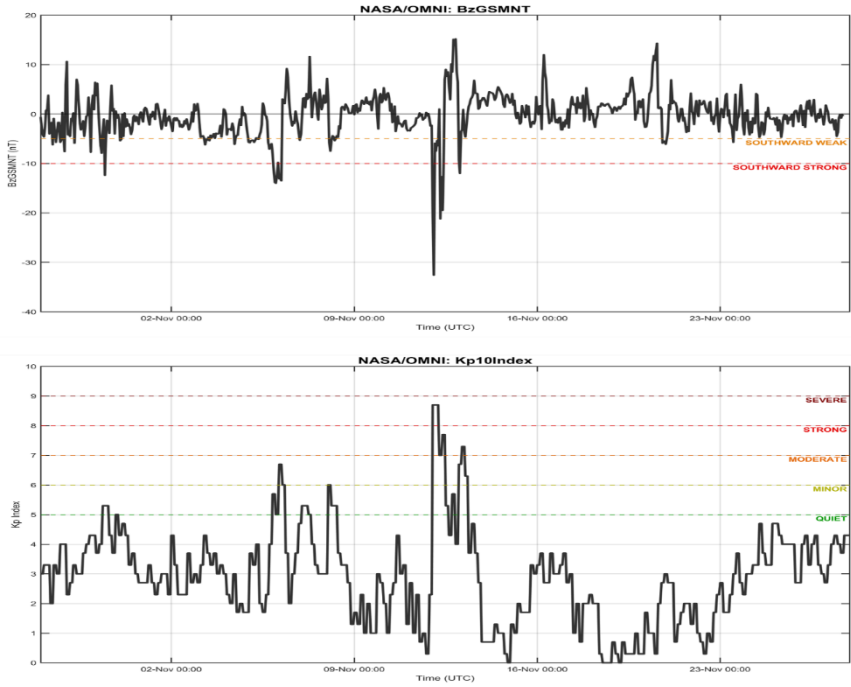


Figure 5. Relationship between the Kp index and the Bz_GSM component

As shown in Figure 5, negative Bz values are particularly associated with increased geomagnetic activity. This can be considered an important indicator of the interaction between the solar wind and the magnetosphere.

2.6. Comparison of the Bz_GSE and Bz_GSM Components

To examine the behavior of the IMF z-component in different coordinate systems, the Bz_GSE and Bz_GSM parameters were compared. This comparison is presented in Figure 6.

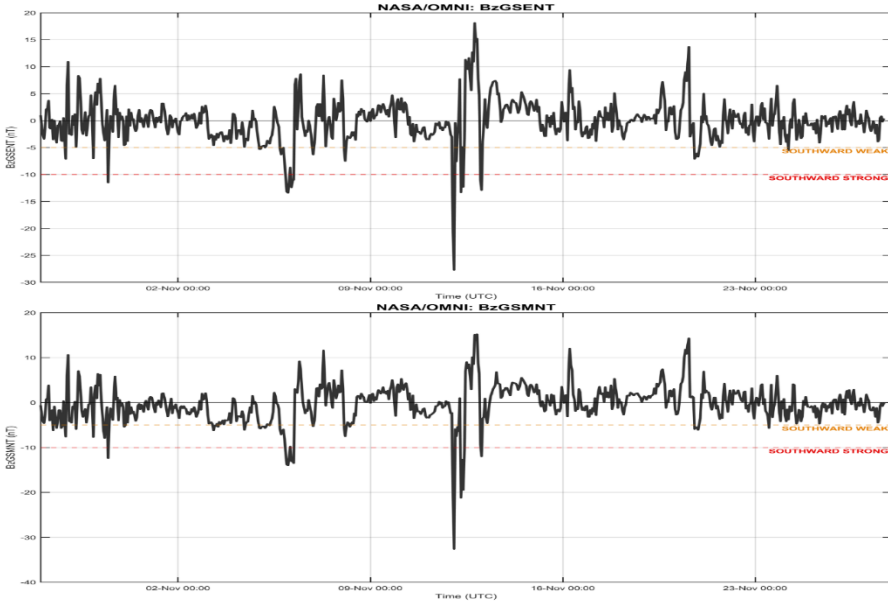


Figure 6. Comparison of the Bz_GSE and Bz_GSM components over the analyzed period.

The analysis results indicate that the temporal variations of these two parameters exhibit largely similar behavior.

2.7. Relationship with Solar Activity (F10.7)

Finally, the relationships between the F10.7 index and the geomagnetic indices and IMF parameters were investigated. This comparison is presented in Figure 7.

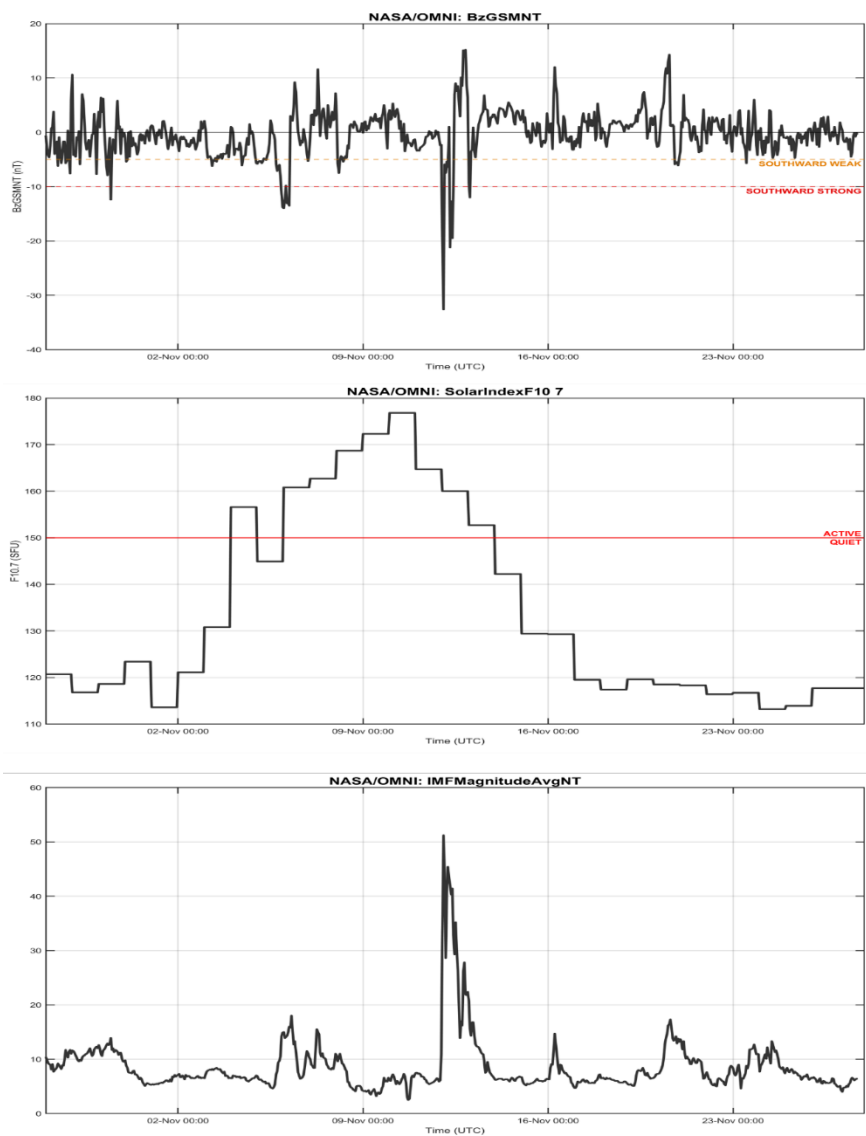


Figure 7. Relationship between the F10.7 index and geomagnetic parameters.

The analysis results indicate that, on short time scales, geomagnetic activity is influenced more strongly by variations in IMF parameters, particularly the Bz component, than directly by the F10.7 index.

3. Conclusion

The analysis of the 31-day dataset demonstrates that significant relationships exist between the geomagnetic indices examined and the interplanetary magnetic field parameters. The evaluation of temporal variations and comparative analyses provided important insights into the interactions between solar wind conditions and geomagnetic activity.

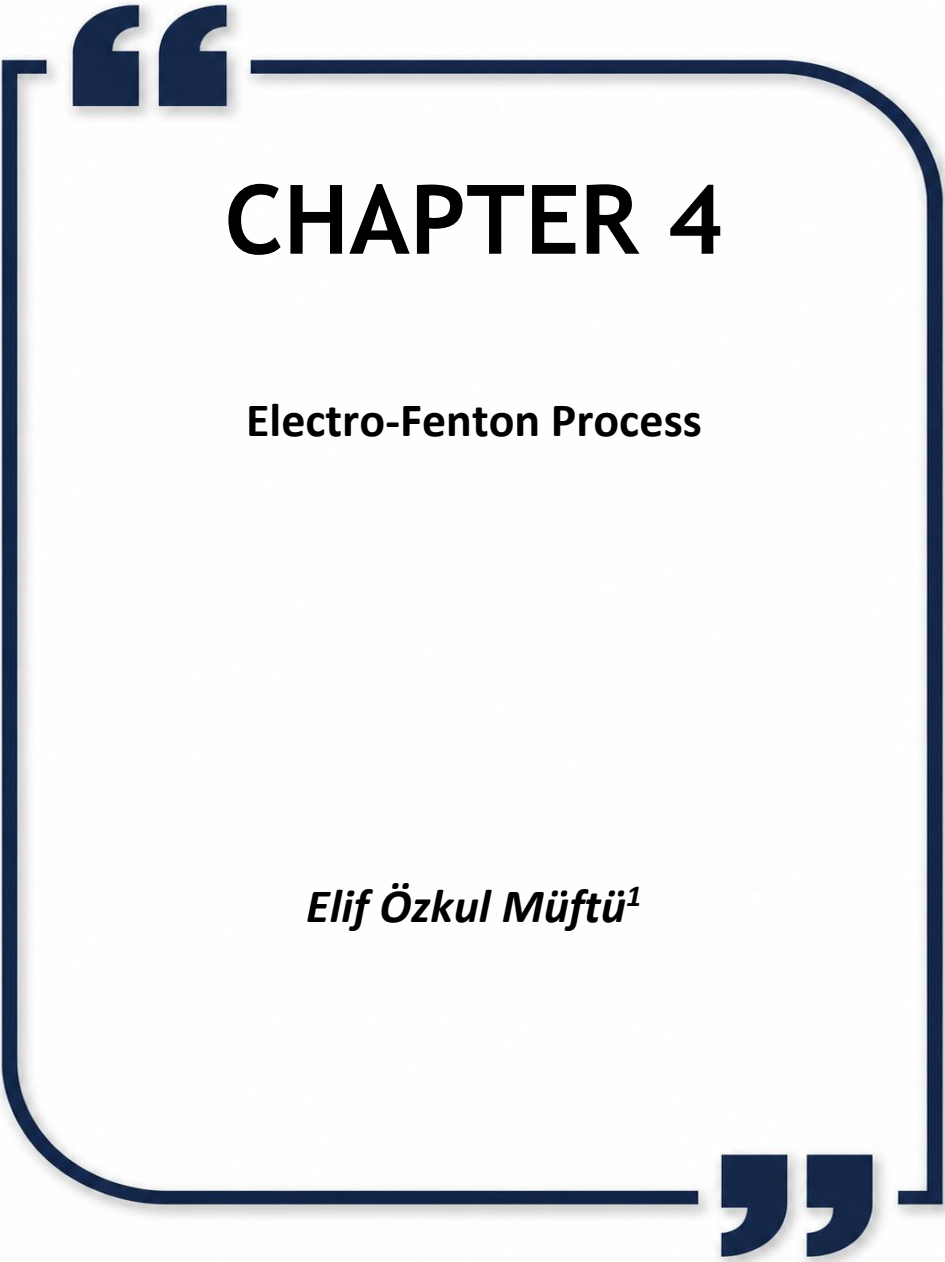
The main findings of the study are summarized below:

- A clear relationship was observed between IMF magnitude and geomagnetic activity. During the analyzed period, increases in IMF magnitude were generally observed to occur simultaneously with increases in geomagnetic activity.
- The Dst index was found to be sensitive to variations in IMF magnitude. During periods characterized by stronger IMF conditions, the Dst index tended to shift toward more negative values, which was associated with intensified geomagnetic disturbances.
- The Bz component was found to play an important role in geomagnetic variability. In particular, geomagnetic activity increased during periods when negative Bz values were observed. This demonstrates the influence of IMF orientation on geomagnetic processes.
- The temporal variations of the Bz_GSE and Bz_GSM components exhibited largely similar behavior. This indicates that both coordinate systems reflect the same underlying physical processes involved in solar wind–magnetosphere interactions.
- The relationship between the F10.7 index and short-term geomagnetic variations was found to be relatively weak. This result suggests that short-term geomagnetic variations are more strongly controlled by solar wind parameters than by solar radio flux.
- The combined evaluation of geomagnetic indices and IMF parameters provides a better understanding of the interaction between the solar wind and the magnetosphere. In particular, IMF magnitude and the direction of the Bz component emerged as important determinants of the observed geomagnetic variations.

In conclusion, the findings demonstrate that, over the analyzed 31-day period, variations in IMF magnitude and the Bz component played an important role in the occurrence and development of geomagnetic activity.

4. References

- Chilingarian, A., Karapetyan, T., Sargsyan, B., Asatryan, K., & Gabaryan, G. (2024). Influence of interplanetary coronal mass ejections on terrestrial particle fluxes through magnetosphere disturbances. *arXiv*. <https://doi.org/10.48550/arXiv.2406.16159>
- Dai, X., Fu, H., Yan, Z., Wang, Z., Xu, F., Wang, C., ... & Jin, Y. (2025). Extreme Solar Storm Reveals Causal Interactions in Space Weather. *arXiv preprint arXiv:2508.06507*.
- Pandey, R., Pandey, A., & Pandey, S. (2025). Space weather impacts of long-duration southward IMF Bz events during Solar Cycle 24. *International Journal of Scientific Innovation and Engineering*, 2(11), 1545–1551.
- Tulasi Ram, S., Veenadhari, B., Dimri, A. P., Bulusu, J., Bagiya, M., Gurubaran, S., ... & Singh, R. (2024). Super-intense geomagnetic storm on 10–11 May 2024: Possible mechanisms and impacts. *Space Weather*, 22(10), e2024SW004126. <https://doi.org/10.1029/2024SW004126>
- Omojola, J., Greer, K., & Moeketsi, D. (2025). Geospace response to May 2024 X-class flare and coronal mass ejection. *Advances in Space Research*. 76 (12), <https://doi.org/10.1016/j.asr.2025.04.036>
- Qian, L., & Mursula, K. (2025). Evaluating F10.7 and F30 radio fluxes as long-term solar proxies of energy deposition in the thermosphere. *Annales Geophysicae*, 43(1), 45–58. <https://doi.org/10.5194/angeo-43-175-2025>



CHAPTER 4

Electro-Fenton Process

Elif Özkul Müftü¹

¹ Res. Asst., Pamukkale University, Chemical Engineering Department, Denizli.
ORCID: 0000-0001-8173-6690

INTRODUCTION

Water resources are of vital importance for the continuity of ecosystems and the sustainability of human life. Although there is a vast amount of water on Earth, saltwater and glaciers are, in their current state, either unusable or inaccessible. Freshwater sources account for only a very small proportion of the world's total water mass. A large proportion of these freshwater sources is also being polluted due to various natural and anthropogenic causes, such as domestic, industrial and irrigation use. In particular, the treatment and recovery of industrial wastewater are of great importance. Furthermore, water-related problems are expected to escalate significantly due to climate change and population growth. Consequently, the development of efficient, sustainable and environmentally friendly methods has become essential (Kurian, 2021). Chlorination, which for many years was regarded as a practical and easily implementable method, produces by-products that are potentially hazardous to health and has become unsuitable for use due to the raised standards for drinking water. However, scientists have tested multifaceted techniques incorporating physical, chemical and biological methods for the treatment of wastewater as efficient cost-effective alternatives. Furthermore, they are not suitable for large-scale treatment due to drawbacks such as the need for adsorption regeneration in adsorption techniques, higher operating costs in membrane filtration technologies, and sludge formation in sedimentation techniques, etc. (Kurian, 2021). Whilst biological methods are simple and inexpensive to implement, they present challenges such as the inability to adequately treat pollutants present at high concentrations. Among these treatment technologies, advanced oxidation processes, which rely on the formation of highly reactive hydroxyl radicals ($\bullet\text{OH}$), emerge as a powerful alternative. The most commonly used advanced oxidation processes are photocatalytic oxidation, ozone combined with hydrogen peroxide ($\text{O}_3/\text{H}_2\text{O}_2$), ultraviolet (UV), ultraviolet combined with ozone (UV/O_3), ultraviolet combined with hydrogen peroxide ($\text{UV}/\text{H}_2\text{O}_2$), the Fenton process ($\text{Fe}^{2+}/\text{H}_2\text{O}_2$), photo-Fenton ($\text{Fe}^{2+}/\text{H}_2\text{O}_2/\text{UV}$) and electro-Fenton. The electro-Fenton method is an electrochemical advanced oxidation process based on Fenton chemistry. It utilises hydroxyl radicals generated as a result of the reaction between Fe^{2+} and H_2O_2 reagents which are either produced in situ under an electric field or added externally for the removal of persistent organic pollutants. This method is effective for the treatment of synthetic and natural wastewater contaminated with various organic substances, such as pharmaceuticals, personal care products, endocrine-disrupting compounds, pesticides, textile waste, polycyclic aromatic hydrocarbons, surfactants and leachate from landfill sites (Ganiyu et. al., 2018). Fenton processes, on the other hand, are divided into two categories: homogeneous and heterogeneous. The aim of this study is to understand the homogeneous electro-Fenton process one of the electro-Fenton variants and to comprehend the reactions that take place within the process.

1. THE HOMOGENEOUS ELECTRO-FENTON PROCESS

Conventional wastewater treatment technologies may prove inadequate for the removal of certain persistent and complex organic pollutants. For this reason,

Advanced Oxidation Processes (AOPs), which possess a high oxidation capacity, are becoming increasingly important as alternative or complementary technologies, particularly for the treatment of toxic organic compounds that are difficult to degrade biologically. Among AOPs, methods based on the Fenton reaction are one of the oxidative technologies capable of demonstrating high efficacy in wastewater treatment. These methods are based on the generation of reactive species with high oxidising power, such as hydroxyl radicals ($\bullet\text{OH}$) and the conversion of organic pollutants by these species into simpler or less harmful compounds (Fenton, 1894).

Among the significant advantages of Fenton-based processes are their relatively simple operating conditions and their ability to achieve high efficiency in the degradation of a wide range of organic and toxic pollutants (Wang and Zhuan, 2020). However, the traditional Fenton process also has certain limitations. Safety risks that may arise during the storage and transport of hydrogen peroxide (H_2O_2), along with the cost of using this chemical, are among the significant disadvantages. Furthermore, the fact that the process generally operates effectively within a limited and acidic pH range (approximately pH 2–4) results in the need for neutralisation following the treatment. The high volume of iron-containing sludge that can form as a result of using dissolved iron species is also one of the process's significant environmental and operational limitations. Fenton-based systems are generally classified as homogeneous and heterogeneous Fenton processes, depending on the physical state of the iron catalyst in the reaction medium (Babuponnusami and Muthukumar, 2014) (Figure 1).

Compared to other Fenton reactions, the electro-Fenton process offers significant advantages, such as versatility, environmental compatibility, better process control, the of secondary pollutants, and the elimination of the need to use harmful reagents. There are three main steps involved in this method (Liu et. al., 2007; Poza-Nogueiras et. al., 2018):

- i. In-situ production of H_2O_2 at the cathode. This reduces the risks associated with transport and storage.
- ii. Efficient production of OH^- via the reaction between iron ions and H_2O_2 (Fenton reaction)
- iii. Regeneration of Fe^{2+} on the cathode surface

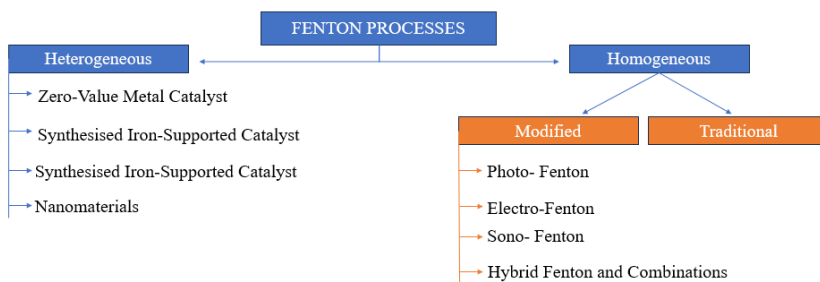
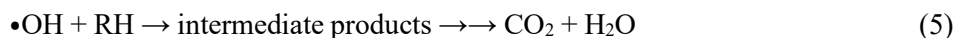


Figure 1. Fenton Processes

Electro-Fenton (EF) technology is an advanced oxidation-based treatment method that combines electrochemical oxidation mechanisms with Fenton reactions. This process, which yields effective results particularly in the treatment of wastewater containing high levels of organic pollution, is based on the generation of strong oxidising species in an electrochemical environment. H_2O_2 formed in the system reacts under the catalytic effect of Fe^{2+} ions, leading to the formation of $\bullet\text{OH}$ with a high oxidation potential. Thanks to their high reactivity, the resulting $\bullet\text{OH}$ radicals attack organic pollutants, enabling the breakdown of these compounds and their conversion into simpler products. The oxidative degradation of organic substances via hydroxyl radicals is generally explained by the sequential chain reactions given below (Kremer, 1999; Barrera-Díaz et. al., 2003; Do et. al., 2017);



The key components of a homogeneous EF system include a carbon-based cathode, a stable anode, a reaction medium and an iron catalyst (Fe^{2+} , Fe^{3+}). During the process, the necessary iron ions can be added to the solution in the form of Fe^{2+} and the reaction is facilitated by the applied electrical energy. In this process, highly reactive hydroxyl radicals ($\bullet\text{OH}$) are formed as a result of H_2O_2 , which is either produced electrochemically or added to the system from an external source, reacting with Fe^{2+} ions. However, the hydroperoxyl radical ($\text{HO}_2\bullet$) may also form in the reaction environment; as this radical has a lower oxidation capacity compared to the $\bullet\text{OH}$ radical, it may adversely affect the oxidative removal efficiency of the process. The $\bullet\text{OH}$ radicals formed, however, play a key role in the degradation and oxidative breakdown of organic pollutants due to their high oxidation capacity. Figure 2 presents a schematic representation of the EF process.

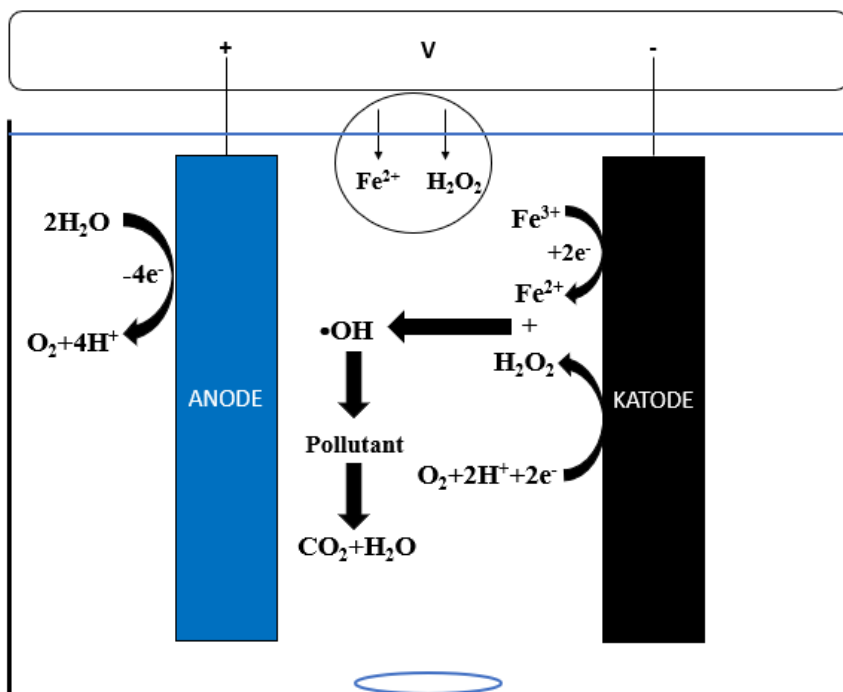


Figure 2. The general operating mechanism and fundamental reactions of the homogeneous EF process

An EF experimental setup generally consists of a power supply that provides electrical energy, a stirring system that ensures the homogeneity of the reaction medium, and an EF cell in which electrochemical reactions take place. The reaction cell usually contains an anode and a cathode positioned at a specific distance from one another. Whilst the current or potential values applied to the system are controlled via the power supply, the solution is continuously stirred using a magnetic stirrer. In laboratory-scale applications, the EF cell generally consists of a beaker of suitable volume and a mounting assembly that secures and positions the electrodes. The distance between the electrodes is a key operating parameter that can influence the cell's electrical resistance, current distribution and, consequently, process performance.

EF processes are classified into two main groups homogeneous and heterogeneous depending, in particular, on the physical form of the iron catalyst used in the reaction medium and the structure of the catalytic system. In homogeneous EF systems, the iron catalyst is present in solution in ionic form, whilst in heterogeneous EF applications, solid catalysts containing iron are utilised. In this section, the homogeneous Electro-Fenton process, in which iron ions are present in solution, will be examined in detail.

In a typical homogeneous EF process, the reactor consists of a single-compartment electrochemical cell operating under direct current, together with the associated anode and cathode electrodes (Mousset et. al., 2017; Poza-

Nogueiras et. al., 2018) (Figure 2). The electrochemical cell facilitates mass transfer and ion movement between the electrodes. Continuous H₂O₂ production is achieved through the electrochemical reduction taking place at the cathode surface in the presence of dissolved oxygen in the system;



The correct selection of the electrocatalytic cathode material is of great importance in order to achieve maximum Faradaic efficiency; in this regard, carbon-based materials in particular are frequently. The controlled in situ production of H₂O₂ in the system eliminates the fundamental limitations encountered in conventional Fenton processes. Indeed, the transport and storage of high-concentration H₂O₂ solutions, as used in traditional applications, are necessary. Given this chemical's as a strong oxidant, explosive and highly reactive substance, its transport and storage pose serious risks in terms of the environment, human health and process safety (Brillas and Garcia-Segura, 2020). Consequently, the controlled production of H₂O₂ can be achieved through a continuous supply of oxygen from the cathode. As with conventional Fenton reactions, the Fe²⁺ ion is the most commonly used catalyst in homogeneous electro-Fenton systems. Soluble Fe²⁺, added externally to the system, reacts with the H₂O₂ produced within the reactor under optimal conditions at approximately pH 3 to facilitate the formation of •OH (Moreira et. al., 2017; Brillas and Garcia-Segura, 2020) (Reaction 1). The most significant advantage that electro-Fenton technology offers over the classical Fenton process is that the Fe³⁺ ions formed as a result of Reaction 1 are continuously reduced back to Fe²⁺ at the cathode surface (Villegas et. al., 2016; Mousset et. al. 2017; Poza-Nogueiras et. al., 2018);



The electrochemical regeneration of Fe²⁺ species occurs more rapidly than the process described in Reaction 1. Thanks to this rapid reduction at the cathode surface, a small amount of Fe²⁺ ions is sufficient to sustain the electro-Fenton process, and consequently, the formation of sludge in the form of Fe(OH)₃ is significantly limited. Furthermore, Fe³⁺ ions can also act as catalysts in electro-Fenton systems (Moreira et. al. 2014).

Electrolytic cells essentially consist of at least two electrodes: the cathode and the anode, where reduction and oxidation reactions take place. The cathode is defined as the electrode where electrons are consumed and reduction reactions occur. The primary functions of the cathode, which plays a critical role in determining the effectiveness of the electro-Fenton process, can generally be listed as follows (Moreira et. al., 2017; Garcia-Segura et. al., 2018):

- i. The formation of H₂O₂ through the reduction of oxygen
- ii. Catalyst regeneration by the electrochemical reduction of the Fe³⁺ ion

In undivided electrochemical cells, oxidation reactions take place simultaneously on the anode surface. The properties and selection of the anode material used in wastewater treatment are key parameters for the advanced

oxidation and mineralisation of organic pollutants. In particular, the oxidative capacity of the anode electrode is a determining factor in the removal of COD and TOC. However, in some studies in the literature, the formation of reactive oxygen species primarily physically adsorbed $\bullet\text{OH}$ on the anode surface and their contribution to the oxidation mechanism have not been sufficiently taken into account. With the anode designated as M (Moreira et. al., 2017; Garcia-Segura et. al., 2018):



The reaction generally accepted for the anode is as follows;



The effectiveness of homogeneous Fenton processes in the removal of organic pollutants including pesticides, chlorophenoxy herbicides and organic pollutants present in real wastewater of varying characteristics has been demonstrated in various studies [18]. However, for the process be carried out effectively, very low pH values (pH 2–4) are generally required. This necessitates the implementation of an additional neutralisation stage to bring the water to an appropriate pH level once the treatment process is complete (Ganiyu et. al., 2018; Poza-Nogueiras et. al., 2018).

2.1. Effective Parameters

2.1.1. Temperature (°C)

Temperature is one of the key parameters affecting the rate of conversion of H_2O_2 in the reaction medium and, consequently, the required treatment time. An increase in temperature may contribute to faster treatment times due to the acceleration of reaction kinetics. However, raising the temperature above approximately 40–50 °C can adversely affect the usability of H_2O_2 in the targeted oxidation reactions. The main reason this is that at high temperatures, H_2O_2 tends to decompose into water and oxygen, thereby reducing the amount of H_2O_2 available for oxidation. For this reason, it is generally preferable to maintain the process temperature within the 20–40 °C range (Deliktaş, 2011).

2.1.2. pH

One of the key operating variables determining the oxidative removal of pollutants is the pH of the medium. The efficiency of the process is higher than that of other advanced oxidation methods, particularly under acidic conditions. Studies conducted in the literature on various industrial wastewaters report that the most suitable pH value for the Fenton reaction is generally around 3, and that the range of 2.0–4.0 generally provides suitable operating conditions (Deliktaş, 2011; Neyens and Baeyens, 2003). However, a pH value below or above this range may adversely affect process efficiency. Under very low pH conditions, the reduced activity of Fe^{2+} species in the formation of $\bullet\text{OH}$ leads to a limitation in radical production. On the other hand, when the pH rises above 4, the formation of different hydroxy complexes by iron ions and the decrease in the amount of soluble Fe^{2+} result in a reduction in the degradation rate of pollutants.

2.1.3. Initial H₂O₂ Dose

The formation of •OH radicals, which plays a role in the oxidation process in the electro-Fenton process, is largely dependent on the participation of H₂O₂ in the reaction. For this reason, the amount of H₂O₂ in the electrochemical system is one of the key operating parameters that directly affects process performance. Increasing the H₂O₂ concentration up to a certain level can accelerate the oxidative degradation of pollutants due to the rise in the amount of •OH radicals formed in the system. However, using more H₂O₂ than necessary does not always result in an increase in treatment efficiency. At high H₂O₂ concentrations, the reaction of hydrogen peroxide with •OH radicals, thereby consuming the radicals, can limit the effectiveness of the oxidation process. Furthermore, as the use of excessive amounts of H₂O₂ may lead to an increase in the COD values measured after treatment, it is important to determine the appropriate H₂O₂ concentration for the process (Deliktaş, 2011).

2.1.4. Current Magnitude

The current applied in the electro-Fenton process is a key operating parameter in terms of controlling both the electrochemical reactions and the release of Fe²⁺ from the electrodes. Increasing the current up to a certain level promotes the formation of Fe²⁺ and, consequently, the oxidation of pollutants. However, at high current values, although energy consumption increases, the improvement in removal efficiency may remain limited. Therefore, it is necessary to determine the optimum current value to achieve an appropriate balance between process performance and energy consumption (Deliktaş, 2011).

2.1.5. Initial Na₂SO₄ Dosage

In the electro-Fenton process, if the solution does not possess sufficient electrical conductivity, this can adversely affect the efficiency of the electrochemical reactions and, consequently, the performance of the process. In such cases, it is common practice to add a supporting electrolyte to the system in order to increase the solution's ionic conductivity. Supporting electrolytes enable a reduction in electrical resistance and more efficient charge transfer between the electrodes by increasing the mobility of ions in the solution. One of the most commonly used electrolytes for this purpose in EF applications is Na₂SO₄ (Nidheesh and Gandhimathi, 2012). In addition, various other electrolytes such as NaCl, KCl, NaClO₄ and NaNO₃ can also be utilised. However, the effect of the electrolyte on the process is not limited solely to increasing the conductivity of the solution. The ions formed upon dissolution of the electrolyte can participate in, or alter the course of, the electrochemical reactions taking place on the electrode surface and within the solution. Consequently, the type and concentration of the electrolyte used may have a direct or indirect effect on the level of contaminant removal and the degradation kinetics (Moreira et al., 2017).

2.1.6. Fe²⁺ Ion Concentration

In the electro-Fenton process, the Fe²⁺ concentration is one of the key parameters that directly affects process efficiency, as it catalyses the conversion

of H_2O_2 into $\bullet\text{OH}$. Whilst increasing the concentration of iron ions up to a certain level improves the reaction rate and contaminant removal, an excess of Fe^{2+} beyond the optimum level provides no additional benefit and may lead to an increase in residual iron in the treated water. In heterogeneous EF systems, the catalyst amount is similarly critical; an increase in catalyst dosage can lead to an increase in active surface area and radical formation. However, excessive catalyst use can adversely affect current transfer and mass transport, thereby reducing process performance (Iglesias et. al., 2014; Sun et. al., 2015).

2.2. Literature Review

In recent years, the homogeneous EF process has continued to be investigated, particularly for the removal of hard-to-degrade organic pollutants and pharmaceutical compounds. Current studies show that research is focused not only on increasing pollutant removal efficiency, but also on practical aspects of the process, such as reactor design, continuous operation, process optimisation and the reduction of energy consumption. For example, the homogeneous EF process was investigated for the treatment of effluent containing phenazopyridine in a 3D-printed filter-press flow cell capable of continuous operation, and process conditions were optimised using Central Composite Design (CCD) and the Response Surface Methodology (RSM). In the study, the optimum conditions were determined to be approximately pH 3, 0.72 mg/L Fe^{2+} and 5 V, resulting in a phenazopyridine removal rate of 75.14 per cent. The researchers noted that this approach holds potential for the development of continuous-flow and scalable EF systems (Gharibian et al., 2025). Similarly, in another study, the performance of a low-energy-consumption homogeneous Electro-Fenton process in the removal of a xanthene-type dye was evaluated using RSM. The effects of the applied current, Fe^{2+} concentration and mixing rate on the process were investigated. The results of the study demonstrated that the statistical optimisation of process conditions is crucial for controlling both contaminant removal efficiency and energy consumption (Kacem et al., 2024). Furthermore, a comprehensive review assessed studies conducted between 2020 and 2024 on the treatment of real wastewater using EF, and reported that the homogeneous EF process, when combined with suitable electrode systems, can achieve high organic matter removal and, in some cases, high mineralisation (Brillas and Garcia-Segura, 2025). However, the need for an acidic operating pH (approximately pH 3) due to the presence of Fe^{2+} / Fe^{3+} ions in solution, the formation of iron hydroxide sludge and the requirement for pH adjustment following treatment remain significant limitations. Consequently, current research is focusing on enhancing the continuous operability of the reactor, improving Fe^{2+} regeneration, reducing energy consumption and developing the process's scalability to real wastewater streams, whilst maintaining the high oxidation capacity of the homogeneous EF process. However, fundamental disadvantages such as the narrow operating pH range and iron sludge formation mean that research into heterogeneous EF systems (Figure 1), which can operate over a wider pH range and ensure the catalyst is retained in the solid phase, remains important (Shokri et al., 2023; Brillas and Garcia-Segura, 2025).

2.3. Industrial Applications

In addition to the dyes used in textile production, surfactants and various process auxiliary chemicals result in the composition of the resulting wastewater being highly complex and variable. Among the characteristic properties of this wastewater, the particularly high organic load and intense colour formation are noteworthy. Traditional physical-chemical methods, such as biological treatment, coagulation-flocculation and activated carbon adsorption, may not achieve sufficient removal due to the high water solubility and low molecular weight of certain textile dyes. Furthermore, as these processes typically involve the transfer of pollutants to another phase than their complete degradation, the resulting secondary waste must be managed separately. These limitations have interest in advanced oxidation processes for the treatment of textile effluents. $\bullet\text{OH}$ generated during the process react rapidly and non-selectively with organic compounds, causing them to be converted into more advanced oxidation products (Lin and Peng, 1995). Another industrial sector in which the process is applied is paper production. The high production volume of the paper industry results in the generation of significant quantities of wastewater and, consequently, an increase in the treatment load. It has been reported that the oxidation of paper industry effluent, which has undergone biological pre-treatment, using the Fenton process achieved 83% COD and 95% colour removal (Pérez et. al., 2002). Similarly, the olive oil industry, which has a seasonal production pattern, also generates effluents containing high levels of organic pollution. The high pollution load of these effluents complicates disposal and treatment processes. The applicability of the Fenton process to olive oil industry effluents and the associated process costs have been investigated in various studies (Rivas et. al., 2001). In studies where the electro-Fenton process was used for pre-treatment, a 68 percent reduction in COD was reported (Lunar et. al., 2000; Rivas et. al., 2003).

One of the applications of the Fenton process is the microbial removal in water and wastewater systems. Research into alternative oxidation methods is ongoing due to the formation of toxic by-products and the environmental impacts of chlorine, which is widely used in conventional disinfection processes. The hydroxyl radicals formed during the Fenton process can damage the cellular structure of microorganisms, leading to the release of cellular components into the external environment and the disruption of the cell surface's integrity. This characteristic demonstrates that the Fenton process can be utilised not only for the oxidation of organic pollutants but also for disinfection purposes (Diao et. al., 2004).

3. CONCLUSION

Among electrochemical advanced oxidation processes, the EF method is regarded as an important treatment alternative due to its high oxidation capacity for the removal of organic pollutants. The efficiency of the process depends on various operational and system parameters, such as the pH of the reaction medium, the applied current, the concentrations of Fe^{2+} and H_2O_2 , the properties of the cathode material, and the electrical conductivity of the solution. The

appropriate control of these variables is crucial for improving both hydroxyl radical formation and contaminant removal.

It has been observed that EF processes can yield effective results, particularly in the treatment of various industrial wastewaters such as those from the textile, paper and olive oil industries. However, factors such as the requirement for a low pH, the consumption of H_2O_2 through side reactions, and the formation of residues due to excessive iron use may limit the process's applicability. Furthermore, whilst increasing the applied current in the Electro-Fenton process enhances treatment performance up to a certain level, higher current conditions beyond the optimum value can lead to increased energy consumption, which may present an economic disadvantage.

Consequently, in order for the EF process to be implemented as an effective and economical treatment technology, energy consumption, chemical usage and operating costs must be assessed in conjunction with high removal efficiency. In particular, the development of cost-effective, high-performance electrode materials, the determination of optimum operating conditions and a more detailed elucidation of the process mechanism are important for adapting the method to different wastewater characteristics. Whilst current studies demonstrate that the EF process has significant potential for the removal of various organic pollutants, further research into energy efficiency and long-term operational performance is required to enable the method to be scaled up for wider-scale applications.

REFERENCES

- Babuponnusami, A., & Muthukumar, K. (2014). A review on Fenton and improvements to the Fenton process for wastewater treatment. *Journal of Environmental Chemical Engineering*, 2(1), 557-572.
- Barrera-Díaz, C., Urena-Nunez, F., Campos, E., Palomar-Pardavé, M., & Romero-Romo, M. (2003). A combined electrochemical-irradiation treatment of highly colored and polluted industrial wastewater. *Radiation physics and chemistry*, 67(5), 657-663.
- Brillas, E., & Garcia-Segura, S. (2020). Benchmarking recent advances and innovative technology approaches of Fenton, photo-Fenton, electro-Fenton, and related processes: A review on the relevance of phenol as model molecule. *Separation and Purification Technology*, 237, 116337.
- Brillas, E., & Garcia-Segura, S. (2025). Electrochemical advanced oxidation processes for real wastewater remediation: systems, performance, and generated oxidants. *ACS Electrochemistry*, 1(11), 2292-2316.
- Deliktaş, E. (2011). Kağıt endüstrisi atıksularının elektro-fenton prosesi ile arıtılması. Selçuk Üniversitesi Fen Bilimleri Enstitüsü.
- Diao, H. F., Li, X. Y., Gu, J. D., Shi, H. C., & Xie, Z. M. (2004). Electron microscopic investigation of the bactericidal action of electrochemical disinfection in comparison with chlorination, ozonation and Fenton reaction. *Process biochemistry*, 39(11), 1421-1426.
- Do, T. M., Byun, J. Y., Kim, S. H. (2017). An electro-Fenton system using magnetite coated metallic foams as cathode for dye degradation. *Catalysis Today*, 295, 48-55. <https://doi.org/10.1016/j.cattod.2017.05.016>.
- Fenton, H. J. H. (1894). LXXIII.—Oxidation of tartaric acid in presence of iron. *Journal of the Chemical Society, Transactions*, 65, 899-910.
- Ganiyu, S. O., Zhou, M., & Martínez-Huitle, C. A. (2018). Heterogeneous electro-Fenton and photoelectro-Fenton processes: a critical review of fundamental principles and application for water/wastewater treatment. *Applied Catalysis B: Environmental*, 235, 103-129.
- Garcia-Segura, S., Ocon, J. D., & Chong, M. N. (2018). Electrochemical oxidation remediation of real wastewater effluents—A review. *Process Safety and Environmental Protection*, 113, 48-67.
- Gharibian, S., Sajedi, F., Hazrati, H., & Rostamizadeh, M. (2025). Homogenous Electro-Fenton degradation of phenazopyridine in wastewater using a 3D Printed filter-press flowcell: Optimization via response surface methodology. *Journal of the Taiwan Institute of Chemical Engineers*, 170, 106026.

- Iglesias, O., Gómez, J., Pazos, M., & Sanromán, M. Á. (2014). Electro-Fenton oxidation of imidacloprid by Fe alginate gel beads. *Applied Catalysis B: Environmental*, 144, 416-424.
- Kacem, S. B., Clematis, D., Elaoud, S. C., & Panizza, M. (2024). Response surface methodology for low-energy consumption electro-Fenton process for xanthene dye electrochemical degradation. *Journal of Applied Electrochemistry*, 54(9), 2095-2110.
- Kurian, M. (2021). Advanced oxidation processes and nanomaterials-a review. *Cleaner Engineering and Technology*, 2, 100090.
- Liu, H., Li, X. Z., Leng, Y. J., & Wang, C. (2007). Kinetic modeling of electro-Fenton reaction in aqueous solution. *Water Research*, 41(5), 1161-1167.
- Lin, S. H., & Peng, C. F. (1995). Treatment of textile wastewater by Fenton's reagent. *Journal of Environmental Science & Health Part A*, 30(1), 89-98.
- Lin, S. H., & Lo, C. C. (1997). Fenton process for treatment of desizing wastewater. *Water research*, 31(8), 2050-2056.
- Lunar, L., Sicilia, D., Rubio, S., Pérez-Bendito, D., & Nickel, U. (2000). Degradation of photographic developers by Fenton's reagent: condition optimization and kinetics for metol oxidation. *Water Research*, 34(6), 1791-1802.
- L. Kremer, M. (1999). Mechanism of the Fenton reaction. Evidence for a new intermediate. *Physical Chemistry Chemical Physics*, 1(15), 3595-3605.
- Moreira, F. C., Garcia-Segura, S., Boaventura, R. A., Brillas, E., & Vilar, V. J. (2014). Degradation of the antibiotic trimethoprim by electrochemical advanced oxidation processes using a carbon-PTFE air-diffusion cathode and a boron-doped diamond or platinum anode. *Applied Catalysis B: Environmental*, 160, 492-505.
- Moreira, F. C., Boaventura, R. A., Brillas, E., & Vilar, V. J. (2017). Electrochemical advanced oxidation processes: a review on their application to synthetic and real wastewaters. *Applied Catalysis B: Environmental*, 202, 217-261.
- Mousset, E., Wang, Z., Hammaker, J., & Lefebvre, O. (2017). Electrocatalytic phenol degradation by a novel nanostructured carbon fiber brush cathode coated with graphene ink. *Electrochimica Acta*, 258, 607-617.
- Neyens, E., & Baeyens, J. (2003). A review of classic Fenton's peroxidation as an advanced oxidation technique. *Journal of Hazardous materials*, 98(1-3), 33-50.
- Nidheesh, P. V., & Gandhimathi, R. (2012). Trends in electro-Fenton process for water and wastewater treatment: an overview. *Desalination*, 299, 1-15.
- Pérez, M., Torrades, F., García-Hortal, J. A., Domènech, X., & Peral, J. (2002). Removal of organic contaminants in paper pulp treatment effluents under

- Fenton and photo-Fenton conditions. *Applied Catalysis B: Environmental*, 36(1), 63-74.
- Poza-Nogueiras, V., Rosales, E., Pazos, M., & Sanroman, M. A. (2018). Current advances and trends in electro-Fenton process using heterogeneous catalysts—a review. *Chemosphere*, 201, 399-416.
- Rivas, F. J., Beltrán, F. J., Gimeno, O., & Frades, J. (2001). Treatment of olive oil mill wastewater by Fenton's reagent. *Journal of agricultural and food chemistry*, 49(4), 1873-1880.
- Rivas, F. J., Beltrán, F. J., Gimeno, O., & Alvarez, P. (2003). Optimisation of Fenton's reagent usage as a pre-treatment for fermentation brines. *Journal of hazardous materials*, 96(2-3), 277-290.
- Shokri, A., Nasernejad, B., & Sanavi Fard, M. (2023). Challenges and future roadmaps in heterogeneous electro-fenton process for wastewater treatment. *Water, Air, & Soil Pollution*, 234(3), 153.
- Sun, H., Chen, T., Kong, L., Cai, Q., Xiong, Y., & Tian, S. (2015). Potential of sludge carbon as new granular electrodes for degradation of acid orange 7. *Industrial & Engineering Chemistry Research*, 54(20), 5468-5474.
- Villegas, L. G. C., Mashhadi, N., Chen, M., Mukherjee, D., Taylor, K. E., & Biswas, N. (2016). A short review of techniques for phenol removal from wastewater. *Current Pollution Reports*, 2(3), 157-167.
- Wang, J., & Zhuan, R. (2020). Degradation of antibiotics by advanced oxidation processes: An overview. *Science of the Total Environment*, 701, 135023.

CHAPTER 5

**Quantum k-Nearest Neighbors for
Photocatalytic Hydrogen Production
Prediction: Encoding
Strategies and Robustness Analysis**

Rezan Bakır¹

¹ Department of Computer Engineering, Faculty of Engineering and Natural Sciences,
Sivas University of Science and Technology, Sivas, Turkey,
ORCID: 0000-0002-4373-2231

1 Introduction

Photocatalytic hydrogen production brings together catalyst development, reaction engineering, and predictive modeling. Process performance depends on the material system and on operating conditions, so a useful predictor must be interpreted within the experimental domain represented by its data. Work on graphene-supported lanthanum ferrite and on related solar-driven hybrid catalysts provides the experimental context for the present chapter (Orak & Yüksel, 2021; Orak & Yüksel, 2022b). Hydrogen generation in the case study uses sucrose as a sacrificial agent; its results should therefore be understood as specific to that photocatalytic reaction system rather than as a general assessment of overall water splitting.

Materials research also connects hydrogen evolution with environmental applications. Graphitic carbon nitride and carbon-supported composites have been studied for adsorption and photocatalysis (Jeon, Lee, & Baek, 2017; Xu et al., 2022), while layered double hydroxides have been investigated for pollutant degradation and pharmaceutical wastewater treatment (Bharali et al., 2023; Chen, Xu, Khan, Han, & Zhu, 2025). Carbon nanomaterials have a broader role in electrochemical energy systems (Yang et al., 2019). Hybrid photocatalysts can combine hydrogen evolution with pollutant removal, and photocatalytic fuel cells link wastewater treatment with energy generation (Orak & Ersöz, 2025; Sekar et al., 2021). These adjacent applications help explain interest in data-driven prediction, although their results cannot be transferred directly to the GLFO dataset examined here.

Classical machine learning has been applied to hydrogen prediction using stacked and ensemble regressors (Bakır & Orak, 2024; Bakır, Orak, & Yüksel, 2024a; Bakır, Orak, & Yüksel, 2024b). Experimental and predictive approaches have also been integrated for Cu/g-C₃N₄ and Fe/g-C₃N₄ systems (Arabacı, Bakır, Orak, & Yüksel, 2024; Arabacı, Bakır, Orak, & Yüksel, 2025). Response-surface designs offer a complementary approach to organizing photocatalytic experiments (Orak & Yüksel, 2022a). Related machine learning applications address photocatalytic pollutant degradation (Bakır, Orak, & Horoz, 2025). Together, these studies motivate models that connect recorded process conditions with measured responses while retaining a clear account of the experimental scope.

Predictive modeling extends beyond operational descriptors to materials discovery. Li et al. (2021) combined machine learning with high-throughput experimentation on photocatalytically active organic molecules. Tang, Liu, Zhou, and Sun (2025) investigated deep-learning-enabled screening of MXene photocatalysts, and G et al. (2024) examined deep learning for photochemical hydrogen production. These tasks differ from the three-variable process

regression considered in this chapter, so their performance results are not direct benchmarks for the present model.

The review by Araujo and Farrusseng (2025) discusses artificial intelligence in photocatalytic hydrogen production. Wayo, Goliatt, and Ganji (2025) review the integration of density functional theory and hybrid classical–quantum machine learning for photocatalyst discovery and hydrogen production. Its discussion provides a broader research perspective; it should not be interpreted as a controlled demonstration that the QkNN implementation used here delivers a quantum advantage.

Quantum feature maps offer another way to define sample similarity by mapping classical inputs into quantum states (Havlíček et al., 2019; Schuld & Killoran, 2019). A neighbor regressor can then select training samples through state fidelity instead of Euclidean distance. Quantum nearest-neighbor algorithms have also been studied from a computational perspective (Wiebe, Kapoor, & Svore, 2015). However, a quantum-defined similarity does not by itself establish faster computation or better prediction. Such claims depend on the feature map, data access, implementation, and comparison protocol.

This chapter focuses on three questions: how angle and amplitude encoding affect hydrogen-yield prediction, how prediction changes under disturbed test inputs, and how the reported results compare with a classical neighbor baseline. The contribution is an empirical case study and an explanation of the underlying similarity geometry. The chapter first introduces the regression method, then describes the dataset and evaluation, presents the reported results, and discusses the scope and limitations of the evidence.

2. Quantum Computing Background and QkNN

Quantum computing is a rapidly evolving field that leverages the principles of quantum mechanics—such as superposition and entanglement—to perform computations in fundamentally different ways compared to classical computers. In a classical bit-based system, information is processed using binary states (0 or 1). In contrast, quantum bits (qubits) can exist in linear combinations of these states, enabling exponential growth in representational power as the number of qubits increases.

One key advantage of quantum computing lies in its potential to represent and process high-dimensional data spaces more efficiently. Quantum algorithms have demonstrated theoretical benefits in search, optimization, and certain linear algebra operations, which are crucial to modern machine learning tasks.

Quantum Machine Learning (QML) combines quantum computing with classical machine learning to design models that can potentially outperform classical algorithms, especially in terms of computational complexity and kernel representation. QML models rely on encoding classical data into quantum states,

processing it through quantum circuits, and extracting useful information through measurement.

2.1. Angle Encoding

Angle encoding (also known as rotation encoding) maps each classical input feature to the rotation angle of a quantum gate. Specifically, the features are embedded into quantum states via single-qubit Y-axis rotations. For an input vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$, each $\mathbf{x}_i$ is used to parameterize a rotation gate $R_y(\mathbf{x}_i)$ acting on the corresponding qubit. This method is particularly well-suited for low-dimensional datasets, offering an interpretable and hardware-efficient quantum representation. Its circuit depth scales linearly with the number of features, making it favorable for near-term quantum devices.

2.2. Amplitude Encoding

Amplitude encoding offers a more compact representation by embedding all feature values simultaneously into the amplitude vector of a single quantum state. For an n -dimensional input vector $\mathbf{x}$, the data must be normalized such that $\sum_{i=1}^n |\mathbf{x}_i|^2 = 1$, after which it is encoded into the amplitudes of a quantum state $|\psi_{\mathbf{x}}\rangle = \sum_{i=1}^n \mathbf{x}_i |i\rangle$. This method allows for an exponentially efficient representation of data in quantum memory but requires a circuit depth that grows with input size and may necessitate zero-padding when the input dimension is not a power of two. In this work, amplitude encoding was implemented using the Mottonen State Preparation template from the PennyLane library. The comparative analysis between these two encodings provides insight into their respective advantages and trade-offs, particularly in the context of small-scale, real-world datasets such as those used in hydrogen production modeling. The impact of encoding choice on predictive accuracy and robustness is further explored in the experimental results section. This quantum similarity is used as a distance metric to identify the closest training samples to a given test input. By leveraging quantum fidelity and related angular metrics, the QkNN approach provides a novel framework for data-driven prediction, particularly suitable for problems like hydrogen production modeling, where the dataset is small, and kernel-based relationships are important. All quantum circuits were developed using the PennyLane quantum computing library and simulated on the default.qubit device. The number of qubits was determined by the input dimensionality (three features), with additional qubits used as needed for amplitude encoding. Quantum nodes (QNodes) were constructed to compute fidelity between input state vectors, and circuits were reset for each test–train pairwise comparison.

2.3. Similarity measure and distance choice for QkNN

This section describes the comparison of encoded samples in Hilbert space using quantum fidelity and two fidelity-derived distance measures. For two pure states prepared from input vectors ($\mathbf{x}$) and ($\mathbf{z}$), fidelity is defined as the squared

magnitude of their inner product: $F(x, z) = |\langle \varphi(x) | \varphi(z) \rangle|^2$. Fidelity takes values in $[0, 1]$ and equals 1 only when the states are identical. In practice, we estimate F with an overlap circuit: (i) prepare $|\varphi(x)\rangle$ from $|0\dots 0\rangle$, (ii) apply the adjoint of the encoding for z , and (iii) measure the probability of the all-zeros outcome; this probability equals F . (A SWAP-test circuit yields the same quantity and can be used interchangeably on suitable hardware.)

For nearest-neighbor search, two distance measures derived from fidelity are considered. The first is a linear transformation of fidelity

$$d_F(x, z) = 1 - F(x, z) \quad (1)$$

It is monotone, simple to compute, and effective in k -nearest neighbors, although it is not a strict mathematical metric. The second is the angular (Fubini–Study) distance

$$d_\theta(x, z) = \arccos\left(\sqrt{F(x, z)}\right), \quad (2)$$

which corresponds to the geodesic angle between the two pure states and lies in $[0, \pi/2]$.

Because the $\arccos(\cdot)$ transform expands values near $F \approx 1$, d_θ provides finer resolution among highly similar encodings—a useful property for low-dimensional inputs and mild perturbations (e.g., Gaussian noise or small adversarial ϵ).

Within QkNN, distances from a query x^* to all training encodings are computed using either d_θ or d_F . The k smallest values define the neighborhood, and the prediction is formed via optional inverse-distance weighting:

$$\hat{y}(x^*) = (\sum w_i y_i) / (\sum w_i) \quad (3)$$

with $w_i = 1/(d(x^*, x_i) + \epsilon)$ and a small $\epsilon > 0$ to avoid division by zero. To ensure a fair comparison with classical kNN, we use identical folds, the same k , and the same train/test partitions.

To prevent leakage, fidelities (and thus distances) are recomputed independently for every cross-validation fold and each perturbation setting. Pairwise evaluations are batched for efficiency, circuits are re-initialized before each comparison, and all simulations are executed on a state-vector backend (e.g., PennyLane’s default.qubit).

2.4 Mathematical Formulation of Quantum k-Nearest Neighbors (QkNN)

Let the input dataset be:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \quad (4)$$

where $x_i \in R^d$ represents a feature vector (e.g., pH, catalyst amount, H_2O_2) and $y_i \in R$ is the corresponding target variable (hydrogen production amount).

1. Quantum Encoding

Each input x is feature-scaled to $[0,1]$ and mapped to a d -qubit state by a state-preparation circuit $U(x)$. With angle (rotation) encoding,

$$|\psi(x)\rangle = U(x)|0\rangle^{\otimes d} \quad (5)$$

Where $U(x) = \bigotimes_{j=1}^d R_Y(x_j)$, and $R_Y(x_j)$ is the rotation around the Y-axis for feature x_j .

2. Quantum similarity and distances
Given a test point x_{ts} and a training point x_i , define the pure-state fidelity

$$F(x_{ts}, x_i) = |\langle \psi(x_{ts}) | \psi(x_i) \rangle|^2 \in [0,1] \quad (6)$$

From F we derive two distances used in QkNN:

-Linear fidelity distance :

$$d_F(x_{ts}, x_i) = 1 - F(x_{ts}, x_i) \quad (7)$$

- Angular (Fubini–Study) distance:

$$d_\theta(x_{ts}, x_i) = \arccos(\sqrt{F(x_{ts}, x_i)}) \in [0, \pi/2] \quad (8)$$

Either d_F or d_θ can serve as the neighborhood metric; d_θ offers finer discrimination when F is close to 1.

3. Neighbor selection

For a given test point x_{ts} , compute its quantum distance $D(x_{ts}, x_i)$ to all training samples using the chosen metric ($D = d_F$ or d_θ), then select the index set of the k smallest distances.

$$\mathcal{N}_k(x_{ts}) = \text{argsort}_i(D(x_{ts}, x_i))[:k] \quad (9)$$

4. Regression rule

Given the neighbor set $N_k(x_{ts})$, the QkNN prediction is an (optionally) inverse-distance-weighted average: The QkNN regression output is obtained by averaging the target values of the k nearest neighbors:

$$\hat{y}(x_{ts}) = (\sum_{i \in N_k(x_{ts})} w_i y_i) / (\sum_{i \in N_k(x_{ts})} w_i) \quad (10)$$

with weights $w_i = 1/(D(x, x_i) + \varepsilon)$ and a small $\varepsilon > 0$ to avoid division by zero. The unweighted k -NN estimate is recovered by setting $w_i = 1$ for all $i \in N_k(x_{ts})$.

3 Hydrogen Prediction Case Study

3.1 Dataset and experimental scope

The case study reuses the published dataset of Bakır, Orak, and Bakır (2026), which documents hydrogen evolution using graphene-supported lanthanum ferrite catalysts and sucrose as a sacrificial agent. The dataset contains 1,370 records, each with three input variables: pH, catalyst amount in g/L, and initial hydrogen peroxide concentration in mM. The output is hydrogen yield in mmol/gcat. The experimental design and collection procedures are described in the dataset article, which should be consulted alongside this chapter.

3.2 Preprocessing and implementation

All input features were normalized using the MinMaxScaler to fit within the $[0, 1]$ range, facilitating quantum circuit compatibility and numerical stability. The target variable was left in its original scale. The dataset was then partitioned into training and test sets using an 80-20 split, and later adapted to k -fold cross-validation ($k=5$) to assess model generalization more robustly. All simulations were performed on a high-performance system equipped with NVIDIA RTX 3060 GPU, using PennyLane (v0.35) and Python 3.10. A 5-fold cross-validation scheme was employed to assess the generalization performance of the QkNN model. Classical kNN with Euclidean distance and identical $k=3$ was used as a baseline. The model was evaluated under increasing levels of Gaussian noise (std = 0.01, 0.05, 0.1) and FGSM-style adversarial perturbations ($\varepsilon = 0.05$).

3.3 Assessing Model Stability Under Gaussian Noise and FGSM Adversarial Attacks

This section examines the robustness of the QkNN model under two input perturbation scenarios: Gaussian noise injection and adversarial perturbations generated using the Fast Gradient Sign Method (FGSM). These evaluations complement the assessment on clean test data by examining how predictive performance changes when input measurements are perturbed. Gaussian noise represents random measurement variability, whereas FGSM introduces targeted

perturbations intended to degrade model predictions. Together, these tests help assess model stability and potential reliability in practical applications.

Gaussian Noise Injection:

The first robustness experiment involved introducing additive Gaussian noise to the normalized feature set during inference. This zero-mean noise was applied with increasing standard deviations ($\sigma = 0.01, 0.05, \text{ and } 0.10$) to reflect varying degrees of sensor imprecision or environmental disturbances. The purpose of this test was to observe how well the QkNN model could maintain predictive stability when exposed to unseen data distortions. It is important to note that no retraining or fine-tuning was performed after noise injection. The model's outputs were directly evaluated on the perturbed test samples, enabling an assessment of generalization under degraded conditions. Such an approach reflects real-world scenarios where retraining is impractical or unavailable during inference time.

FGSM Adversarial Attack:

Adversarial attacks have become a critical concern in machine learning, particularly in applications where reliability and safety are paramount. These attacks involve introducing small, carefully crafted perturbations to input data, which may be imperceptible to humans but can significantly alter a model's output. While initially studied in the context of deep neural networks [22], [23], adversarial vulnerabilities have also been observed in traditional models such as support vector machines [24] and even non-parametric methods like k-nearest neighbors [25].

Classical kNN is generally considered robust to noise due to its simplicity and local decision-making. However, targeted perturbations can mislead its neighbor selection process, leading to degraded performance. These perturbations exploit the lack of global generalization and can shift test instances toward misleading neighborhoods, causing incorrect regression or classification predictions.

The concept of adversarial attacks has recently extended into the realm of quantum machine learning (QML). Quantum models encode classical inputs into quantum states, which are then processed using unitary operations. While this offers richer representational power, it also introduces new vulnerabilities. Adversarial noise in QML can propagate through the quantum feature space, potentially altering measurement outcomes or fidelity-based similarities used in distance-based models like QkNN [26]. Notably, QkNN lacks a differentiable structure and does not learn parameters, which makes traditional gradient-based attacks inapplicable. However, adversarial robustness can still be studied by generating perturbations via surrogate models (e.g., neural networks) and evaluating the effect of these crafted inputs on the QkNN prediction pipeline. This black-box attack simulation helps assess the QkNN model's stability under worst-case input scenarios. This chapter assesses the resilience of QkNN using

surrogate-based adversarial perturbations generated through the Fast Gradient Sign Method (FGSM). FGSM uses the gradient of a loss function with respect to the input to construct perturbations intended to increase prediction error. Because the discrete nearest-neighbor selection in QkNN does not readily support direct gradient-based attacks, a differentiable surrogate model—a shallow neural network—was trained on the same training folds. Input gradients were then computed from the surrogate model’s loss. For each input sample, with features scaled to $[0, 1]$, an adversarial example was generated under an (L_∞) constraint with $(\epsilon = 0.05)$, as defined in Equation (11). The resulting examples were used to evaluate the robustness of QkNN to transferred adversarial perturbations.

$$x_{adv} = \text{clip}(x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(g(x), y)), 0, 1), \quad (11)$$

where g is the surrogate and $\mathcal{L}$ is the training loss. The perturbed inputs were then passed—without any retraining—through the original QkNN pipeline (angle encoding; fidelity/angular distance) to test whether the quantum distance metric sustains targeted perturbations designed to mislead the model.

4. Predictive Performance and Robustness Analysis

This section presents and discusses the predictive performance and robustness of the QkNN model across three evaluation scenarios: clean inputs, Gaussian noise injection, and surrogate-based FGSM adversarial perturbations. Comparisons with classical kNN provide a benchmark under the same evaluation conditions. Model performance is assessed using mean squared error (MSE), mean absolute error (MAE), and the coefficient of determination (R^2).

4.1. Encoding Comparison

To investigate the impact of different encoding strategies in the quantum feature space, both angle encoding and amplitude encoding were evaluated using the QkNN framework. Due to the dataset’s low dimensionality (three features), angle encoding showed superior performance. Amplitude encoding, which requires padding the input vector to fit the 2^n amplitude vector format, likely diluted meaningful feature information during state preparation. The performance comparison between angle and amplitude encoding strategies, summarized in Table 2, reveals that angle encoding significantly outperforms amplitude encoding across all evaluation metrics. Specifically, angle encoding achieves a lower MSE and MAE, and a substantially higher R^2 score, indicating superior predictive accuracy and robustness in the quantum kNN model.

Table 1. Comparison of Encoding Strategies using Fidelity-Based Distance

Encoding Type	MSE	MAE	R ²
Angle (1 - Fidelity)	0.0094	0.0335	0.9498
Amplitude (1 - Fidelity)	0.0839	0.1501	0.5514

These results confirm that angle encoding is more suitable for compact datasets with few features, as it allows each input variable to influence a qubit directly without requiring complex padding or normalization.

4.2 QkNN Performance on Clean Data

The initial evaluation of the QkNN model was performed on the clean, normalized test dataset to establish a baseline for predictive accuracy. As summarized in Table 3 (row 1), the model attains strong scores on all regression metrics. These results indicate that the QkNN model effectively captures the underlying functional relationships within the experimental hydrogen production dataset. The high R² value signifies a strong correlation between the predicted and actual hydrogen outputs, while the low error values confirm the model's precision in estimating the target variable under unperturbed input conditions.

4.3 Robustness to Gaussian Noise

The model was initially evaluated using clean test data (std = 0.00) and then under increasing Gaussian noise levels: 0.01, 0.05, and 0.10. The results, summarized in Figure 1 and Table 3, show a progressive degradation in performance as noise increases, which aligns with expectations for both classical and quantum models. The QkNN model demonstrated notable robustness at low to moderate noise levels.

Table 2. Performance of Quantum kNN under different Gaussian noise levels

Noise Std	MSE	MAE	R ²
0.00 (clean data)	0.0094	0.0335	0.9498
0.01	0.0137	0.0390	0.9267
0.05	0.0304	0.0838	0.8373
0.10	0.1142	0.2123	0.3890

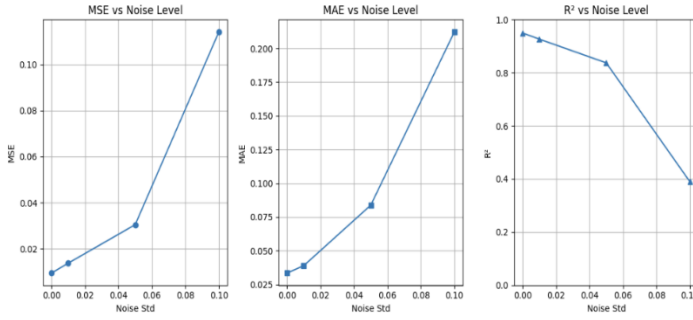


Figure 1. QkNN performance under varying noise levels

4.4 Comparison with Classical kNN

To assess the quantum advantage, QkNN results were compared against classical kNN using the same data splits, number of neighbors, and noise settings. As shown in Figure 2 and Table 4 both models exhibit similar trends; however, QkNN slightly underperforms classical kNN at higher noise levels, though it remains competitive.

Table 3. Comparison of QkNN and classical kNN under increasing Gaussian noise levels

Noise Std	Model	MSE	MAE	R ²
0.00	Classical KNN	0.0090	0.0331	0.9519
	QKNN	0.0094	0.0335	0.9498
0.01	Classical KNN	0.0130	0.0377	0.9304
	QKNN	0.0137	0.0390	0.9267
0.05	Classical KNN	0.0280	0.0808	0.8502
	QKNN	0.0304	0.0838	0.8373
0.10	Classical KNN	0.1130	0.2030	0.3959
	QKNN	0.1142	0.2123	0.3890

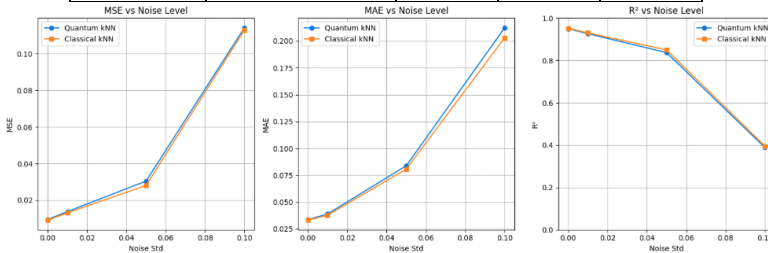


Figure 2. Side-by-Side Comparison of QkNN vs Classical kNN

4.5 Performance Under FGSM Adversarial Attack

Beyond the stochastic perturbations introduced by Gaussian noise, the model’s resilience was further tested using an FGSM adversarial attack with a perturbation strength of $\epsilon = 0.05$. As presented in Table 5, the adversarial setting led to a slightly greater degradation in performance compared to the Gaussian noise scenario at $\sigma = 0.05$. This behavior aligns with expectations, as adversarial attacks are specifically crafted to exploit model sensitivities and induce maximal disruption.

Table 4. QkNN performance under FGSM attack

Perturbation Type	MSE	MAE	R ²
FGSM ($\epsilon = 0.05$)	0.0369	0.0852	0.8025

Despite this, the model retained an R² above 0.80, confirming its general robustness even under adversarial threats.

5. Interpretation and Practical Limits

5.1 What the encoding comparison establishes

The case study shows that a choice made before neighbor selection can materially change regression accuracy. Angle and amplitude representations induce different relationships among the same three input descriptors. This makes encoding a modeling assumption that should be assessed against data characteristics, rather than selected solely to minimize the number of qubits.

The strongest empirical finding is the difference between the two encodings on the reported split. The result is specific to the evaluated feature scaling, circuit definitions, neighborhood size, and dataset. A useful extension would compare these representations with classical kernels that reproduce or approximate their similarity geometry. Such comparisons would help separate the value of a similarity function from the hardware used to evaluate it.

5.2 Conclusion, Limitations and Future Directions

This chapter has examined Quantum k-Nearest Neighbors regression for predicting hydrogen production efficiency from key photocatalytic process parameters. Angle and amplitude encoding strategies were evaluated using quantum fidelity to quantify similarity between encoded samples. Two fidelity-derived distance measures—linear fidelity distance and quantum angular distance—were compared to assess their influence on predictive accuracy. The results demonstrate that angle encoding combined with angular distance provides the strongest performance among the evaluated QkNN configurations, achieving an R² of 0.9498 on clean data. Comparisons with classical kNN show competitive predictive performance and comparable or greater resilience under the tested

perturbation conditions. Although prediction accuracy decreases as Gaussian noise increases, the best-performing configuration retains an R^2 of 0.8152 under the evaluated high-noise condition.

Five-fold cross-validation provides evidence of predictive consistency across the evaluated data partitions. Adversarial testing using surrogate-based FGSM perturbations further examines the model's sensitivity to targeted input disturbances. These findings support the potential of QkNN for robustness-aware regression within the investigated dataset and experimental conditions.

The chapter illustrates how quantum encoding strategies and fidelity-derived distances can support regression in photocatalytic hydrogen modeling and related problems with a small number of input features. Future research could evaluate these approaches on real quantum hardware with error mitigation to quantify differences between simulated and device-based performance. Other directions include richer feature maps incorporating entanglement or data reuploading, adaptive selection of the number of neighbors and distance measure, and integration with classical regression models in hybrid pipelines. Calibrated uncertainty estimates, such as conformal prediction or neighbor-based variance, could further support decision-making. Extending robustness assessments to stronger adversarial attacks and physically motivated perturbations would also help establish the practical scope of these approaches.

Data availability

The dataset analyzed in this chapter is publicly available on Kaggle at <https://www.kaggle.com/datasets/razanghanem/machine-learning-dataset-for-hydrogen-evolution>.

Its experimental design, variables, and collection procedures are documented in the published dataset article by Bakır et al. (2026), A benchmark dataset for machine learning-based prediction of hydrogen evolution efficiency from GLFO-catalyzed sucrose photocatalysis,

<https://doi.org/10.1016/j.dib.2026.112996>.

Acknowledgements

The experimental dataset analyzed in this chapter was generated within the scope of project No. 224M008 supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK).

Declaration of AI assisted writing

AI-assisted tools were used to support language editing and organization. The authors are responsible for reviewing the content, verifying its accuracy, and approving the final chapter.

References

- Akter, M. S., Shahriar, H., Cuzzocrea, A., & Wu, F. (2024). Quantum adversarial attacks: Developing quantum FGSM algorithm. In 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC) (pp. 1073–1079). IEEE.
- Arabacı, B., Bakır, R., Orak, C., & Yüksel, A. (2024). Integrating experimental and machine learning approaches for predictive analysis of photocatalytic hydrogen evolution using Cu/g-C₃N₄. *Renewable Energy*, 237, 121737. doi:10.1016/j.renene.2024.121737
- Arabacı, B., Bakır, R., Orak, C., & Yüksel, A. (2025). Predictive modeling of photocatalytic hydrogen production: Integrating experimental insights with machine learning on Fe/g-C₃N₄ catalysts. *Industrial & Engineering Chemistry Research*, 64(10), 5184–5199. doi:10.1021/acs.iecr.4c03919
- Araujo, L. G., & Farrusseng, D. (2025). Artificial intelligence-driven advances in photocatalytic hydrogen production. *New Journal of Chemistry*, 49(17), 6888–6913. doi:10.1039/D5NJ00505A
- Bakır, R., & Orak, C. (2024). Stacked machine learning approach for predicting evolved hydrogen from sugar industry wastewater. *International Journal of Hydrogen Energy*, 85, 75–87. doi:10.1016/j.ijhydene.2024.08.342
- Bakır, R., Orak, C., & Bakır, H. (2026). A benchmark dataset for machine learning–based prediction of hydrogen evolution efficiency from GLFO-catalyzed sucrose photocatalysis. *Data in Brief*, 67, 112996. doi:10.1016/j.dib.2026.112996
- Bakır, R., Orak, C., & Horoz, S. (2025). Enhancing photocatalytic degradation of hazardous pollutants with green-synthesized catalysts: A machine learning approach. *Journal of Environmental Management*, 385, 125695. doi:10.1016/j.jenvman.2025.125695
- Bakır, R., Orak, C., & Yüksel, A. (2024a). A machine learning ensemble approach for predicting solar-sensitive hybrid photocatalysts on hydrogen evolution. *Physica Scripta*, 99(7), 076015. doi:10.1088/1402-4896/ad562a
- Bakır, R., Orak, C., & Yüksel, A. (2024b). Optimizing hydrogen evolution prediction: A unified approach using random forests, LightGBM, and bagging regressor ensemble model. *International Journal of Hydrogen Energy*, 67, 101–110. doi:10.1016/j.ijhydene.2024.04.173
- Bharali, D., Saikia, S., Devi, R., Choudary, B. M., Gour, N. K., & Deka, R. C. (2023). Photocatalytic degradation of phenol and its derivatives over ZnFe layered double hydroxide. *Journal of Photochemistry and Photobiology A: Chemistry*, 438, 114509. doi:10.1016/j.jphotochem.2022.114509

- Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., ... Roli, F. (2013). Evasion attacks against machine learning at test time. In *Machine learning and knowledge discovery in databases: European conference, ECML pKDD 2013, prague, czech Republic, September 23-27, 2013, proceedings, part III* 13 (pp. 387–402). Springer.
- Chen, Y., Xu, H., Khan, M. S., Han, S., & Zhu, S. (2025). Recent advances in layered double hydroxides for pharmaceutical wastewater treatment: A critical review. *Critical Reviews in Environmental Science and Technology*, 55(14), 1097–1123. doi:10.1080/10643389.2025.2488808
- G, A., Priyadarshini, R., Titus, A., Sahoo, S., Muppala, C., Ramkumar, G., ... Hojjati-Najafabadi, A. (2024). Deep learning for the encounter of inorganic nanomaterial for efficient photochemical hydrogen production. *International Journal of Hydrogen Energy*, 52, 664–673. doi:10.1016/j.ijhydene.2023.05.171
- Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. Retrieved from <https://arxiv.org/abs/1412.6572>
- Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., & Gambetta, J. M. (2019). Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747), 209–212. doi:10.1038/s41586-019-0980-2
- Jeon, P., Lee, M. E., & Baek, K. (2017). Adsorption and photocatalytic activity of biochar with graphitic carbon nitride (g-C₃N₄). *Journal of the Taiwan Institute of Chemical Engineers*, 77, 244–249. doi:10.1016/j.jtice.2017.05.010
- Li, X., Maffettone, P. M., Che, Y., Liu, T., Chen, L., & Cooper, A. I. (2021). Combining machine learning and high-throughput experimentation to discover photocatalytically active organic molecules. *Chemical Science*, 12(32), 10742–10754. doi:10.1039/d1sc02150h
- Möttönen, M., Vartiainen, J. J., Bergholm, V., & Salomaa, M. M. (2005). Transformation of quantum states using uniformly controlled rotations. *Quantum Information and Computation*, 5(6), 467–473. doi:10.26421/qic5.6-5
- Orak, C., & Ersöz, G. (2025). Solar-driven hybrid photocatalytic fuel cells for concurrent wastewater treatment and energy generation. *Journal of Power Sources*, 645, 237198. doi:10.1016/j.jpowsour.2025.237198
- Orak, C., & Yüksel, A. (2021). Graphene-supported LaFeO₃ for photocatalytic hydrogen energy production. *International Journal of Energy Research*, 45(9), 12898–12914. doi:10.1002/er.6620

- Orak, C., & Yüksel, A. (2022a). Box–Behnken design for hydrogen evolution from sugar industry wastewater using solar-driven hybrid catalysts. *ACS Omega*, 7(46), 42489–42498. doi:10.1021/acsomega.2c05721
- Orak, C., & Yüksel, A. (2022b). Comparison of photocatalytic performances of solar-driven hybrid catalysts for hydrogen energy evolution from 1,8-diazabicyclo[5.4.0]undec-7-ene (DBU) solution. *International Journal of Hydrogen Energy*, 47(14), 8841–8857. doi:10.1016/j.ijhydene.2021.12.254
- Papernot, N., & McDaniel, P. (2018). Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. Retrieved from <https://arxiv.org/abs/1803.04765>
- Schuld, M., & Killoran, N. (2019). Quantum machine learning in feature Hilbert spaces. *Physical Review Letters*, 122(4), 040504. doi:10.1103/physrevlett.122.040504
- Sekar, K., Kassam, A., Bai, Y., Coulson, B., Li, W., Douthwaite, R. E., ... Lee, A. F. (2021). Hierarchical bismuth vanadate/reduced graphene oxide composite photocatalyst for hydrogen evolution and bisphenol A degradation. *Applied Materials Today*, 22, 100963. doi:10.1016/j.apmt.2021.100963
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. Retrieved from <https://arxiv.org/abs/1312.6199>
- Tang, K., Liu, S., Zhou, J., & Sun, Z. (2025). Deep-learning-enabled high-throughput screening of MXene photocatalysts for hydrogen production. *Nanoscale*, 17(37), 21717–21728. doi:10.1039/D5NR02764K
- Wayo, D. D. K., Goliatt, L., & Ganji, M. D. (2025). DFT and hybrid classical–quantum machine learning integration for photocatalyst discovery and hydrogen production. *Reviews in Chemical Engineering*, 41(8), 741–774. doi:10.1515/revce-2025-0022
- Wiebe, N., Kapoor, A., & Svore, K. M. (2015). Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *Quantum Information and Computation*, 15(3–4), 316–356. doi:10.26421/qic15.3-4-7
- Xu, H., Zhang, T., Wang, D., Cai, D., Chen, S., Wang, H., ... Zhu, Y. (2022). Degradation of tetracycline using persulfate activated by a honeycomb structured S-doped g-C3N4/biochar under visible light. *Separation and Purification Technology*, 300. doi:10.1016/j.seppur.2022.121833
- Yang, Z., Tian, J., Yin, Z., Cui, C., Qian, W., & Wei, F. (2019). Carbon nanotube- and graphene-based nanomaterials and applications in high-voltage supercapacitor: A review. *Carbon*, 141, 467–480. doi:10.1016/j.carbon.2018.10.010

CHAPTER 6

**Bell-Shaped Transfer
Function-Based Binary
Starfish Optimization Algorithm (SAO_{Bin})**

Emine Baş¹

¹ Assoc. Prof. Dr., Department of Software Engineering, Faculty of Computer and Informatics, Konya Technical University, Konya, Türkiye
ORCID: 0000-0003-4322-6010

1. Introduction

The concept of optimization has gained increasing importance in recent years. The fact that many engineering problems can be formulated and solved as optimization problems has further heightened this significance. Optimization problems can be modeled either as continuous optimization problems or as combinatorial problems—specific to structures involving discrete variables. Optimization problems are categorized based on the relationships between the variables they involve. In this study, the 0-1 knapsack problem, a type of combinatorial problem, is solved. Combinatorial problems are frequently addressed using meta-heuristic algorithms. These algorithms are developed by mathematically modeling processes inspired by nature. This study examines the Starfish Optimization Algorithm (SOA), a meta-heuristic algorithm proposed in recent years (Zhong et al., 2025). SOA was first proposed by Zhong and their colleagues in 2025. They tested the performance of SOA on 65 benchmark functions drawn from classical functions and the CEC 2017 and CEC 2022 test suites. Additionally, SOA has been used by many researchers to solve various problems. Akbulut presented a modified starfish optimization algorithm (M-SOA)—an enhanced version of SAO. M-SAO incorporates three key improvements: an exponentially decreasing adaptive step size, momentum vector integration, and a multi-candidate local search strategy (Akbulut, 2026). Jameel et al. present the Multi-Objective Starfish Optimization Algorithm (MOSFOA), designed to address complex challenges in engineering design and optimal power flow analysis. MOSFOA is an enhanced version of the Starfish Optimization Algorithm (SFOA) that incorporates elitist non-dominated sorting (NDS) and crowding distance (CD) mechanisms to maintain solution diversity and guide convergence toward the Pareto-optimal front (Jameel et al., 2026). To further enhance the convergence and global optimization capabilities of the Starfish Optimization Algorithm (SFOA), an improved version—SFOAL—Liu et al. was introduced, integrating sine chaotic mapping, t-distribution mutation, and logarithmic spiral reverse learning methods (Liu et al., 2025). Aktaş and Kiliç have introduced a novel discrete Starfish Optimization Algorithm (D-SFOA) to solve the complex and discrete Symmetric Traveling Salesman Problem (STSP) (Aktaş and Kiliç, 2025). Singla et al. propose the Starfish Optimization Algorithm (SAO), a new heuristic method (meta-heuristic approach) based on the regeneration and coordinated feeding behaviors of starfish. The algorithm minimizes the sum of squared errors (SSE) between experimental data and model predictions for twelve different PEMFC stacks with power ratings ranging from 12 W to 6 kW (Singla et al., 2025). Gafar et al. have developed an intelligent Enhanced Starfish Optimization (ESFO) algorithm to optimize a secure wireless communication infrastructure (Gafar et al., 2026). Raghu et al. propose an attribute selection approach based on the Adaptive Mutation Probability-based Starfish Optimization Algorithm (AMP-SFOA) for the automated diagnosis and classification of DR into different categories (Raghu et al., 2026). A review of the literature on SAO reveals that a feature selection study proposing a binary version of the algorithm has been conducted in only one instance; it appears that

the performance of SAO in binary optimization problems has not been sufficiently investigated. In this study, the binary 0-1 knapsack problem—a type of combinatorial problem—is solved. Solutions to the 0-1 knapsack problem using various metaheuristic algorithms have been presented in the literature, and the resulting optimal values have been reported. The fact that the optimal results for 0-1 knapsack problem instances are known has frequently led researchers to select them as test problems for binary optimization. Similarly, 0-1 knapsack problem instances were chosen for this study to evaluate the performance of binary SAO precisely because their optimal results are known.

To derive a binary version of a continuous optimization algorithm, the continuous search space must be mapped to a binary search space. Transfer functions are frequently employed at this stage. Commonly used families of transfer functions in the literature include S-shaped, V-shaped, U-shaped, Taper-shaped, and Z-shaped functions (Yildizdan and Bas, 2024). This study utilizes bell-shaped transfer functions, which have been proposed recently. Bell-shaped transfer functions were first introduced by Aghdasi et al. in 2026. The rationale for selecting bell-shaped transfer functions in this study is their novelty and the fact that their performance is being tested with the SAO for the first time. Consequently, this study will serve as a resource and provide guidance for researchers intending to use the family of bell-shaped transfer functions.

The study proceeds as follows: Section 2 presents the mathematical model of SAO, the definition of the 0-1 knapsack problem and the datasets used, and the definitions of bell-shaped transfer functions. In Section 3, variations based on bell-shaped transfer functions—developed using SAO—are tested on 25 different 0-1 knapsack instances, and the results are presented. Finally, the results are evaluated in the last section.

2. Materials and Methods

2.1. Starfish Optimization Algorithm (SAO)

The Starfish Optimization Algorithm (SAO), inspired by the behavior of starfish, was proposed by Zhong et al. in 2025. Starfish possess the ability to regenerate. Inspired by this capability, a model based on the SFO algorithm was designed. SAO first generates its variables within a continuous search space using Equations (1)-(3) (Zhong et al., 2025).

$$X = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1dim} \\ X_{21} & X_{22} & \dots & X_{2dim} \\ \vdots & \vdots & \dots & \vdots \\ X_{pop1} & \vdots & \dots & X_{popdim} \end{bmatrix} \quad (1)$$

$$X_{ij} = lower_j + r(upper_j - lower_j), \quad i = 1, 2, \dots, pop, \quad j = 1, 2, \dots, dim \quad (2)$$

$$Fitness = \begin{bmatrix} Fitness(X_1) \\ Fitness(X_2) \\ \vdots \\ Fitness(X_{pop}) \end{bmatrix} \quad (3)$$

where, pop represents the population size, dim the number of search space variables, and $lower$ and $upper$ represent bounds of the search space. Fitness values are calculated using the initial starfish positions contained in matrix X . Fitness is a matrix that stores the obtained suitability values. Once the initialization phase concludes, the SAO proceeds to the exploration and exploitation phases within the main loop (Zhong et al., 2025).

Exploration: At this stage, the starfish conducts a search based on its five arms. Different exploration equations are employed depending on whether the dimensionality variable is greater than or less than 5.

If the decision variables in the optimization problem exceed 5, the starfish explore the search space using all their arms. This stage is modeled by Equations (4)-(7). Where, $X_{i,q}^{iter+1}$ and $X_{i,q}^{iter}$ denote the obtained and current positions of a star, respectively, while $X_{best,q}^{iter}$ represents the best current position. The value q refers to five dimensions randomly selected from the total dimensions (dim), and $rand$ is a value chosen from the interval (0, 1). T_{max} represents the maximum iteration number (Zhong et al., 2025).

$$\begin{cases} X_{i,q}^{iter+1} = X_{i,q}^{iter} + a_1(X_{best,q}^{iter} - X_{i,q}^{iter}) \cos \theta, rand \leq 0.5 \\ X_{i,q}^{iter+1} = X_{i,q}^{iter} - a_1(X_{best,q}^{iter} - X_{i,q}^{iter}) \sin \theta, rand > 0.5 \end{cases} \quad (4)$$

$$a_1 = (2rand - 1)\pi \quad (5)$$

$$\theta = \frac{\pi}{2} \cdot \frac{iter}{T_{max}} \quad (6)$$

$$X_{i,q}^{iter+1} = \begin{cases} X_{i,q}^{iter+1}, & lower_{b,q} \leq Y_{i,q}^T \leq upper_{b,q} \\ X_{i,q}^{iter}, & \text{other wise} \end{cases} \quad (7)$$

If the search space dimension (dim) is 5 or less, the starfish searches the space using a single arm. This scenario is described in Equations (8) and (9). A_1 and A_2 are two numbers randomly selected from the interval (-1, 1), respectively. $X_{k1,q}^{iter}$ and $X_{k2,q}^{iter}$ are two randomly selected starfish, respectively. E_t is the energy of the starfish (Zhong et al., 2025).

$$X_{i,q}^{iter} = E_t X_{i,q}^{iter} + A_1(X_{k1,q}^{iter} - X_{i,q}^{iter}) + A_2(X_{k2,q}^{iter} - X_{i,q}^{iter}) \quad (8)$$

$$E_t = \frac{T_{max} - iter}{T_{max}} \cdot \cos \theta \quad (9)$$

Exploitation: In the exploitation phase, SAO employs hunting strategies. Distances between a starfish's current position and other starfish are calculated; two of these distances are randomly selected, and a new position is computed. This process is illustrated in Equations (10)–(13) (Zhong et al., 2025).

$$d_g = (X_{best}^{iter} - X_{g_q}^{iter}), \quad g = 1, \dots, 5 \quad (10)$$

$$X_i^{iter+1} = X_i^{iter} + rand_1 d_{g_1} + rand_2 d_{g_2} \quad (11)$$

$$X_i^{iter+1} = \exp(-T - iter \cdot iter / T_{max}) \cdot X_i^{iter} \quad (12)$$

$$X_i^{iter+1} = \begin{cases} X_i^{iter+1} & lower_b \leq X_i^{iter+1} \leq upper_b \\ lower_b & X_i^{iter+1} < lower_b \\ upper_b & X_i^{iter+1} > upper_b \end{cases} \quad (13)$$

Bell-Shaped Transfer Functions

Transfer functions play a crucial role in mapping continuous optimization dynamics to binary decisions. Although S-, V-, U-, and taper-shaped transfer functions are widely used in the literature, new transfer functions continue to be proposed. One such type is the bell-shaped transfer function, first introduced by Aghdasi et al. in 2026. Distinguished by their Gaussian-like curves and exponential decay characteristics, these functions offer an innovative balance between exploration and exploitation (Aghdasi et al., 2026). In this study, bell-shaped functions are utilized to map SAO's continuous search space to a binary search space. This enables the algorithm to strike a balance between exploration and exploitation when solving the 01-knapsack problem.

Table 1. Bell-shaped transfer functions (Aghdasi et al., 2026)

Bell-Shaped	
Label	Formula
Bell_1	$Bell_1(x) = \exp(-(x - 1)^2)$
Bell_2	$Bell_2(x) = \exp(-(x - 1)^4)$
Bell_3	$Bell_3(x) = \exp(-(x - 1)^6)$
Bell_4	$Bell_4(x) = \exp(-(x - 1)^8)$

2.2. 0-1 Knapsack Problem

The 0-1 knapsack problem was selected to test the performance of the bell-shaped transfer function-based SAO (SAO_{Bin}) in binary optimization problems. The 0-1 knapsack problem is a discrete optimization problem. The objective is to optimally place objects—defined by their value and weight—into a knapsack to achieve maximum profit; essentially, the goal is to determine which items to select. When formulated as an optimization problem, a value of 0 in the binary search space indicates that an item is not selected, while a value of 1 indicates that it is. The aim is to maximize profit without exceeding the knapsack's capacity. A fitness function is defined by encoding the item selection information into the binary search space of the binary SAO, thereby allowing the algorithm's performance to be evaluated. The 0-1 knapsack problem is described by Equations (14)–(16). Table 2 shows examples from the 0-1 knapsack dataset used in the study. This is an online dataset accessible to everyone (Bas & Guner, 2025). The average fitness values derived relative to the optimum value were calculated using Equation 17.

$$\text{Maximize Fitness}(x) = \sum_{i=1}^{dim} X_i v_i \quad (14)$$

Object to:

$$\text{Fitness}(x) = \sum_{i=1}^{dim} X_i w_i < C \quad (15)$$

$$i = 1, 2, 3, \dots, dim; X_i = 0 \text{ or } 1; V_i > 0; w_i > 0; C > 0 \quad (16)$$

$$\text{Gap} = \left(\frac{\text{Optimum Value} - \text{Mean Fitness}}{\text{Optimum Value}} \right) \times 100 \quad (17)$$

Table 2. 0–1 Knapsack Problem Dataset Details

Instance	Capacity (C)	dim	Optimum Value
ID1	1863633	8	3924400
ID2	1822718	8	3813669
ID3	1609419	8	3347452
ID4	2112292	8	4187707
ID5	2493250	8	4955555
ID6	2805213	12	5688887
ID7	3259036	12	6498597
ID8	2489815	12	5170626
ID9	3453702	12	6992404
ID10	2520392	12	5337472
ID11	3780355	16	7850983
ID12	4426945	16	9352998
ID13	4323280	16	9151147
ID14	4550938	16	9348889
ID15	3760429	16	7769117
ID16	5169647	20	10727049
ID17	4681373	20	9818261
ID18	5063791	20	10714023
ID19	4286641	20	8929156

ID20	4476000	20	9357969
ID21	6404180	24	13549094
ID22	5971071	24	12233713
ID23	5870470	24	12448780
ID24	5762284	24	11815315
ID25	6654569	24	13940099

3. Experimental Analyses and Discussion

The tests in this study were conducted on a computer equipped with a 13th-generation Intel Core i5-1035G1 1.00 GHz processor and 12 GB of RAM, running a 64-bit operating system. Additionally, MATLAB version R2018b was used in the study.

In this study, binary versions of the SAO algorithm (SAO_{Bin}(Bell₁), SAO_{Bin}(Bell₂), SAO_{Bin}(Bell₃), and SAO_{Bin}(Bell₄)) were developed using bell-shaped transfer functions (Bell₁, Bell₂, Bell₃, and Bell₄). The performance of the bell-shaped transfer function-based binary SAO algorithm was tested on 25 different 0-1 knapsack datasets, comprising both low-dimensional and large-scale instances. Thus, the performance of the binary SAO was evaluated across 0-1 knapsack instances of varying sizes (8, 12, 16, 20, and 24). Tests for the binary SAO variants were conducted using the parameter settings listed in Table 3. Each binary SAO variant was executed independently 20 times, and the statistical results—including mean fitness, fitness standard deviation, best fitness, worst fitness, mean gap value, and average time—are presented in Tables 4 through 9. To facilitate the rapid interpretation of results in the tables, the sample 0-1 knapsack results for each SAO variation were aggregated and subsequently sorted.

Based on the results for average fitness, average standard deviation, and average gap, the most successful binary SAO variation was SAO_{Bin}Bell₁, followed by SAO_{Bin}Bell₂, and SAO_{Bin}Bell₃. The least successful binary SAO variation was SAO_{Bin}Bell₄. In terms of the best fitness outcome, SAO_{Bin}Bell₃ achieved the best results, while SAO_{Bin}Bell₁ yielded the worst. In this case, the results indicate that while SAO_{Bin}Bell₁ achieves good performance in terms of average fitness, average standard deviation, and average gap, it fails to perform adequately in attaining the best fitness results. When the mean execution times of the SAO variants are examined, it is observed that the SAO_{Bin}Bell₁ and SAO_{Bin}Bell₄ variants operate and produce results faster than the other Binary SAO variants. Based on the results for average fitness, average standard deviation, average gap, and average time, the most successful Binary SAO version utilizing a bell-shaped transfer function is SAO_{Bin}Bell₁. The Wilcoxon Signed-Rank Test was used to determine whether there were significant differences among the results of the Binary SAO versions. The results of the Wilcoxon Signed-Rank Test comparing the Binary SAO versions are presented in Table 10. An examination of the results reveals that the *p*-values are generally 0.05 or higher; this indicates that the results of the Binary SAO variations are similar to one another and demonstrate comparable levels of performance.

Convergence plots for the dual SAO versions were generated using the ID1, ID7, ID13, ID19, ID21, and ID25 0-1 knapsack sample datasets, and the results

are presented in Figure 1. For the ID1 0-1 knapsack dataset, $SAO_{Bin}Bell_1$ and $SAO_{Bin}Bell_3$ converged fastest, while $SAO_{Bin}Bell_4$ converged slowest to the optimal result. For the ID7 0-1 knapsack dataset, $SAO_{Bin}Bell_2$ converged fastest, while $SAO_{Bin}Bell_4$ converged slowest. For the ID13 0-1 knapsack dataset, $SAO_{Bin}Bell_1$ converged fastest, while $SAO_{Bin}Bell_4$ converged slowest. For the ID19 0-1 knapsack dataset, $SAO_{Bin}Bell_3$ converged fastest, while $SAO_{Bin}Bell_4$ converged slowest. For the ID21 0-1 knapsack dataset, $SAO_{Bin}Bell_1$ converged fastest, while $SAO_{Bin}Bell_3$ converged slowest. For the ID25 0-1 knapsack dataset, $SAO_{Bin}Bell_1$ converged fastest, while $SAO_{Bin}Bell_4$ converged slowest. Overall, $SAO_{Bin}Bell_4$ was the slowest to converge to the optimal results, whereas $SAO_{Bin}Bell_1$ and $SAO_{Bin}Bell_3$ were the fastest.

Table 3. Parameter settings

Parameters	Values
Population size (pop)	20
Maximum Iterations (T_{max})	5000
Run	20
G_p	0.5

Table 4. The mean fitness results of Binary SAO (SAO_{Bin}) based on a bell-shaped transfer functions

Instance	$SAO_{Bin}(Bell_1)$	$SAO_{Bin}(Bell_2)$	$SAO_{Bin}(Bell_3)$	$SAO_{Bin}(Bell_4)$
ID1	3924400	3924400	3924400	3924400
ID2	3813669	3813669	3813669	3813669
ID3	3347452	3347452	3347452	3347452
ID4	4187707	4187707	4187707	4187707
ID5	4955555	4955555	4955555	4955555
ID6	5688562.85	5688887	5688510.65	5688186.5
ID7	6498597	6497318.1	6496039.2	6498597
ID8	5170626	5170626	5170578.5	5170626
ID9	6992404	6992187.55	6990605.15	6990821.6
ID10	5335076.9	5337472	5335076.9	5337472
ID11	7837918.55	7837957.15	7828951.75	7828375.6
ID12	9338072.35	9341925	9340113.8	9344782.65
ID13	9134608.75	9128737.65	9130188.55	9134219.25
ID14	9340936.2	9330288.1	9327325.3	9338439.1
ID15	7764461.3	7758790.1	7764839.8	7757285.3
ID16	10701751.05	10686571	10692456.15	10687626
ID17	9781212.75	9784712.2	9776136	9760662.75
ID18	10695938.35	10669210.4	10693107.95	10649086.45
ID19	8888415.4	8882454.4	8866162.65	8852185.8
ID20	9343857.3	9339994.65	9348758.35	9330594.5
ID21	13453516.1	13447289.65	13471905.65	13428270.05
ID22	12160713.25	12177160.3	12138900.85	12158525.45
ID23	12356824.45	12354848.15	12360287.45	12386243.15
ID24	11762613.25	11763716	11754228.2	11751466.95
ID25	13859899.3	13867635.05	13840797	13844252.5
Total	206334788.10	206286563.45	206243752.85	206166501.60
Rank	1	2	3	4

Table 5. The mean standard deviation of fitness results of Binary SAO (SAO_{Bin}) based on a bell-shaped transfer functions

Instance	SAO_{Bin} (Bell₁)	SAO_{Bin} (Bell₂)	SAO_{Bin} (Bell₃)	SAO_{Bin} (Bell₄)
ID1	0	0	0	0
ID2	0	0	0	0
ID3	0	0	0	0
ID4	0	0	0	0
ID5	0	0	0	0
ID6	1449.64286981311	0	1683.08836666409	2162.73478190048
ID7	0	5719.41467284897	7872.74228689022	0
ID8	0	0	212.426457862480	0
ID9	0	967.993827459659	7091.94843151466	7076.70793519133
ID10	10711.2128258195	0	10711.2128258195	0
ID11	16204.2507758889	21612.2561823634	27326.7165683676	21987.1896899799
ID12	35150.2115218237	26382.3055235293	30864.8835795463	20234.9433257760
ID13	21562.2093679880	27854.6836471280	35610.1801241029	20438.6974656436
ID14	11475.6202795317	21878.1288069784	24939.4819008778	17693.6395408067
ID15	8363.65597533066	18666.7127061149	7757.30876827240	25540.4445703796
ID16	32313.0966449649	40278.8205362257	34907.1319666084	46713.5365009015
ID17	26911.0050426939	31855.0739172535	38386.7351143184	34699.5964664463
ID18	27509.6663188835	56663.8129816645	31427.2837724350	65817.1121737509
ID19	40476.5810294247	34391.5779246514	55253.8065735842	40539.1062719236
ID20	17476.0240573373	18782.9675287704	18767.5391378866	24233.9903908365
ID21	56134.0024217427	86398.8835224373	41647.5795269065	74230.7316565082
ID22	44096.1913403378	53565.7350670527	54347.7383602212	50796.4305340185
ID23	44650.1391681394	43582.3773382086	53034.6599007370	46637.0701495517
ID24	27086.7077867297	33487.1100936025	35137.6570297002	65004.2909283890
ID25	50201.9201059712	48721.9566298345	61774.4116681605	66447.5877495703
Total	471772.14	570809.81	578754.53	630253.81
Rank	1	2	3	4

Table 6. The best of fitness results of Binary SAO (SAO_{Bin}) based on a bell-shaped transfer functions

Instance	SAO _{Bin} (Bell ₁)	SAO _{Bin} (Bell ₂)	SAO _{Bin} (Bell ₃)	SAO _{Bin} (Bell ₄)
ID1	3924400	3924400	3924400	3924400
ID2	3813669	3813669	3813669	3813669
ID3	3347452	3347452	3347452	3347452
ID4	4187707	4187707	4187707	4187707
ID5	4955555	4955555	4955555	4955555
ID6	5688887	5688887	5688887	5688887
ID7	6498597	6498597	6498597	6498597
ID8	5170626	5170626	5170626	5170626
ID9	6992404	6992404	6992404	6992404
ID10	5337472	5337472	5337472	5337472
ID11	7850983	7850983	7850983	7850983
ID12	9352998	9352998	9352998	9352998
ID13	9151147	9151147	9151147	9151147
ID14	9348889	9348889	9348889	9348889
ID15	7769117	7769117	7769117	7769117
ID16	10727049	10727049	10727049	10727049
ID17	9818261	9818261	9818261	9818261
ID18	10714023	10714023	10714023	10714023
ID19	8929156	8929156	8929156	8929156
ID20	9357969	9357969	9357969	9357969
ID21	13549094	13549094	13549094	13549094
ID22	12220754	12233713	12233713	12233713
ID23	12440695	12416840	12445379	12445379
ID24	11810051	11815315	11810051	11815315
ID25	13930566	13940099	13940099	13925043
Total	206887521.00	206891422.00	206914697.00	206904905.00
Rank	4	3	1	2

Table 7. The worst of fitness results of Binary SAO (SAO_{Bin}) based on a bell-shaped transfer functions

Instance	SAO _{Bin} (Bell ₁)	SAO _{Bin} (Bell ₂)	SAO _{Bin} (Bell ₃)	SAO _{Bin} (Bell ₄)
ID1	3924400	3924400	3924400	3924400
ID2	3813669	3813669	3813669	3813669
ID3	3347452	3347452	3347452	3347452
ID4	4187707	4187707	4187707	4187707
ID5	4955555	4955555	4955555	4955555
ID6	5682404	5688887	5681360	5681360
ID7	6498597	6473019	6473019	6498597
ID8	5170626	5170626	5169676	5170626
ID9	6992404	6988075	6960756	6960756
ID10	5289570	5337472	5289570	5337472
ID11	7806012	7772352	7772352	7779541
ID12	9214922	9267143	9237573	9267143
ID13	9068381	9060066	9027774	9079887
ID14	9299615	9287810	9276597	9292759
ID15	7746722	7689144	7746722	7655263
ID16	10619367	10582648	10584562	10593266
ID17	9714369	9711754	9694195	9698258
ID18	10611486	10508515	10619769	10507079

ID19	8791224	8819664	8703468	8776852
ID20	9301184	9283482	9276842	9276328
ID21	13332069	13239325	13382271	13276421
ID22	12025498	12039746	12058908	12038545
ID23	12261269	12286668	12251119	12267210
ID24	11699025	11694603	11671866	11525428
ID25	13754307	13756127	13741029	13705736

Table 8. The mean time results of Binary SAO (SAO_{Bin}) based on a bell-shaped transfer functions

Instance	SAO_{Bin} (Bell₁)	SAO_{Bin} (Bell₂)	SAO_{Bin} (Bell₃)	SAO_{Bin} (Bell₄)
ID1	0.822072495	0.9696065	1.041973075	0.89952215
ID2	0.768573655	0.804546935	0.931682065	0.821622165
ID3	0.73460084	0.794940745	0.825703325	0.77624685
ID4	0.73377518	0.8001286	0.8211236	0.79631543
ID5	0.730213655	0.819941085	0.794610745	0.771156775
ID6	0.78101038	0.84990295	0.85761029	0.83791517
ID7	0.74834766	0.84906083	0.877627095	0.814205895
ID8	0.784875475	0.845556315	0.87843733	0.8499005
ID9	0.75430615	0.8454639	0.85261635	0.81653186
ID10	0.78450464	0.836144105	0.881796865	0.843677615
ID11	0.77935753	0.895803495	0.907133695	0.85939305
ID12	0.777265865	0.89544	0.926155865	0.857193655
ID13	0.766825575	0.88673334	0.921526745	0.868989845
ID14	0.7794685	0.89231521	0.90964959	0.89003498
ID15	0.797809005	0.89610215	0.919569415	0.88531218
ID16	0.873451995	0.94084704	0.96783822	0.9054337
ID17	0.815677895	0.949253785	0.97564325	0.920314995
ID18	0.832541465	0.945275125	0.97748295	0.94796922
ID19	0.842516985	0.949478185	0.96710324	0.92087563
ID20	0.804376465	0.94083519	0.95829249	0.90813154
ID21	0.84282586	1.011587125	1.038485905	0.973505805
ID22	0.85858539	1.0024076	1.05769758	1.008500855
ID23	0.91586239	1.017846075	1.05543178	0.983332955
ID24	0.8537112	1.017129375	1.035522035	0.995519735
ID25	0.897309055	1.029086685	1.04832193	1.00996747
Total	20.08	22.69	23.43	22.16
Rank	1	3	4	2

Table 9. The mean gap results of Binary SAO (SAO_{Bin}) based on a bell-shaped transfer functions

Instance	SAO_{Bin} (Bell₁)	SAO_{Bin} (Bell₂)	SAO_{Bin} (Bell₃)	SAO_{Bin} (Bell₄)
ID1	0	0	0	0
ID2	0	0	0	0
ID3	0	0	0	0
ID4	0	0	0	0
ID5	0	0	0	0
ID6	5.6979e-07	0	6.6150e-07	1.2313e-06
ID7	0	1.9679e-06	3.9359e-06	0
ID8	0	0	9.1865e-08	0
ID9	0	3.0955e-07	2.5725e-06	2.2630e-06

ID10	4.4873e-06	0	4.4873e-06	0
ID11	1.6640e-05	1.6591e-05	2.8061e-05	2.8795e-05
ID12	1.5958e-05	1.1838e-05	1.3775e-05	8.7836e-06
ID13	1.8072e-05	2.4488e-05	2.2902e-05	1.8497e-05
ID14	8.5066e-06	1.9896e-05	2.3065e-05	1.1177e-05
ID15	5.9925e-06	1.3292e-05	5.5053e-06	1.5229e-05
ID16	2.3583e-05	3.7734e-05	3.2248e-05	3.6751e-05
ID17	3.7734e-05	3.4169e-05	4.2904e-05	5.8660e-05
ID18	1.6879e-05	4.1826e-05	1.9521e-05	6.0608e-05
ID19	4.56264e-05	5.2302e-05	7.0547e-05	8.62009e-05
ID20	1.50798e-05	1.9207e-05	9.8425e-06	2.9252e-05
ID21	7.05419e-05	7.5137e-05	5.6969e-05	8.9174e-05
ID22	5.96709e-05	4.6226e-05	7.7500e-05	6.1459e-05
ID23	7.3867e-05	7.5454e-05	7.1085e-05	5.0235e-05
ID24	4.4604e-05	4.3671e-05	5.1701e-05	5.4038e-05
ID25	5.7531e-05	5.1982e-05	7.1234e-05	6.8755e-05
Total	0.000515	0.000566	0.000609	0.000681
Rank	1	2	3	4

Table 10. The p -values of Wilcoxon Signed-Rank Test among Binary SAO (SAOBin) variants based on bell-shaped transfer functions

Instance	SAOBin (Bell ₁) - SAOBin (Bell ₂)	SAOBin (Bell ₁) - SAOBin (Bell ₃)	SAOBin (Bell ₁) - SAOBin (Bell ₄)	SAOBin (Bell ₂) - SAOBin (Bell ₃)	SAOBin (Bell ₂) - SAOBin (Bell ₄)	SAOBin (Bell ₃) - SAOBin (Bell ₄)
ID1	1	1	1	1	1	1
ID2	1	1	1	1	1	1
ID3	1	1	1	1	1	1
ID4	1	1	1	1	1	1
ID5	1	1	1	1	1	1
ID6	1	1	0.75	1	0.5	1
ID7	1	0.5	1	1	1	0.5
ID8	1	1	1	1	1	1
ID9	1	0.5	1	0.75	1	1
ID10	1	1	1	1	1	1
ID11	0.9873	0.3499	0.1476	0.2397	0.1498	0.9015
ID12	0.8535	0.8164	0.4453	0.9609	0.8886	0.7924
ID13	0.3584	0.5269	0.8354	0.6883	0.5009	0.8344
ID14	0.0605	0.0516	0.7759	0.835	0.2400	0.1191
ID15	0.3047	0.7129	0.3047	0.1016	0.9414	0.4101
ID16	0.3269	0.3504	0.2144	0.4897	0.9245	1
ID17	0.9055	0.7171	0.0854	0.5381	0.0700	0.2273
ID18	0.0532	0.8684	0.0061	0.0776	0.3980	0.0157
ID19	0.4220	0.1569	0.0148	0.2431	0.0400	0.3759
ID20	0.5013	0.2176	0.1304	0.0175	0.1978	0.0030
ID21	0.8519	0.2179	0.1913	0.5460	0.3980	0.0619
ID22	0.2471	0.0859	0.6813	0.0793	0.1364	0.1913
ID23	0.5015	0.7089	0.1168	0.4330	0.0486	0.1212
ID24	0.9702	0.4114	0.8519	0.5015	0.6873	0.6813
ID25	0.6813	0.1560	0.6813	0.2043	0.2627	0.6580

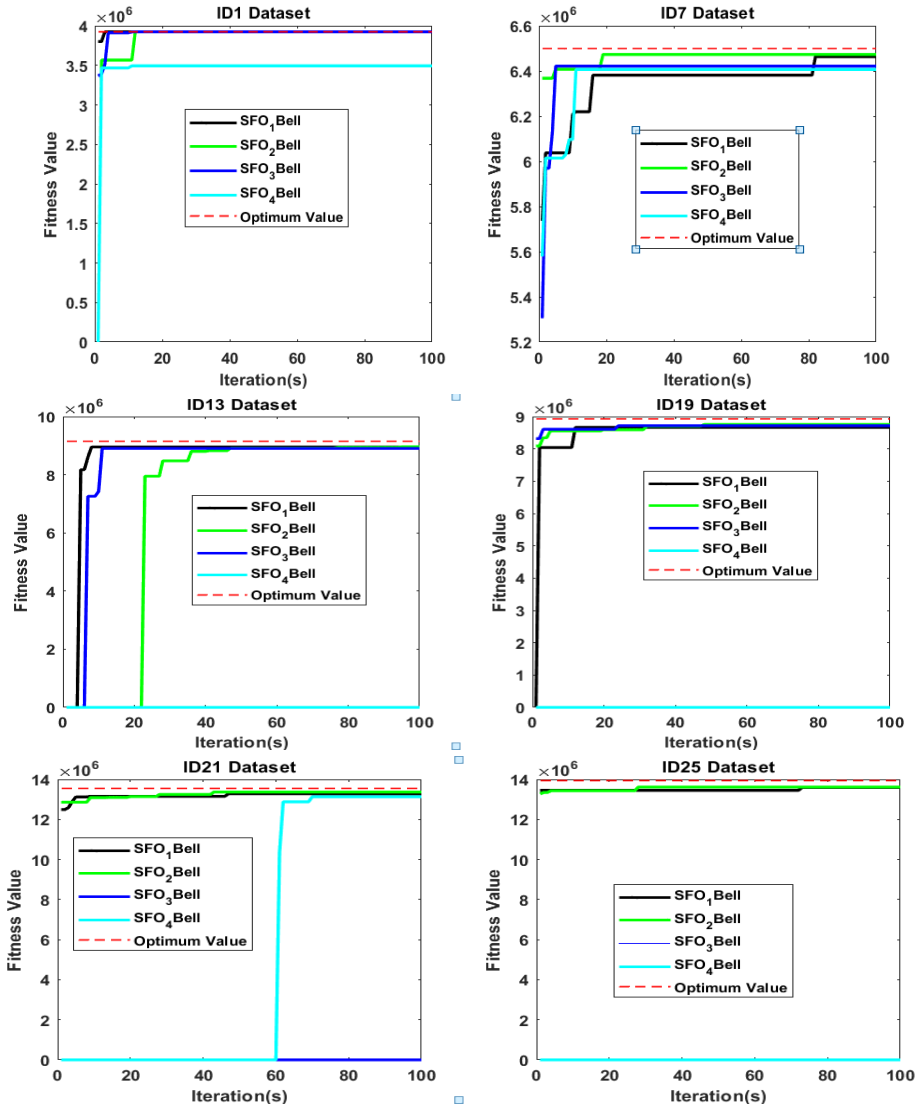


Figure 1. Convergence plots of Binary SAO (SAOBin) variants based on bell-shaped transfer functions on the ID1, ID7, ID13, ID19, ID21, and ID25 datasets.

4. Conclusions

This study focuses on Starfish Optimization Algorithm (SAO), a successful meta-heuristic algorithm proposed in recent years. A review of the literature revealed that the binary versions of SAO have not been sufficiently analyzed or applied to combinatorial problems. This study was conducted to address this gap in the literature. Variables from SAO's continuous search space were mapped to binary variables using a family of bell-shaped transfer functions recently introduced in the field. The aim was to enhance the exploration and exploitation capabilities of Binary SAO (SAOBin) within the binary search space through the use of these bell-shaped transfer functions. Consequently, four distinct SAOBin

variants were developed ($SAO_{Bin}Bell_1$, $SAO_{Bin}Bell_2$, $SAO_{Bin}Bell_3$, and $SAO_{Bin}Bell_4$). The performance of these variants was evaluated using the 0-1 knapsack problem, a well-known type of combinatorial problem. Twenty-five different 0-1 knapsack datasets—all open-access and with known optimal solutions from the literature—were utilized in this study. The four SAO_{Bin} variants underwent various statistical analyses; based on metrics such as average fitness, average standard deviation, and average execution time, $SAO_{Bin}Bell_1$ emerged as the most successful variant. The Wilcoxon Signed-Rank Test was employed to determine if there were significant differences between the $SAO_{Bin}Bell_1$ variant and the others; the results indicated that all Bell-shaped SAO_{Bin} variants achieved comparable performance outcomes.

Future research will aim to evaluate the performance of SAO_{Bin} on a different type of combinatorial problem: the Uncapacitated Facility Location Problem (UFLP). Additionally, the effectiveness of other new families of transfer functions (M-Shaped and L-Shaped) in conjunction with SAO_{Bin} will be investigated.

Summary of Researcher Contribution Statement

The author declares that they contributed to the entire paper.

Declaration of Competing Interest

The author declares that there are no known financial conflicts of interest or personal relationships that could be considered to have influenced the work presented in this study.

Funding

The author did not receive any financial support for the research, writing, or publication of this study.

The Declaration of Ethics Committee Approval

This study does not require ethics committee permission or any special permission

Data and Code Availability Statement:

The data will be provided upon request.

Dataset access source: http://artemisa.unicauc.edu.co/~johnyortega/instances_01_KP/

References

- Aghdasi, H., Mohammed, B. O., Oskouei, S., & Salehpour, P. (2026). Binary Dhole Optimization Algorithm with bell-shaped transfer function for feature selection. *Scientific Reports*.
- Akbulut, H. (2026). A modified starfish optimization algorithm (M-SFOA) for global optimization problems and its application to heart disease risk prediction. *Expert Systems with Applications*, 131088.
- Aktaş, M., & Kiliç, F. (2025). Discrete starfish optimization algorithm for symmetric travelling salesman problem. *IEEE Access*, 13, 102675-102687.
- Bas, E., & Guner, L. B. (2025). The binary crayfish optimization algorithm with bitwise operator and repair method for 0–1 knapsack problems: an improved model. *Neural Computing and Applications*, 37(6), 4733-4767.
- Gafar, M., Sarhan, S., Shaheen, A. M., & Alwakeel, A. S. (2026). A Novel Starfish Optimization Algorithm for Secure STAR-RIS Communications. *Biomimetics*, 11(4), 243.
- Jameel, M., Merah, H., El-latif, A. M. A., Al-Shami, T. M., Almutairi, A., & Abouhawwash, M. (2026). Multiobjective starfish optimization algorithm for engineering design and optimal power flow problems. *Scientific Reports*, 16(1), 3302.
- Liu, Y., Fu, M., Liu, Z., Liu, H., Peng, W., Li, L., ... & Jia, C. (2025). An enhanced starfish optimization algorithm via joint strategy and its application in ultra-wideband indoor positioning. *Biomimetics*, 10(5), 338.
- Raghu, B. R., Krishna, J. V., & Pradeep, N. (2026). Feature Selection Using Adaptive Mutation Probability Based Starfish Optimization Algorithm in Diabetic Retinopathy Classification. *Journal of Image and Graphics*, 14(2).
- Singla, M. K., Aljaidi, M., Gupta, J., Jangir, P., Arpita, Kumar, R., ... & Alharbi, A. H. (2025). Enhanced mathematical modeling of PEM fuel cells using the starfish optimization algorithm. *Ionics*, 31(12), 13229-13268.
- Yildizdan, G., & Bas, E. (2024). A new binary coati optimization algorithm for binary optimization problems. *Neural Computing and Applications*, 36(6), 2797-2834.
- Zhong, C., Li, G., Meng, Z., Li, H., Yildiz, A. R., & Mirjalili, S. (2025). Starfish optimization algorithm (SFOA): a bio-inspired metaheuristic algorithm for global optimization compared with 100 optimizers. *Neural Computing and Applications*, 37(5), 3641-3683.

CHAPTER 7

**Wearable Biosensors for Pesticide
Analysis: Design Principles, Current
Applications and Future Perspectives**

Esra Armağan¹

¹ Lecturer, Karamanoğlu Mehmetbey University, Ermenek Uysal and Hasan Kalan
Vocational School of Health Services, Karaman, Türkiye,
ORCID: 0000-0002-5414-2392

1. Introduction

Feeding a growing world population requires more intensive agricultural production. Plant diseases and pests are estimated to cut global agricultural productivity by up to about 40% each year, with losses exceeding USD 220 billion (Teixeira, Gomes, de Oliveira, SAOres, & Raymundo-Pereira, 2025), and pesticides remain the most widely used means of protecting yields. Roughly 3 billion kg of plant protection products are applied worldwide every year, at a cost of more than USD 40 billion (Pimentel, 2009).

These benefits carry environmental and health costs that are increasingly hard to ignore. Many pesticides persist in soil, leach into ground and surface waters and accumulate along the food chain, and residues reach the human body. A human biomonitoring study that applied harmonized suspect screening to 2088 urine samples from 525 parent-child pairs in Latvia, Hungary, Czechia, Spain and the Netherlands found at least two different pesticides in 84% of the samples (Ottensmeyer et al., 2023). Chronic low-level exposure has been linked to immunosuppression, endocrine disruption, neurodegenerative diseases and several cancers (Damalas & Eleftherohorinos, 2011), so rapid on-site monitoring of residues is a necessity. Residues can also be phytotoxic, causing oxidative damage and impairing plant health (Tanveer et al., 2024; Zhou et al., 2022).

Organophosphorus (OP) compounds deserve particular attention, because the same chemical class includes widely used agricultural and household insecticides as well as nerve agents that can be weaponized (Sharma, Nagpal, Pakade, & Katnoria, 2010). Rapid, selective detection of OP compounds therefore matters for defense and security as well as for food safety (Zhu, Azzarelli, & Swager, 2016). Fungicides such as carbendazim, herbicides such as diquat and paraquat, and post-harvest agents such as diphenylamine are further targets for monitoring (Teixeira et al., 2026).

Over the past decade, wearable sensors measuring glucose, lactate or alcohol in sweat, tears or interstitial fluid have proved successful in health care, encouraging transfer of the approach to hazardous chemicals and agriculture (Bandodkar et al., 2015; Giraldo, Wu, Newkirk, & Kruss, 2019; Jia et al., 2013; Kim, Campbell, de Ávila, & Wang, 2019; Kim et al., 2016; Malzahn, Windmiller, Valdés-Ramírez, Schöning, & Wang, 2011; Yao et al., 2012; Zhu et al., 2016). Flexible, biocompatible and light, these sensors can be mounted on a glove fingertip or on leaves, stems and fruits, converting chemical information into easily analyzed electrical or optical signals (Li et al., 2024). The World Economic Forum named wearable plant sensors among the top ten emerging technologies of 2023 (Teixeira et al., 2025).

This chapter outlines the design principles of wearable biosensors for pesticide analysis, compares pioneering and recent studies, and discusses barriers to field use and future directions.

2. From the Laboratory to the Field: Limitations of Conventional Methods

Chromatographic techniques are the internationally accepted reference methods for pesticide residues. Liquid and gas chromatography, capillary electrophoresis and supercritical fluid chromatography coupled with electron capture, flame photometric or mass spectrometric detectors offer high accuracy, low limits of determination and simultaneous quantification of many compounds (Raj, Dubey, Malla, & Kumar, 2023; Samsidar, Siddiquee, & Shaarani, 2018). GC×GC–TOFMS, for example, has been used to determine 53 pesticides and metabolites in water, and combined LC-MS/MS and GC-MS/MS 72 pesticides in rice (Harischandra Naik et al., 2021; Romagnoli et al., 2023).

These methods, however, need expensive instruments, trained operators, lengthy extraction and clean-up and laboratory infrastructure; polar or non-volatile analytes may require derivatization, and plant matrices cause matrix effects (Bakanov et al., 2023; Romagnoli et al., 2023). They are therefore poorly suited to rapid screening in the field, in storage or along the supply chain. Remote sensing, drones and multispectral imaging cover large areas quickly but rely on phenotypic changes and cannot detect stress before symptoms appear (Dutta et al., 2024).

Biosensors couple a biological recognition element (enzyme, antibody, aptamer or cell) with a transducer that converts a molecular interaction into a measurable signal. Their portability, speed, low cost and selectivity make them a complementary alternative to chromatography (Rapini & Marrazza, 2016). Wearable biosensors go one step further by combining sampling and analysis at the fingertip or on the produce surface. Table 1 compares the two approaches.

Table 1. Comparison of Conventional Laboratory Methods and Wearable Biosensors for Pesticide Analysis

Criterion	Conventional methods (GC-MS, LC-MS/MS, HPLC)	Wearable biosensors
Place of analysis	Central laboratory	Field, storage facility, market, border checkpoint; directly on the produce surface
Sample preparation	Extraction, clean-up, preconcentration; derivatization when needed	Usually not required; sampling by swiping or direct contact
Analysis time	Hours to days (including preparation)	A few minutes (e.g., 3.28 min; 8 min incubation)
Cost	High instrument and running costs	Low; can fall to a few cents per disposable sensor
Operator	Trained staff	Non-specialist users; smartphone interface
Multi-analyte analysis	Dozens of analytes in a single run	Usually 1–3 analytes; multichannel designs are emerging
Accuracy and validation	Gold standard; accredited methods	Screening/early warning; validation against a reference method required

3. Basic Components of Wearable Pesticide Biosensors

The performance of a wearable pesticide biosensor depends on how well its biorecognition element, transducer, sensing material, flexible substrate and sensor–surface interface work together; Y. Wang, Yu, Chu, Huang, and Li (2026) describe it as a systemic compromise between sensing requirements and interface compatibility rather than the result of any single factor.

3.1. Biorecognition Elements

Enzymes are the most common biorecognition elements in pesticide biosensors, and two strategies dominate (Figure 1). The first is **enzyme inhibition**. Acetylcholinesterase (AChE) and butyrylcholinesterase (BChE) are inhibited by organophosphates and carbamates; in the sensor, the enzyme converts a substrate such as acetylthiocholine into electroactive thiocholine, and in the presence of a pesticide both enzyme activity and the current signal fall. Cholinesterase biosensors are valued for their sensitivity and practicality at low pesticide concentrations (Pachapur et al., 2019). An AChE biosensor enhanced with gold nanoparticles and graphene oxide reached a detection limit of 0.1 ng/mL for chlorpyrifos (Berkal & Nardin, 2023), and a BChE glove biosensor

modified with Prussian blue and carbon black detected dichlorvos directly on fruit peels (Miglione et al., 2025).

Inhibition-based measurement is indirect, needs an added substrate, risks irreversible enzyme inactivation and, because the signal decreases, discriminates poorly at low concentrations (Lan et al., 2012). The **catalytic approach** largely avoids these problems. Organophosphorus hydrolase (OPH) hydrolyzes the P–O or P–S bonds of OP compounds directly, releasing electroactive products such as p-nitrophenol (PNP). The product concentration rises with the amount of pesticide, giving a stronger and more direct signal, and the absence of an external substrate simplifies portable designs (Han, Wang, & Wang, 2024; Lan et al., 2012; Viveros, Paliwal, McCrae, Wild, & Simonian, 2006). All the wearable OP sensors examined in detail here are therefore OPH-based (Liu et al., 2026; F. Zhao et al., 2020). F. Zhao et al. (2020) note that the thermostability of their methyl parathion hydrolase (Luo et al., 2014) helped it stay active on produce surfaces under ambient conditions.

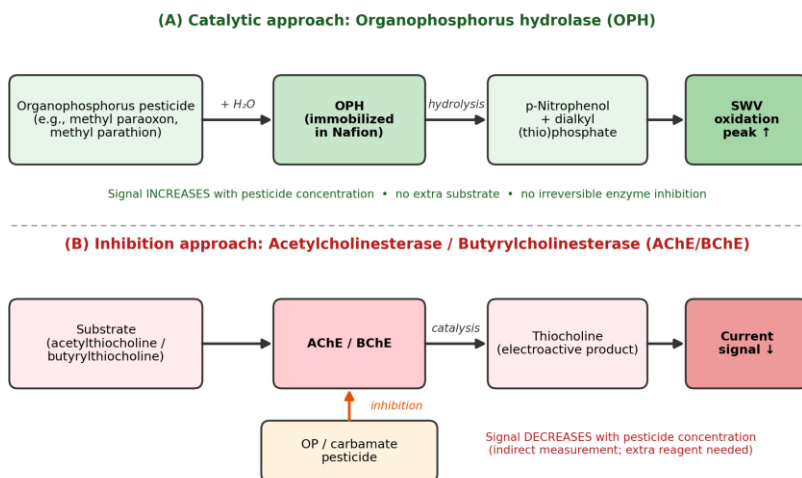


Figure 1. Two main enzymatic recognition strategies in pesticide biosensors (Berkal & Nardin, 2023; Liu et al., 2026; Mishra et al., 2017). SWV: square wave voltammetry.

Aptamers and antibodies are also used. In paper-based colorimetric sensors, aptamer-functionalized gold nanoparticles aggregate in the presence of malathion and turn from red to purple, allowing low-cost screening by eye (Marimuthu, Xu, Song, Chen, & Wen, 2025). Electroactive pesticides such as carbendazim, diquat and diphenylamine can instead be determined by direct electro-oxidation without a biological element. This enzyme-free route improves shelf life and field stability, but selectivity must then come from careful choice of potential window and electrode material (Teixeira et al., 2026).

3.2. Signal Transduction Methods

Electrochemical transduction dominates wearable pesticide sensing because it is easily miniaturized, low in power and cost, and compatible with handheld wireless potentiostats (Hernandez-Vargas et al., 2018). In OPH-based systems, PNP oxidation is usually followed by square wave voltammetry (SWV). Mishra et al. (2017) point out that, unlike constant-potential amperometry, SWV reads the PNP peak at its characteristic potential (+0.85 V), which adds selectivity. Teixeira et al. (2026) used two techniques on one platform, SWV for diquat and differential pulse voltammetry (DPV) for carbendazim and diphenylamine, after showing that DPV failed to resolve the diquat peak at low concentrations and that simultaneous SWV increased the standard deviation; the technique has to suit each channel.

Among **optical transduction** methods, surface-enhanced Raman scattering (SERS), fluorescence and colorimetry are the most common. A silk fibroin/anodic aluminum oxide/gold SERS sensor attached to a nitrile glove detected thiram on fruit (D. Wang et al., 2022); tiered polyvinyl alcohol/hyaluronic acid microneedles collected residues from both the surface and the inner tissue of produce for SERS (Yi, Yuan, Yu, Zheng, & Wang, 2023); and CdTe quantum dots in a polyvinyl alcohol/agarose hydrogel allowed thiram degradation to be followed by smartphone RGB analysis (Chen et al., 2025). Optical sensors can exceed electrochemical ones in sensitivity and speed, but their need for laser sources or confocal microscopes and their sensitivity to ambient light have limited their uptake (Y. Wang et al., 2026).

3.3. Role of Nanomaterials

Nanomaterials are central to the sensitivity and stability of wearable biosensors. Gold nanoparticles (AuNPs), carbon nanotubes, graphene, metal–organic frameworks and quantum dots combine a high surface-to-volume ratio with conductivity, catalytic activity and biocompatibility, which allows dense and stable immobilization of recognition elements, a larger interaction surface and faster electron transfer (Christopher, Kumar, Christopher, Joshiba, & Madhesh, 2020).

Liu et al. (2026) found that, relative to a screen-printed carbon electrode (SPCE) carrying OPH alone, graphene markedly increased the PNP oxidation current and AuNPs raised it further; impedance spectra showed that the graphene and AuNP layers lowered the charge transfer resistance and the enzyme and Nafion layers raised it. F. Zhao et al. (2020) reported that AuNPs electrodeposited on laser-induced graphene (LIG) shifted the PNP oxidation potential from about +0.85 V to about −0.10 V and increased the current; a lower working potential can reduce interference from other electroactive species in plant matrices.

3.4. Flexible Substrates and Fabrication Techniques

A wearable sensor must tolerate the bending of a glove finger, the curvature of a fruit or wind-induced deformation, which depends on the mechanical compliance of its substrate and conductor. Three fabrication approaches are prominent.

Screen printing is inexpensive, commercially available and suited to mass production. Mishra et al. (2017) printed directly onto a nitrile glove finger over a flat 3D-printed mold, using “stress-enduring” inks in which carbon was blended with an elastic styrene–isoprene–styrene block copolymer (PS-PI-PS) and Ag/AgCl with Ecoflex silicone. The intrinsic stretchability of these inks, combined with serpentine interconnects, gave a device that tolerated 50% strain. Liu et al. (2026) printed carbon working and counter electrodes, an Ag/AgCl reference electrode and tracks on flexible thermoplastic polyurethane (TPU) and fixed them to a nitrile glove fingertip. Noting that liquid metal electrodes stretch up to 1000% (Niu et al., 2023) and silver electrodes offer low resistance (Wu et al., 2023) but that both are costly and complex to make, they chose the SPCE as a disposable option that avoids cross-contamination.

In the **laser-induced graphene (LIG)** technique, a computer-controlled laser writes three-dimensional porous graphene patterns directly into polyimide film. F. Zhao et al. (2020) showed that laser power and scanning speed govern LIG morphology: porosity and conductivity rise with power, but very high power also patterns the underlying glass, and very low speed collapses the porous structure. The optimum was 3.6 W and 20 cm/s, and the D, G and 2D Raman bands confirmed graphene formation. The LIG was transferred to a stretchable electrode by casting, curing and peeling off polydimethylsiloxane (PDMS). LIG can produce a set of six sensor electrodes in only 15.6 seconds (Chu et al., 2024).

Biodegradable substrates answer sustainability concerns, since most wearable devices are built on slowly degrading petroleum-based polymers (Teixeira et al., 2023). Teixeira et al. (2026) plasticized cellulose acetate (CA), which can be obtained from agricultural waste, with glycerol (GLY) or triethyl citrate (TEC), cast it into films and screen-printed electrodes onto them. Poor ink adhesion on TEC-plasticized films blurred the voltammetric peaks, whereas films with 5.4 mM glycerol performed best.

3.5. The Solid/Solid Interface and Semi-Solid Electrolytes

What sets wearable pesticide sensors apart from conventional electrochemical sensors is that analysis occurs not in solution but at a solid/solid interface between a dry produce surface and the sensor. An ionic medium is needed to carry the residue to the enzyme layer and to complete the electrochemical cell (F. Zhao et al., 2020).

Mishra et al. (2017) used a 2.5% gelatin gel in KCl and phosphate buffer, adapted from de Jong et al. (2016). Held on the index finger with an O-ring, the gel let the analyte diffuse from the thumb pad to the enzyme layer and supplied the conductivity needed by the cell; it stayed stable for two weeks and gave a smooth, noise-free baseline. F. Zhao et al. (2020) used a gelatin electrolyte about 0.5 mm thick whose natural stickiness held the sensor on leaves and fruits without adhesive. Because hydrogels swell well and adsorb many chemical species, they are effective for transferring poorly water-soluble or apolar contaminants to the sensor (Y. Wang et al., 2026). Teixeira et al. (2026) instead added 500 μL of phosphate buffer to the surface or used a single 150 μL sample drop.

4. Classification of Wearable Pesticide Biosensors

Wearable pesticide biosensors can be classified by wearing format, biorecognition element, transduction principle and substrate/fabrication technique (Figure 2). These axes are interdependent; the physicochemical properties of the analyte, for example, influence both the recognition element and the wearing format (Y. Wang et al., 2026).

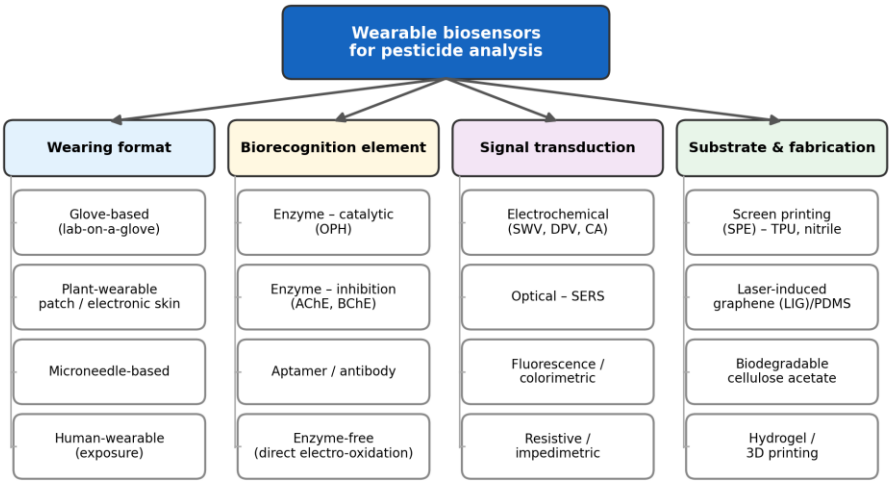


Figure 2. Four-axis classification of wearable biosensors for pesticide analysis (Rehman et al., 2026; Y. Wang et al., 2026). CA: chronoamperometry; DPV: differential pulse voltammetry; SPE: screen-printed electrode; TPU: thermoplastic polyurethane.

4.1. Glove-Based Sensors

Glove-based sensors (“lab-on-a-glove”) turn the fingertip into an analytical platform. Unlike patches, which sample from a fixed position, gloves can scan the whole surface of a fruit or vegetable, and their low cost makes frequent hygienic replacement affordable (Tsong, Robert, & Khor, 2023). Different

fingers can carry different sensors: an enzyme-free glove with three carbon electrodes, modified with carbon spherical shells or porous carbon nanoballs, used the index, middle and ring fingers to determine carbendazim, diuron, paraquat and fenitrothion (Raymundo-Pereira, Gomes, Shimizu, Machado, & Oliveira, 2021).

4.2. Plant-Wearable Patch Sensors

Plant-wearable sensors are flexible, miniaturized devices attached directly to leaves, stems or fruits. First developed to monitor temperature, humidity, growth and disease (Nassar et al., 2018; Paul et al., 2019; Tang, Yan, Ping, Wu, & Ying, 2017), they have only recently been applied to pesticides (Teixeira et al., 2023; F. Zhao et al., 2020). Because non-systemic pesticides largely stay on the surface until they degrade (Chung, 2018), timely measurement of surface residues can guide harvest decisions (F. Zhao et al., 2020). A microfluidic mold that enriched methyl parathion on vegetable surfaces by capillary action shortened measurement time by about 381 seconds (Zhang et al., 2024).

4.3. Microneedle-Based Sensors

Microneedle-based sensors penetrate the cuticle to determine pesticides or metabolites within leaf tissue, reducing interference from surface contamination. Microneedles can, however, break or clog on insertion, lose sensitivity with repeated use and trigger local defense responses, and hydrogel microneedles, though biocompatible, are mechanically weak and relatively expensive to make (Y. Wang et al., 2026).

4.4. Wearable Sensors for Monitoring Human Exposure

Wearable sensors can also monitor personal exposure in high-risk groups such as pesticide applicators and farm workers. Built into gloves, headbands or clothing patches, they can track pesticides or metabolites in sweat, interstitial fluid or surrounding air during daily work (Salamone, Masullo, & Sibilio, 2021; Tsong et al., 2023) and, unlike laboratory blood or urine biomonitoring, give immediate feedback that can help prevent acute poisoning and manage cumulative exposure. Examples include a smartphone-linked resistive nanosensor platform that measures blood cholinesterase activity (Maanaki et al., 2025) and the plant-wearable dual sensors of Teixeira et al. (2026), which were also tested on human saliva.

5. Selected Studies in Detail

Four original studies that trace the development of the field are examined below. The layer architectures of the three OPH-based sensors are compared in Figure 3.

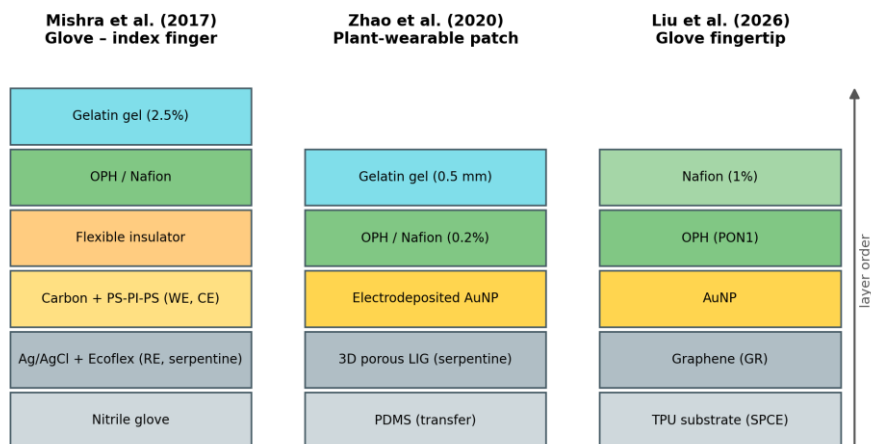


Figure 3. Layer architectures of selected OPH-based wearable biosensors (Liu et al., 2026; Mishra et al., 2017; F. Zhao et al., 2020). Layer thicknesses are not to scale. AuNP: gold nanoparticle; CE: counter electrode; LIG: laser-induced graphene; PON1: paraoxonase-1; RE: reference electrode; WE: working electrode.

5.1. Flexible and Stretchable Glove Biosensor (Mishra et al., 2017)

Mishra et al. (2017) introduced the first glove biosensor to perform an enzymatic assay at the fingertip; earlier “forensic finger” sensors detected explosive and drug residues voltammetrically without an enzymatic step (Bandonkar et al., 2013; de Jong et al., 2016). The index finger served as the “sensing finger”, with smiley-face-shaped carbon working and counter electrodes and an Ag/AgCl reference electrode, and the thumb as the “collecting finger”, with a carbon pad 1 cm in diameter. The suspect surface is first swiped with the thumb to collect residue (swipe); the thumb is then pressed against the gelatin-coated index finger to complete the cell and run the measurement (scan). The working electrode carried 5 μ L of a 7:3 (v/v) mixture of 1% Nafion and OPH.

A handheld wireless potentiostat on the back of the hand, connected to the glove pads with an adjustable ring bandage, sent results to a smartphone. Methyl parathion (MP) and methyl paraoxon (MPOx) residues (from 200 μ M solutions) on glass, wood, stainless steel and plastic and on apple, tomato, grape and green pepper skins gave distinct PNP peaks at +0.85 V. No peak appeared on clean surfaces or with OPH-free control sensors, confirming the enzymatic origin of the signal. Glove-to-glove relative standard deviations of the peak potential were 3.5% (MPOx) and 2.77% (MP) on surfaces and 3.05% (MPOx) and 2.10% (MP) on foods ($n = 12$).

The study tested mechanical durability systematically. Sixty full bending cycles and sixty 50% stretching cycles caused negligible changes in ferricyanide cyclic voltammograms. Dynamic resistance was about $329 \pm 38 \Omega$ flat and $246 \pm$

48 Ω bent over the first 20 cycles and stayed at 430 and 416 Ω after 100 bends; under 100% stretching it changed by no more than 8% after 30–60 cycles, conditions the authors consider far harsher than real use. The aim was a qualitative or semi-quantitative “early warning” rather than quantification. The enzyme remained stable in Nafion for about three days, after which the response gradually declined.

5.2. LIG-Based Smart Plant-Wearable Biosensor (F. Zhao et al., 2020)

F. Zhao et al. (2020) developed one of the first electrochemical plant-wearable biosensors for pesticide analysis on the plant surface. A serpentine three-electrode system was patterned in polyimide by LIG and transferred to PDMS, coated with AuNPs by electrodeposition from 1% H₂AuCl₄ at -0.66 V for 60 seconds and modified with OPH and Nafion. An OPH loading of 0.01 U gave the highest current, since more enzyme hindered electron transfer, and 0.2% Nafion and pH 6 were optimal.

Resistance change remained acceptable at strains above 10%, and voltammograms changed negligibly under 5% stretching. On the curved surfaces of culture dishes of different diameters, at bending angles of about 10°, 16° and 25°, the electrochemical response was preserved. Although its tensile strength is lower than that of some wearable sensors, the authors consider it sufficient for the short-term shape changes of most fruits and vegetables.

Performance at the solid/solid interface was tested on PET films sprayed with methyl parathion and dried. With an optimum reaction time of 8 minutes, the current was linear with the logarithm of concentration over 20–500 μ M ($I = 2.13 \lg C - 2.69$; $R^2 = 0.984$), with a detection limit of 0.01 μ M (2.63 μ g/L). Spinach leaves and apples sprayed with 100 μ M methyl parathion gave distinct PNP peaks absent in controls, but weaker than on PET because the pesticide partly penetrated the cuticle (Chung, 2018); the waxy apple peel hindered penetration, whereas abundant leaf stomata facilitated it. Surface morphology thus affects the result directly, and sensors need crop-specific calibration.

5.3. TPU-Based Screen-Printed Glove Biosensor (Liu et al., 2026)

Liu et al. (2026) modified an SPCE printed on TPU with successive layers of graphene, AuNPs, OPH and Nafion (SPCE/GR/AuNPs/OPH/Nafion) and fixed it to a nitrile glove fingertip. Optimal conditions were 10 μ L graphene (0.2 mg/mL), 10 μ L AuNPs (120 μ g/mL), 7.5 μ L OPH (30 μ g/mL) and 8 minutes of incubation. SEM and EDS confirmed each layer, and the Nafion layer covered surface wrinkles and protected the enzyme.

The response to methyl paraoxon was linear over 30–400 μ M ($R^2 = 0.995$), with a detection limit of 0.321 μ M. Fiftyfold excesses of sucrose, NaCl, MgCl₂, Na₂SO₄ and Na₂CO₃ barely affected the 100 μ M MPOx signal, and recoveries in cabbage and tap water at 20 and 200 μ M were 101–109%. Resistance ranged

between 300 and 350 Ω ; compression and stretching cycles changed it by only 1.3% and 1.7%, respectively, and 100 bending cycles by 1.9–3.14%. Even when immersed while bent through 180°, the electrode gave voltammograms very close to normal.

Surface type strongly affected signal intensity. Among citrus peel, a plastic Petri dish, polystyrene foam and a wooden table, hydrophobic plastic and wood gave the strongest signals, as the solution was retained there and formed a stable liquid film with the electrode; it spread quickly on hydrophilic peel, and porous foam adsorbed the target. With the surface dependence seen by F. Zhao et al. (2020), this shows that the “matrix–solution–electrode” interface cannot be ignored in sensor design.

Two critical points should be noted. According to the supplier, the enzyme called OPH by Liu et al. (2026) is a fragment of recombinant human paraoxonase-1 (PON1) expressed in *E. coli*, and its specific activity was not provided. Moreover, the reported detection limit lies well below the lower end of the linear range (30 μM), as is also the case for F. Zhao et al. (2020). Such extrapolated values should be verified experimentally at low concentrations if studies are to be compared fairly.

5.4. Biodegradable Dual-Sensor Platform (Teixeira et al., 2026)

Teixeira et al. (2026) developed a biodegradable “dual sensor” with two complete electrochemical cells screen-printed on plasticized cellulose acetate. One channel determines diquat by SWV (−0.73 V); the other determines carbendazim (+0.49 V) and diphenylamine (+0.27 V) simultaneously by DPV. A two-channel handheld analyzer linked by Bluetooth to a smartphone, tablet or laptop switches between techniques automatically. Diquat is measured in the first 52 seconds and the other two analytes in a further 197 seconds, so all three are determined from one microliter-scale drop in 3.28 minutes.

Diquat and carbendazim each showed two linear ranges (0.1–0.5 and 0.6–1.0 μM ; 0.2–1.4 and 1.4–2.0 μM), and diphenylamine one (2.5–25 μM). Detection limits of 3.2 nM, 0.18 μM and 1.34 μM for diquat, carbendazim and diphenylamine, respectively, were reported to be below the maximum residue limits of the European Commission and FAO/WHO. No analyte altered the calibration slopes of the others. Relative standard deviations of 6.1–8.2% over twelve consecutive measurements suggest partial electrode fouling, which the disposable design sidesteps; reproducibility across five platforms was 3.7–7.0%.

In standard addition tests on apple and pepper skin, saliva and tap water, the largest recovery deviations were 6%, 3% and 11% for diquat, carbendazim and diphenylamine, respectively. In an on-site scenario, a 1000 μM pesticide solution was sprayed on produce skin and dried for 5 hours; the sensor was placed on the skin with 500 μL of buffer, and all three pesticides were determined without

cross-channel interference. Responses changed negligibly after 50, 100 and 200 bending cycles in vertical, horizontal and diagonal directions. The cost per sensor was below USD 0.077. Other wearable pesticide sensors are summarized in Table 2.

Table 2. Selected examples of other wearable pesticide sensors reported in the literature

Platform and material	Wearing format	Analyte	Technique / performance	Reference
Carbon spherical shells / porous carbon nanoballs, multi-finger	Glove	Carbendazim, diuron, paraquat, fenitrothion	DPV/SWV; LOD 2.4×10^{-8} – 9.2×10^{-7} mol/L	Raymundo-Pereira et al. (2021)
Silk fibroin/AAO/Au	Glove	Thiram	SERS; LOD 0.168 μ M	D. Wang et al. (2022)
PVA/HA tiered microneedle	Microneedle	Thiram, thiabendazole	SERS; LOD 0.1 and 0.01 μ M	Yi et al. (2023)
CdTe quantum dots @ PVA/agarose hydrogel	Surface	Thiram	Fluorescence, smartphone RGB; LOD 0.27 μ M	Chen et al. (2025)
Microfluidic mold + LIG/AuNP/lipase	Surface	Methyl parathion	DPV; 0.001–200 μ M; measurement time shortened by about 381 s	Zhang et al. (2024)
Screen printing on cellulose acetate	Surface	Carbendazim, paraquat	DPV/SWV; LOD 54.9 and 19.8 nM	Teixeira et al. (2023)
Prussian blue + carbon black + BChE	Glove	Dichlorvos	Electrochemical; low-nanomolar LOD	Miglione et al. (2025)

AAO: anodic aluminum oxide; BChE: butyrylcholinesterase; HA: hyaluronic acid; PVA: polyvinyl alcohol; SERS: surface-enhanced Raman scattering.

6. Sustainability and Effects on Plant Physiology

6.1. Biodegradability and Green Analytical Chemistry

Disposable sensors avoid cross-contamination and enzyme instability, but their widespread use could generate considerable electronic and plastic waste. In soil, sensors on glycerol-plasticized CA disintegrated completely within 240 days, while those on neat CA remained intact (Teixeira et al., 2026), so the plasticizer governs both electrochemical performance and degradation time.

The same study scored the method 0.77 on AGREE, which is based on the twelve principles of green analytical chemistry, 81 on the Analytical Eco-Scale and 77.5 on the Blue Applicability Grade Index (BAGI) for practicality (Teixeira et al., 2026). Routine reporting of such metrics would make “green” claims comparable. Biodegradable substrates such as polylactic acid also suit sustainable agriculture (Y. Wang et al., 2026).

6.2. Possible Effects on Plant Physiology

Poorly designed plant-wearable sensors can disturb photosynthesis and transpiration: an opaque or thick device can shade the leaf, partial stomatal closure can restrict water vapor flux and CO₂ diffusion into the mesophyll, and heat and moisture trapped in the boundary layer can alter the local microclimate (Lo Presti et al., 2024; Qu et al., 2021). Ultrathin, light, transparent and gas-permeable materials, intermittent measurement and non-invasive attachment are recommended (Lo Presti et al., 2024; Qu et al., 2021; Y. Wang et al., 2026). Most pesticide sensors are used for brief measurements, which limits the risk, but compatibility with plant physiology will become essential once long-term continuous monitoring is the goal.

7. Challenges and Limitations

Despite impressive laboratory results, several interconnected challenges stand in the way of routine field use.

- **Stability of the biorecognition element:** Enzymes are sensitive to temperature, humidity and pH. The OPH/Nafion layer of Mishra et al. (2017) stayed stable for only about three days, illustrating the shelf-life problem. Thermostable enzymes (Luo et al., 2014; F. Zhao et al., 2020) and enzyme-free approaches (Teixeira et al., 2026) offer two routes forward.
- **Surface and matrix effects:** The signal varies considerably with waxy, hydrophilic or porous surfaces (Liu et al., 2026; F. Zhao et al., 2020), so quantification needs crop-specific calibration. Transferring poorly water-soluble pesticides to the sensor is a further difficulty (Y. Wang et al., 2026).

- **Multiple residues and selectivity:** Real produce usually carries several pesticides. OPH- and cholinesterase-based sensors respond to a pesticide class but generally cannot distinguish its members. Multichannel designs (Teixeira et al., 2026) are promising but still fall short of chromatographic multi-analyte capacity.
- **Environmental robustness:** Wind-induced vibration, raindrops and daily temperature swings can dislodge or crack sensors or cause signal drift (Y. Wang et al., 2026); encapsulation improves durability (Nassar et al., 2018) but can slow the response (Nguyen, Chu, Tu, & Khine, 2021).
- **Data transmission and power:** Bluetooth and ZigBee signals are strongly attenuated in dense vegetation (Y. Zhao et al., 2019), while in long-range low-power networks such as LoRa, trade-offs among power use, device size and cost limit large-scale deployment (Singh, Aernouts, De Meyer, Weyn, & Berkvens, 2020; Y. Wang et al., 2026).
- **Cost and scalability:** Advanced sensors are expensive relative to the low economic value of individual produce items, which hampers commercialization (Y. Wang et al., 2026). Per-sensor costs of a few cents (Teixeira et al., 2026) are a real advance, but potentiostat and reader electronics must be included in the total system cost.
- **Validation and standardization:** Studies differ in how they calculate detection limits, test mechanical performance and validate with real samples, and few have checked results against parallel measurements by accredited chromatographic methods.

Y. Wang et al. (2026) argue that these challenges are interrelated and call for system-level solutions rather than incremental gains in single device parameters. Field use prioritizes stability, efficiency and scalability, whereas laboratory and greenhouse work focuses on sensitivity and functionality, leaving a gap between basic research and agricultural practice.

8. Future Perspectives

Since 2017 the field has moved from single-analyte prototypes with proven mechanical durability toward biodegradable, multichannel platforms, and further progress is expected in the following directions.

Multiplexed and multimodal sensing. Simultaneous analysis of several residues with a single device is considered essential for dealing with real-world mixture exposures (Kapoor et al., 2024; Murugesan, Raj, & Moses, 2023). For plant health monitoring, measuring residues together with strain, microclimate and biochemical markers, combined with machine-learning-assisted data fusion,

could produce a comprehensive “physiological fingerprint” of the plant (Y. Wang et al., 2026).

More stable recognition elements. Genetically engineered enzymes, synthetic aptamers and molecularly imprinted polymers that withstand harsh conditions are being developed to extend shelf life (Rehman et al., 2026). F. Zhao et al. (2020) likewise name more stable recognition elements, suitable for monitoring throughout the growth period, as a priority.

Integration with the exposome approach. The exposome covers all environmental exposures of an individual from conception onward, together with the internal biological responses to them (Vrijheid, 2014). Rehman et al. (2026) argue that integrating wearable biosensors into this framework, alongside human biomonitoring, wastewater-based epidemiology and omics, would link external exposure to internal effect more accurately. Plant-wearable sensors measuring residues on produce and human-wearable sensors tracking a farm worker’s exposure would then be complementary links in one risk assessment chain.

Low-cost and sustainable fabrication. Inexpensive materials such as carbon nanomaterials and natural fibers, processed by screen printing, 3D printing or roll-to-roll methods, could make mass production feasible (Y. Wang et al., 2026), and combining these routes with biodegradable substrates (Teixeira et al., 2026) would greatly reduce the footprint of disposable sensors.

9. Conclusion

By moving pesticide analysis from the central laboratory to the fingertip and the produce surface, wearable biosensors open a new route for food safety, precision agriculture and occupational health. The studies reviewed here show that glove sensors with stress-enduring inks and serpentine geometries withstand extreme deformation (Mishra et al., 2017), that LIG/PDMS structures allow in situ analysis on plants (F. Zhao et al., 2020), that nanomaterial-modified TPU electrodes yield low-cost, durable glove sensors (Liu et al., 2026), and that several pesticides can be determined from one drop within minutes on biodegradable substrates (Teixeira et al., 2026).

Gaps remain in recognition-element stability, surface and matrix effects, multi-residue analysis, environmental robustness, data transmission and standardization. At their current maturity, these sensors are best positioned as rapid screening and early-warning tools rather than replacements for accredited chromatographic methods. With closer collaboration among materials science, electrochemistry, plant physiology, data science and toxicology, wearable pesticide biosensors can be expected to become an integral part of Internet of Things and AI-supported smart farming networks and of exposome-based risk assessment.

References

- Bakanov, N., Honert, C., Eichler, L., Lehmann, G. U. C., Schulz, R., & Brühl, C. A. (2023). A new sample preparation approach for the analysis of 98 current-use pesticides in soil and herbaceous vegetation using HPLC-MS/MS in combination with an acetonitrile-based extraction. *Chemosphere*, 331, 138840. doi:10.1016/j.chemosphere.2023.138840
- Bandodkar, A. J., Jia, W., Yardımcı, C., Wang, X., Ramirez, J., & Wang, J. (2015). Tattoo-based noninvasive glucose monitoring: A proof-of-concept study. *Analytical Chemistry*, 87(1), 394–398. doi:10.1021/ac504300n
- Bandodkar, A. J., O'Mahony, A. M., Ramírez, J., Samek, I. A., Anderson, S. M., Windmiller, J. R., & Wang, J. (2013). Solid-state forensic finger sensor for integrated sampling and detection of gunshot residue and explosives: Towards “lab-on-a-finger.” *Analyst*, 138(18), 5288–5295. doi:10.1039/c3an01179h
- Berkal, M. A., & Nardin, C. (2023). Pesticide biosensors: Trends and progresses. *Analytical and Bioanalytical Chemistry*, 415(24), 5899–5924. doi:10.1007/s00216-023-04911-4
- Chen, J., Wang, C., Huang, X., Wan, R., Zhu, Z., Sun, G., . . . Chi, Z. (2025). Quantum dots-engineered flexible hydrogel as plant-wearable sensor for on-site profiling dynamic pesticide degradation. *Advanced Functional Materials*, 35(23), 2423643. doi:10.1002/adfm.202423643
- Christopher, F. C., Kumar, P. S., Christopher, F. J., Joshiba, G. J., & Madhesh, P. (2020). Recent advancements in rapid analysis of pesticides using nano biosensors: A present and future perspective. *Journal of Cleaner Production*, 269, 122356. doi:10.1016/j.jclepro.2020.122356
- Chu, G., Yu, S., Wang, Y., Zhang, W., Xue, B., Huang, J., . . . Li, M. (2024). A wearable electrochemical sensor for the monitoring of neonicotinoid insecticides, salicylic acid and the pH in plant guttation. *Chemical Engineering Journal*, 498, 155340. doi:10.1016/j.cej.2024.155340
- Chung, S. W. C. (2018). How effective are common household preparations on removing pesticide residues from fruit and vegetables? A review. *Journal of the Science of Food and Agriculture*, 98(8), 2857–2870. doi:10.1002/jsfa.8821
- Damalas, C. A., & Eleftherohorinos, I. G. (2011). Pesticide exposure, safety issues, and risk assessment indicators. *International Journal of Environmental Research and Public Health*, 8(5), 1402–1419. doi:10.3390/ijerph8051402
- de Jong, M., Slegers, N., Kim, J., Van Durme, F., Samyn, N., Wang, J., & De Wael, K. (2016). Electrochemical fingerprint of street samples for fast on-site screening of cocaine in seized drug powders. *Chemical Science*, 7(3), 2364–2370. doi:10.1039/c5sc04309c

- Dutta, A., Tyagi, R., Chattopadhyay, A., Chatterjee, D., Sarkar, A., Lall, B., & Sharma, S. (2024). Early detection of wilt in *Cajanus cajan* using satellite hyperspectral images: Development and validation of disease-specific spectral index with integrated methodology. *Computers and Electronics in Agriculture*, 219, 108784. doi:10.1016/j.compag.2024.108784
- Giraldo, J. P., Wu, H., Newkirk, G. M., & Kruss, S. (2019). Nanobiotechnology approaches for engineering smart plant sensors. *Nature Nanotechnology*, 14(6), 541–553. doi:10.1038/s41565-019-0470-6
- Han, Q., Wang, H., & Wang, J. (2024). Multi-mode/signal biosensors: Electrochemical integrated sensing techniques. *Advanced Functional Materials*, 34(39), 2403122. doi:10.1002/adfm.202403122
- Harischandra Naik, R., Pallavi, M. S., Pavan Kumar, K., Vanitha, B. K., Chandra Sekhara Reddy, V., Shwetha, A., . . . Bheemanna, M. (2021). Determination of 72 chemical pesticides and estimation of measurement of uncertainty in rice using LC-MS/MS and GC-MS/MS. *Food Analytical Methods*, 14(9), 1788–1805. doi:10.1007/s12161-021-02000-9
- Hernandez-Vargas, G., Sosa-Hernández, J. E., Saldarriaga-Hernandez, S., Villalba-Rodríguez, A. M., Parra-Saldivar, R., & Iqbal, H. M. N. (2018). Electrochemical biosensors: A solution to pollution detection with reference to environmental contaminants. *Biosensors*, 8(2), 29. doi:10.3390/bios8020029
- Jia, W., Bandodkar, A. J., Valdés-Ramírez, G., Windmiller, J. R., Yang, Z., Ramírez, J., . . . Wang, J. (2013). Electrochemical tattoo biosensors for real-time noninvasive lactate monitoring in human perspiration. *Analytical Chemistry*, 85(14), 6553–6560. doi:10.1021/ac401573r
- Kapoor, A., Ramamoorthy, S., Sundaramurthy, A., Vaishampayan, V., Sridhar, A., Balasubramanian, S., & Ponnuchamy, M. (2024). Paper-based lab-on-a-chip devices for detection of agri-food contamination. *Trends in Food Science & Technology*, 147, 104476. doi:10.1016/j.tifs.2024.104476
- Kim, J., Campbell, A. S., de Ávila, B. E.-F., & Wang, J. (2019). Wearable biosensors for healthcare monitoring. *Nature Biotechnology*, 37(4), 389–406. doi:10.1038/s41587-019-0045-y
- Kim, J., Jeerapan, I., Imani, S., Cho, T. N., Bandodkar, A., Cinti, S., . . . Wang, J. (2016). Noninvasive alcohol monitoring using a wearable tattoo-based iontophoretic-biosensing system. *ACS Sensors*, 1(8), 1011–1019. doi:10.1021/acssensors.6b00356
- Lan, W., Chen, G., Cui, F., Tan, F., Liu, R., & Yushupujiang, M. (2012). Development of a novel optical biosensor for detection of organophosphorus pesticides based on methyl parathion hydrolase immobilized by metal-chelate affinity. *Sensors*, 12(7), 8477–8490. doi:10.3390/s120708477

- Li, X.-H., Li, M.-Z., Li, J.-Y., Gao, Y.-Y., Liu, C.-R., & Hao, G.-F. (2024). Wearable sensor supports in-situ and continuous monitoring of plant health in precision agriculture era. *Plant Biotechnology Journal*, 22(6), 1516–1535. doi:10.1111/pbi.14283
- Liu, Z., Zhu, H., Yang, K., Liu, Z., Yang, X., Ze, Y., . . . Chen, Z. (2026). A wearable electrochemical sensing platform for rapid detection of organophosphorus pesticides: A flexible biosensor based on screen-printed electrodes and organophosphorus hydrolase. *Sensors*, 26(8), 2348. doi:10.3390/s26082348
- Lo Presti, D., Cimini, S., De Tommasi, F., Massaroni, C., Cinti, S., De Gara, L., & Schena, E. (2024). Flexible matrices for the encapsulation of plant wearable sensors: Influence of geometric and color features on photosynthesis and transpiration. *Sensors*, 24(5), 1611. doi:10.3390/s24051611
- Luo, X.-J., Kong, X.-D., Zhao, J., Chen, Q., Zhou, J., & Xu, J.-H. (2014). Switching a newly discovered lactonase into an efficient and thermostable phosphotriesterase by simple double mutations His250Ile/Ile263Trp. *Biotechnology and Bioengineering*, 111(10), 1920–1930. doi:10.1002/bit.25272
- Maanaki, H., Bussiere, L., Smirnov, A., Du, X., Sun, Y., Arcury, T. A., . . . Wang, J. (2025). An integrated nanosensor/smartphone platform for point-of-care biomonitoring of human exposure to pesticides. *Analytical Chemistry*, 97(18), 9701–9712. doi:10.1021/acs.analchem.4c06421
- Malzahn, K., Windmiller, J. R., Valdés-Ramírez, G., Schöning, M. J., & Wang, J. (2011). Wearable electrochemical sensors for in situ analysis in marine environments. *Analyst*, 136(14), 2912–2917. doi:10.1039/c1an15193b
- Marimuthu, M., Xu, K., Song, W., Chen, Q., & Wen, H. (2025). Safeguarding food safety: Nanomaterials-based fluorescent sensors for pesticide tracing. *Food Chemistry*, 463, 141288. doi:10.1016/j.foodchem.2024.141288
- Miglione, A., Raucci, A., Mancini, M., Gioia, V., Frugis, A., & Cinti, S. (2025). An electrochemical biosensor for on-glove application: Organophosphorus pesticide detection directly on fruit peels. *Talanta*, 283, 127093. doi:10.1016/j.talanta.2024.127093
- Mishra, R. K., Hubble, L. J., Martín, A., Kumar, R., Barfidokht, A., Kim, J., . . . Wang, J. (2017). Wearable flexible and stretchable glove biosensor for on-site detection of organophosphorus chemical threats. *ACS Sensors*, 2(4), 553–561. doi:10.1021/acssensors.7b00051
- Murugesan, P., Raj, G., & Moses, J. A. (2023). Microfluidic devices for the detection of pesticide residues. *Reviews in Environmental Science and Bio/Technology*, 22(3), 625–652. doi:10.1007/s11157-023-09664-1
- Nassar, J. M., Khan, S. M., Villalva, D. R., Nour, M. M., Almuslem, A. S., & Hussain, M. M. (2018). Compliant plant wearables for localized microclimate and

- plant growth monitoring. *npj Flexible Electronics*, 2(1), 24. doi:10.1038/s41528-018-0039-8
- Nguyen, T., Chu, M., Tu, R., & Khine, M. (2021). The effect of encapsulation on crack-based wrinkled thin film soft strain sensors. *Materials*, 14(2), 364. doi:10.3390/ma14020364
- Niu, Y., Tian, G., Liang, C., Wang, T., Ma, X., Gong, G., & Qi, D. (2023). Thermal-sinterable EGaIn nanoparticle inks for highly deformable bioelectrode arrays. *Advanced Healthcare Materials*, 12(10), e2202531. doi:10.1002/adhm.202202531
- Ottenbros, I., Lebre, E., Huber, C., Lommen, A., Antignac, J.-P., Čupr, P., . . . Vlaanderen, J. (2023). Assessment of exposure to pesticide mixtures in five European countries by a harmonized urinary suspect screening approach. *International Journal of Hygiene and Environmental Health*, 248, 114105. doi:10.1016/j.ijheh.2022.114105
- Pachapur, P. K., Larios Martínez, A. D., Pulicharla, R., Pachapur, V. L., Brar, S. K., & Galvez-Cloutier, R. (2019). Advances in protein/enzyme-based biosensors for the detection of pesticide contaminants in the environment. In S. K. Brar, K. Hegde, & V. L. Pachapur (Eds.), *Tools, techniques and protocols for monitoring environmental contaminants* (pp. 231–243). Amsterdam, Netherlands: Elsevier. doi:10.1016/B978-0-12-814679-8.00011-X
- Paul, R., Saville, A. C., Hansel, J. C., Ye, Y., Ball, C., Williams, A., . . . Wei, Q. (2019). Extraction of plant DNA by microneedle patch for rapid detection of plant diseases. *ACS Nano*, 13(6), 6540–6549. doi:10.1021/acsnano.9b00193
- Pimentel, D. (2009). Pesticides and pest control. In R. Peshin & A. K. Dhawan (Eds.), *Integrated pest management: Innovation-development process* (pp. 83–87). Dordrecht, Netherlands: Springer. doi:10.1007/978-1-4020-8992-3_3
- Qu, C.-C., Sun, X.-Y., Sun, W.-X., Cao, L.-X., Wang, X.-Q., & He, Z.-Z. (2021). Flexible wearables for plants. *Small*, 17(50), 2104482. doi:10.1002/smll.202104482
- Raj, A., Dubey, A., Malla, M. A., & Kumar, A. (2023). Pesticide pestilence: Global scenario and recent advances in detection and degradation methods. *Journal of Environmental Management*, 338, 117680. doi:10.1016/j.jenvman.2023.117680
- Rapini, R., & Marrazza, G. (2016). Biosensor potential in pesticide monitoring. In V. Scognamiglio, G. Rea, F. Arduini, & G. Palleschi (Eds.), *Comprehensive analytical chemistry: Vol. 74. Biosensors for sustainable food: New opportunities and technical challenges* (pp. 3–31). Amsterdam, Netherlands: Elsevier. doi:10.1016/bs.coac.2016.03.016
- Raymundo-Pereira, P. A., Gomes, N. O., Shimizu, F. M., Machado, S. A. S., & Oliveira, O. N., Jr. (2021). Selective and sensitive multiplexed detection of

- pesticides in food samples using wearable, flexible glove-embedded non-enzymatic sensors. *Chemical Engineering Journal*, 408, 127279. doi:10.1016/j.cej.2020.127279
- Rehman, M. U., Sehar, N., Elboughdiri, N., Nisar, B., Naqash, N., Sheikh, W. M., . . . Wani, A. K. (2026). Analytical advances in pesticide toxicology: Integrating omics, biosensors, microfluidics and exposome-based monitoring for risk assessment. *Microchemical Journal*, 220, 116373. doi:10.1016/j.microc.2025.116373
- Romagnoli, M., Scarparo, A., Catani, M., Gianni, B., Pasti, L., Cavazzini, A., & Franchina, F. A. (2023). Development and validation of a GC × GC-ToFMS method for the quantification of pesticides in environmental waters. *Analytical and Bioanalytical Chemistry*, 415(18), 4545–4555. doi:10.1007/s00216-023-04686-8
- Salamone, F., Masullo, M., & Sibilio, S. (2021). Wearable devices for environmental monitoring in the built environment: A systematic review. *Sensors*, 21(14), 4727. doi:10.3390/s21144727
- Samsidar, A., Siddiquee, S., & Shaarani, S. M. (2018). A review of extraction, analytical and advanced methods for determination of pesticides in environment and foodstuffs. *Trends in Food Science & Technology*, 71, 188–201. doi:10.1016/j.tifs.2017.11.011
- Sharma, D., Nagpal, A., Pakade, Y. B., & Katnoria, J. K. (2010). Analytical methods for estimation of organophosphorus pesticide residues in fruits and vegetables: A review. *Talanta*, 82(4), 1077–1089. doi:10.1016/j.talanta.2010.06.043
- Singh, R., Aernouts, M., De Meyer, M., Weyn, M., & Berkvens, R. (2020). Leveraging LoRaWAN technology for precision agriculture in greenhouses. *Sensors*, 20(7), 1827. doi:10.3390/s20071827
- Tang, W., Yan, T., Ping, J., Wu, J., & Ying, Y. (2017). Rapid fabrication of flexible and stretchable strain sensor by chitosan-based water ink for plants growth monitoring. *Advanced Materials Technologies*, 2(7), 1700021. doi:10.1002/admt.201700021
- Tanveer, S., Ilyas, N., Akhtar, N., Akhtar, N., Bostan, N., Hasnain, Z., . . . Fitriatin, B. N. (2024). Unlocking the interaction of organophosphorus pesticide residues with ecosystem: Toxicity and bioremediation. *Environmental Research*, 249, 118291. doi:10.1016/j.envres.2024.118291
- Teixeira, S. C., Gomes, N. O., Calegaro, M. L., Machado, S. A. S., de Oliveira, T. V., de Fátima Ferreira Soares, N., & Raymundo-Pereira, P. A. (2023). Sustainable plant-wearable sensors for on-site, rapid decentralized detection of pesticides toward precision agriculture and food safety. *Biomaterials Advances*, 155, 213676. doi:10.1016/j.bioadv.2023.213676

- Teixeira, S. C., Gomes, N. O., de Oliveira, T. V., Soares, N. F. F., & Raymundo-Pereira, P. A. (2025). Sustainable wearable sensors for plant monitoring and precision agriculture. *Analytical Chemistry*, 97(28), 14875–14884. doi:10.1021/acs.analchem.5c01565
- Teixeira, S. C., Gomes, N. O., Machado, S. A. S., de Oliveira, T. V., Soares, N. F. F., & Raymundo-Pereira, P. A. (2026). Biodegradable wearable sensors for rapid non-destructive analysis of pesticides on plants and foods. *Biosensors and Bioelectronics: X*, 29, 100758. doi:10.1016/j.biosx.2026.100758
- Tsong, J. L., Robert, R., & Khor, S. M. (2023). Emerging trends in wearable glove-based sensors: A review. *Analytica Chimica Acta*, 1262, 341277. doi:10.1016/j.aca.2023.341277
- Viveros, L., Paliwal, S., McCrae, D., Wild, J., & Simonian, A. (2006). A fluorescence-based biosensor for the detection of organophosphate pesticides and chemical warfare agents. *Sensors and Actuators B: Chemical*, 115(1), 150–157. doi:10.1016/j.snb.2005.08.032
- Vrijheid, M. (2014). The exposome: A new paradigm to study the impact of environment on health. *Thorax*, 69(9), 876–878. doi:10.1136/thoraxjnl-2013-204949
- Wang, D., Xu, G., Zhang, X., Gong, H., Jiang, L., Sun, G., . . . Liang, X. (2022). Dual-functional ultrathin wearable 3D particle-in-cavity SF-AAO-Au SERS sensors for effective sweat glucose and lab-on-glove pesticide detection. *Sensors and Actuators B: Chemical*, 359, 131512. doi:10.1016/j.snb.2022.131512
- Wang, Y., Yu, S., Chu, G., Huang, J., & Li, M. (2026). Wearable sensors: In situ detection tools for plant physiological markers and growth environment information. *Microchimica Acta*, 193(1), 38. doi:10.1007/s00604-025-07763-0
- Wu, W., Guo, N., Zhang, Y., Liu, G., Yu, L., Ma, X., . . . Chen, M. (2023). The fabrication, properties, and application of a printed green Ag NWs-based flexible electrode and circuit. *ACS Applied Materials & Interfaces*, 15(11), 14643–14653. doi:10.1021/acsami.2c20270
- Yao, H., Liao, Y., Lingley, A. R., Afanasiev, A., Lähdesmäki, I., Otis, B. P., & Parviz, B. A. (2012). A contact lens with integrated telecommunication circuit and sensors for wireless and continuous tear glucose monitoring. *Journal of Micromechanics and Microengineering*, 22(7), 075007. doi:10.1088/0960-1317/22/7/075007
- Yi, X., Yuan, Z., Yu, X., Zheng, L., & Wang, C. (2023). Novel microneedle patch-based surface-enhanced Raman spectroscopy sensor for the detection of pesticide residues. *ACS Applied Materials & Interfaces*, 15(4), 4873–4882. doi:10.1021/acsami.2c17954

- Zhang, Q., Ma, S., Meng, W., Zheng, Y., Yin, L., Wang, H., . . . Su, S. (2024). Smartphone-based plant-wearable microfluidic sensor with self driven electrolyte for in-situ detection of methyl parathion. *Sensors and Actuators B: Chemical*, 418, 136254. doi:10.1016/j.snb.2024.136254
- Zhao, F., He, J., Li, X., Bai, Y., Ying, Y., & Ping, J. (2020). Smart plant-wearable biosensor for in-situ pesticide analysis. *Biosensors and Bioelectronics*, 170, 112636. doi:10.1016/j.bios.2020.112636
- Zhao, Y., Gao, S., Zhu, J., Li, J., Xu, H., Xu, K., . . . Huang, X. (2019). Multifunctional stretchable sensors for continuous monitoring of long-term leaf physiology and microclimate. *ACS Omega*, 4(5), 9522–9530. doi:10.1021/acsomega.9b01035
- Zhou, C., Cheng, H., Wu, Y., Zhang, J., Li, D., & Pan, C. (2022). Bensulfuron-methyl, terbutylazine, and 2,4-D butylate disturb plant growth and resistance by deteriorating rhizosphere environment and plant secondary metabolism in wheat seedlings. *Journal of Agricultural and Food Chemistry*, 70(40), 12796–12806. doi:10.1021/acs.jafc.2c03126
- Zhu, R., Azzarelli, J. M., & Swager, T. M. (2016). Wireless hazard badges to detect nerve-agent simulants. *Angewandte Chemie*, 128(33), 9814–9818. doi:10.1002/ange.201604431

CHAPTER 8

Spiking Neuron Models: Dynamics, Firing Patterns, and Hardware Considerations

Mert Opoz¹

Tuna Goksu²

¹ Res. Asst., Department of Electrical and Electronics Engineering, Isparta University of Applied Sciences, Isparta, Turkey, ORCID: 0009-0002-8409-099X

² Asst. Prof., Department of Electrical and Electronics Engineering, Isparta University of Applied Sciences, Isparta, Turkey, ORCID: 0000-0002-9028-6422

1. Introduction

Artificial intelligence systems have evolved through different generations according to the representation and processing of information. The first generation was based on binary neuron models in which inputs and outputs were represented by discrete values. The McCulloch–Pitts neuron, introduced in 1943, described neural activity according to an all-or-none principle and represented each neuron as a logical computational unit (McCulloch & Pitts, 1943). Although this model did not aim to reproduce the biological dynamics of neurons, it established an important conceptual foundation for artificial neural computation. Networks composed of McCulloch–Pitts neurons can implement basic logical operations and, in principle, arbitrary digital functions (Maass, 1997; McCulloch & Pitts, 1943).

The second generation of neural networks introduced continuous-valued inputs, outputs, and activation functions. Artificial neural networks employ nonlinear activation functions to learn complex relationships that cannot be represented by a single linear transformation (Maass, 1997; Rasamoelina, Adjailia, & Sinčák, 2020; Rashid, 2016; Sharma, Sharma, & Athaiya, 2017). Functions such as sigmoid, rectified linear unit, and their variants have therefore become important components of modern neural network architectures (Nair & Hinton, 2010; Narayan, 1997; Qiumei, Dan, & Fenghua, 2019; Rasamoelina et al., 2020; Sharma et al., 2017). These networks have achieved considerable success in areas such as image recognition, natural language processing, and medical data analysis (Alqahtani et al., 2025; Kim, Hwang, & Kamyod, 2022; Lin et al., 2019; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013; Perez & Gutierrez, 2025; Tian, 2020; Zhang, Gu, He, Lin, & Zheng, 2025). However, the increasing computational and memory requirements of large neural networks have also raised concerns regarding energy consumption and implementation cost (Liu et al., 2017; Sabbella et al., 2025).

Spiking neural networks represent a third generation of neural computation in which information is conveyed through discrete spikes distributed over time (Maass, 1997). Unlike conventional artificial neural networks, SNNs directly incorporate the temporal dimension into neuronal communication and processing. A neuron generally emits a spike when its membrane potential exceeds a predefined threshold, while remaining inactive when the accumulated stimulation is insufficient (Gerstner, 1999; Gerstner, Kistler, Naud, & Paninski, 2014; Güttig, 2014; Izhikevich, 2007; Schuman et al., 2017). Therefore, both the occurrence and timing of spikes can contribute to information representation.

The event-driven and sparse nature of spike communication is one of the main motivations for SNN research. Since neurons and synapses are not necessarily active at every computational step, SNNs can reduce redundant operations and provide efficient information processing, particularly on architectures that

support parallel and event-driven computation (Bezugam, Shaban, & Suri, 2023; Fang, Shrestha, Zhao, Li, & Qiu, 2019; Garcia-Palencia et al., 2025; Isik, 2023; Pfeiffer & Pfeil, 2018; Pham et al., 2021; Rahman & Yusoff, 2025; Schuman et al., 2017). In addition, spike-based communication enables the processing of spatiotemporal information and is therefore suitable for applications in which temporal structure carries meaningful information (Ahmed, Yusob, & Hamed, 2014; Ferster & Spruston, 2005; Garcia-Palencia et al., 2025; Lee, 2025). However, the discontinuous nature of spike generation also creates challenges related to training, parameter selection, numerical simulation, and hardware implementation (Ahmed et al., 2014; Pfeiffer & Pfeil, 2018; Rahman & Yusoff, 2025).

A central component of an SNN is the mathematical neuron model used to describe membrane dynamics and spike generation. Different neuron models provide different balances among biological plausibility, mathematical complexity, computational efficiency, and firing-pattern diversity. Detailed conductance-based models such as the Hodgkin–Huxley model reproduce ionic mechanisms with high biological accuracy (Hodgkin & Huxley, 1952), whereas simplified models such as Integrate-and-Fire and Leaky Integrate-and-Fire mainly represent membrane integration, threshold-based firing, and reset behavior (Abbott, 1999; Burkitt, 2006; Gerstner et al., 2014; Lopicque, 1907). Other models, including Morris–Lecar, Hindmarsh–Rose, and Izhikevich, provide intermediate solutions capable of reproducing different firing and bursting behaviors with lower complexity than detailed physiological models (Hindmarsh & Rose, 1984; Izhikevich, 2003; Morris & Lecar, 1981).

The selection of a neuron model therefore depends strongly on the intended application. Studies focusing on membrane physiology may require detailed conductance-based formulations, whereas applications emphasizing firing diversity, nonlinear dynamics, large-scale simulation, or real-time processing may benefit from simpler models (Izhikevich, 2004). In addition, hardware-oriented SNN implementations introduce further design considerations such as arithmetic complexity, numerical representation, memory requirements, resource utilization, and parallel processing capability (Isik, 2023; Pham et al., 2021; Schuman et al., 2017).

This chapter presents the biological foundations of spike generation and reviews commonly used mathematical neuron models for spiking neural networks. The Hodgkin–Huxley, Morris–Lecar, Hindmarsh–Rose, Izhikevich, Integrate-and-Fire, and Leaky Integrate-and-Fire models are examined with emphasis on their fundamental dynamics, computational characteristics, and firing capabilities. Common neuronal firing patterns are then discussed, followed by a comparative evaluation of the considered models. In addition, application-oriented neuron model selection and major hardware implementation

considerations are addressed. The overall aim is to provide a structured overview that supports the selection of an appropriate neuron model according to biological requirements, firing behavior, computational constraints, and implementation objectives.

2. Biological Foundations of Spike Generation

A biological neuron is a specialized cell of the nervous system that receives, processes, and transmits electrochemical signals (Stevens, 1979; Yamazaki, Vo-Ho, Bulsara, & Le, 2022). Although neurons may exhibit different structural forms, a typical neuron consists of three main components: the dendrites, the soma, and the axon (Gerstner et al., 2014; Schuman et al., 2017; Stevens, 1979). Dendrites receive signals from other neurons, the soma integrates these signals, and the axon transmits the generated electrical activity toward other cells (Izhikevich, 2007; Stevens, 1979). Communication between neurons occurs through junctions called synapses, which connect the axon terminal of a presynaptic neuron to the dendrite, soma, or axon of a postsynaptic neuron (Garcia-Palencia et al., 2025; Izhikevich, 2007; Stevens, 1979).

The neuronal membrane separates the intracellular and extracellular environments and plays a central role in electrical signaling. Different concentrations of ions such as sodium, potassium, calcium, and chloride exist on the two sides of the membrane (Dayan & Abbott, 2003; Izhikevich, 2007). This unequal ionic distribution produces an electrical potential difference known as the membrane potential. In a resting neuron, the intracellular potential is typically negative with respect to the extracellular environment, and a commonly cited resting value is approximately -70 mV (Dahasert, Öztürk, & Kiliç, 2012; Dayan & Abbott, 2003; Gerstner et al., 2014; Stevens, 1979).

Changes in membrane potential are commonly described as depolarization and hyperpolarization. Depolarization refers to an increase in membrane potential toward more positive values, whereas hyperpolarization refers to a decrease toward more negative values (Dayan & Abbott, 2003; Gerstner et al., 2014). Excitatory synaptic inputs generally promote depolarization and increase the probability of spike generation. Inhibitory inputs, in contrast, promote hyperpolarization and reduce the probability that the neuron will reach its firing threshold (Dayan & Abbott, 2003; Garcia-Palencia et al., 2025; Gerstner et al., 2014; Pfeiffer & Pfeil, 2018; Stevens, 1979).

Synaptic communication may be electrical or chemical. In chemical synapses, neurotransmitters released by the presynaptic neuron bind to receptors on the postsynaptic membrane and alter ionic conductance (Gerstner et al., 2014; Izhikevich, 2007; Stevens, 1979). This process generates a postsynaptic potential. An excitatory postsynaptic potential increases the membrane potential, whereas an inhibitory postsynaptic potential decreases it (Ahmed et al., 2014; Gerstner et

al., 2014; Lee, 2025). The combined effect of multiple postsynaptic potentials determines whether the neuron reaches its firing threshold.

When the membrane potential exceeds a critical threshold, voltage-dependent ionic channels are activated and a rapid electrical response occurs (Dayan & Abbott, 2003; Stevens, 1979). Sodium ions initially move into the neuron, causing a sharp depolarization. This is followed by potassium outflow, which drives the membrane potential back toward its resting level (Dayan & Abbott, 2003; Gerstner et al., 2014; Stevens, 1979). This rapid change in membrane potential is called an action potential or spike. During a typical action potential, the membrane potential may rise from a resting value near -70 mV to a positive level and then return toward rest (Dayan & Abbott, 2003; Stevens, 1979).

The amplitude of an action potential is generally not used as the main carrier of information because spikes are commonly treated as all-or-none events. Instead, information may be represented by spike timing, spike frequency, temporal intervals, and firing patterns (Ferster & Spruston, 2005; Gerstner, 1999; Gütig, 2014; Izhikevich, 2007; Lee, 2025). This temporal nature distinguishes spike-based neural communication from the continuous-valued output representation commonly used in conventional artificial neural networks.

If the accumulated input remains below the firing threshold, the neuron does not produce an action potential. In this case, the membrane potential gradually returns toward its resting value (Ahmed et al., 2014; Gerstner et al., 2014). This subthreshold behavior is important because many computational neuron models represent the neuron as an integrator that accumulates incoming signals while simultaneously losing part of the stored membrane potential over time.

After a spike is generated, the neuron enters a refractory period during which the generation of another spike is temporarily suppressed (Dayan & Abbott, 2003; Gerstner et al., 2014). The refractory period allows ionic concentrations and membrane dynamics to recover after the action potential. It also limits the maximum firing frequency of the neuron and influences the temporal structure of spike trains (Ahmed et al., 2014; Dayan & Abbott, 2003; Gerstner et al., 2014).

Biological spike generation can therefore be summarized by several fundamental mechanisms: integration of synaptic inputs, leakage of membrane potential, threshold-based firing, reset after a spike, and temporary refractoriness. Mathematical neuron models represent these mechanisms at different levels of detail. Some models explicitly describe ionic conductances and channel dynamics, whereas others simplify biological processes into a small number of state variables and threshold rules.

3. Spiking Neuron Models

Mathematical neuron models describe the electrical behavior of biological neurons at different levels of abstraction. Some models aim to reproduce ionic

channel dynamics and membrane physiology in detail, whereas others simplify neuronal behavior into a small number of state variables and threshold-based rules. The selection of a neuron model therefore depends on the desired balance between biological plausibility, mathematical complexity, and computational efficiency.

The models considered in this chapter are the Hodgkin–Huxley, Morris–Lecar, Hindmarsh–Rose, Izhikevich, Integrate-and-Fire, and Leaky Integrate-and-Fire models. These models represent a broad spectrum ranging from biophysically detailed formulations to computationally efficient abstractions.

3.1. Hodgkin–Huxley Neuron Model

The Hodgkin–Huxley neuron model was introduced in the early 1950s based on experimental studies conducted on the giant axon of the squid (Hodgkin & Huxley, 1952). The model explains the generation and propagation of action potentials through the movement of sodium and potassium ions across the neuronal membrane. It is regarded as one of the most biologically realistic mathematical descriptions of neuronal electrical activity.

In the Hodgkin–Huxley model, the neuronal membrane is represented by an electrical equivalent circuit composed of a membrane capacitance and several parallel ionic conductance branches. These branches represent sodium, potassium, and leakage currents. The total membrane current is expressed as Eq. (1).

$$I = C_m \frac{dV}{dt} + I_{Na} + I_K + I_L \quad (1)$$

where I is the externally applied membrane current, C_m is the membrane capacitance, and V is the membrane potential. The ionic currents are given by Eq. (2).

$$\begin{aligned} I_{Na} &= \bar{g}_{Na} m^3 h (V - E_{Na}) \\ I_K &= \bar{g}_K n^4 (V - E_K) \\ I_L &= g_L (V - E_L) \end{aligned} \quad (2)$$

Here, $\bar{g}_{Na}$ and $\bar{g}_K$ denote the maximum sodium and potassium conductances, respectively. The term g_L represents the leakage conductance. The parameters E_{Na} , E_K , and E_L are the corresponding equilibrium potentials.

The variables m , h , and n are gating variables that describe the state of voltage-dependent ion channels. The variable m represents sodium channel activation, h represents sodium channel inactivation, and n represents potassium channel activation. Each gating variable takes a value between 0 and 1 and evolves according to the first-order differential equations given in Eq. (3).

$$\begin{aligned}
\frac{dm}{dt} &= \alpha_m(V)(1 - m) - \beta_m(V)m \\
\frac{dh}{dt} &= \alpha_h(V)(1 - h) - \beta_h(V)h \\
\frac{dn}{dt} &= \alpha_n(V)(1 - n) - \beta_n(V)n
\end{aligned} \tag{3}$$

The functions $\alpha_x(V)$ and $\beta_x(V)$ define the voltage-dependent transition rates between open and closed channel states. In general, $\alpha_x(V)$ represents the rate at which a gating variable moves toward the activated state, whereas $\beta_x(V)$ represents the rate at which it returns toward the inactive state (Hodgkin & Huxley, 1952; Johnson & Chartier, 2017).

Using the conventional Hodgkin–Huxley formulation, the voltage-dependent rate functions are given by Eq. (4).

$$\begin{aligned}
\alpha_n(V) &= \frac{0.01(10 - V)}{\exp\left(\frac{10 - V}{10}\right) - 1} \\
\beta_n(V) &= 0.125 \exp\left(-\frac{V}{80}\right) \\
\alpha_m(V) &= \frac{0.1(25 - V)}{\exp\left(\frac{25 - V}{10}\right) - 1} \\
\beta_m(V) &= 4 \exp\left(-\frac{V}{18}\right) \\
\alpha_h(V) &= 0.07 \exp\left(-\frac{V}{20}\right) \\
\beta_h(V) &= \frac{1}{\exp\left(\frac{30 - V}{10}\right) + 1}
\end{aligned} \tag{4}$$

In these expressions, V denotes the membrane potential relative to the resting state in the original formulation. The exact numerical form of the rate equations may change depending on the voltage reference convention used in a study.

The Hodgkin–Huxley model can reproduce major phases of an action potential, including depolarization, repolarization, and recovery. Sodium channel activation is primarily responsible for the rapid increase in membrane potential, whereas potassium channel activation contributes to repolarization. Sodium channel inactivation and leakage currents also play important roles in returning the membrane potential toward its resting level.

The main strength of the Hodgkin–Huxley model is its high biological plausibility. It provides a direct mathematical relationship between membrane potential, ionic conductances, and channel dynamics (Hodgkin & Huxley, 1952;

Johnson & Chartier, 2017). For this reason, it is widely used in computational neuroscience and in studies requiring detailed analysis of neuronal electrophysiology.

Its main disadvantage is mathematical and computational complexity. The model requires the numerical solution of several coupled nonlinear differential equations at each simulation step. Consequently, it has higher computational cost than simplified neuron models such as Hindmarsh–Rose, Izhikevich, and Integrate-and-Fire (Petousakis, Apostolopoulou, & Poirazi, 2023). Therefore, although it is highly suitable for biologically detailed studies, its computational requirements may limit its use in very large neural networks or resource-constrained implementations.

3.2. Morris–Lecar Neuron Model

The Morris–Lecar neuron model was introduced by Catherine Morris and Harold Lecar in 1981 to describe voltage oscillations observed in the giant muscle fiber of barnacles (Morris & Lecar, 1981). The model can be regarded as a reduced conductance-based formulation inspired by the Hodgkin–Huxley model. Instead of describing several independent ionic channel variables, it represents neuronal dynamics using two main state variables: membrane potential and a potassium activation variable.

The Morris–Lecar model considers calcium, potassium, and leakage currents. The membrane potential dynamics are described by Eq. (5).

$$C \frac{dV}{dt} = I - g_{Ca} M_{\infty}(V)(V - V_{Ca}) - g_K N(V - V_K) - g_L(V - V_L) \quad (5)$$

where C is the membrane capacitance, V is the membrane potential, and I is the externally applied current. The parameters g_{Ca} , g_K , and g_L denote the maximum calcium, potassium, and leakage conductances, respectively. The terms V_{Ca} , V_K , and V_L represent the corresponding equilibrium potentials.

The variable $M_{\infty}(V)$, which defines the steady-state activation of calcium channels, is given by Eq. (6).

$$M_{\infty}(V) = \frac{1}{2} \left[1 + \tanh \left(\frac{V - V_1}{V_2} \right) \right] \quad (6)$$

The potassium activation variable N changes according to Eq. (7).

$$\frac{dN}{dt} = \phi \frac{N_{\infty}(V) - N}{\tau_N(V)} \quad (7)$$

where ϕ is a temperature- or time-scale-related parameter. The steady-state potassium activation function is given by Eq. (8).

$$N_{\infty}(V) = \frac{1}{2} \left[1 + \tanh \left(\frac{V - V_3}{V_4} \right) \right] \quad (8)$$

The voltage-dependent time constant is given by Eq. (9).

$$\tau_N(V) = \frac{1}{\cosh \left(\frac{V - V_3}{2V_4} \right)} \quad (9)$$

The parameters V_1 , V_2 , V_3 , and V_4 determine the voltage dependence and slope of the activation functions. In this formulation, calcium activation is assumed to be sufficiently fast to remain close to its steady-state value $M_{\infty}(V)$, whereas potassium activation evolves more slowly through the variable N . This separation of fast and slow dynamics reduces the number of state equations compared with the Hodgkin–Huxley model (Morris & Lecar, 1981).

The Morris–Lecar model can reproduce several important neuronal behaviors, including periodic spiking, excitability, oscillation, and bursting-like responses under suitable parameter and stimulus conditions (Hu, Liu, Liu, Ni, & Li, 2016; Liu, Liu, & Liu, 2014). It can also exhibit different excitability classes. Depending on the selected parameters, the model may demonstrate Class 1, Class 2, or Class 3 firing characteristics (Hu, Liu, Liu, Ni, & Li, 2016; Liu, Liu, & Liu, 2014).

In Class 1 excitability, the firing frequency can increase gradually from very low values as the input current exceeds the firing threshold. This behavior enables the neuron to encode stimulus intensity through firing rate. In Class 2 excitability, oscillatory firing begins at a nonzero frequency, and the firing rate varies within a comparatively narrower range. Class 3 behavior is characterized by a transient spike response rather than sustained repetitive firing under a constant stimulus.

The model may also generate slow spiking, fast spiking, and bursting responses depending on the conductance values and applied current (Hu et al., 2016). Bursting-like activity generally depends on the interaction between fast membrane dynamics and slower recovery dynamics. In some implementations, an external current or additional slow mechanism is needed to produce repeated burst sequences (Hu et al., 2016).

The Morris–Lecar model offers a useful compromise between biological plausibility and computational complexity. Since it preserves conductance-based calcium and potassium dynamics, it provides greater physiological interpretation than simple threshold-based models. At the same time, its two-dimensional structure makes it mathematically simpler than the Hodgkin–Huxley model.

This reduced complexity has made the Morris–Lecar model suitable for bifurcation analysis, nonlinear dynamics studies, and digital implementations (Ghanbarpour, Assaad, & Ghanbarpour, 2024; Hu et al., 2016; Leigh, Heidarpur,

& Mirhassani, 2022; Liu et al., 2014). However, it still requires the numerical solution of nonlinear differential equations and the evaluation of hyperbolic functions. Therefore, its computational cost remains higher than that of simplified models such as Integrate-and-Fire and Leaky Integrate-and-Fire.

3.3. Hindmarsh–Rose Neuron Model

The Hindmarsh–Rose neuron model was introduced by J. L. Hindmarsh and R. M. Rose as a modification of the FitzHugh–Nagumo model (Hindmarsh & Rose, 1984). The model was developed to reproduce important neuronal behaviors such as repetitive spiking, bursting, and transitions between different firing regimes. Compared with conductance-based models such as Hodgkin–Huxley, the Hindmarsh–Rose model uses a smaller set of state variables and provides a mathematically simpler representation of neuronal dynamics.

The standard Hindmarsh–Rose model is defined by the three coupled first-order differential equations given in Eq. (10).

$$\begin{aligned}\frac{dx}{dt} &= y - ax^3 + bx^2 - z + I \\ \frac{dy}{dt} &= c - dx^2 - y \\ \frac{dz}{dt} &= r[s(x - x_R) - z]\end{aligned}\tag{10}$$

In these equations, x represents the membrane potential of the neuron, y is a fast recovery variable associated with rapid ionic processes, and z is a slow adaptation variable. The term I denotes the externally applied current. The parameters a , b , c , and d determine the shape and interaction of the fast subsystem, whereas r , s , and x_R control the slow dynamics and adaptation behavior.

The model contains two distinct time scales. The variables x and y form the fast subsystem and are mainly responsible for individual spike generation. In contrast, the variable z evolves more slowly and modulates the transition between active and inactive firing phases. This interaction between fast and slow dynamics enables the model to reproduce both isolated spikes and burst sequences.

Under a relatively weak stimulus, the membrane potential may remain close to a stable resting state. As the applied current increases, the neuron may begin to generate periodic spikes. For suitable parameter values, the slow variable gradually suppresses the fast oscillatory activity, causing a temporary quiescent phase. When the inhibitory influence of the slow variable decreases, spiking resumes. This repeated alternation between active spiking and silent intervals results in bursting behavior.

The Hindmarsh–Rose model can therefore generate several firing regimes, including slow spiking, fast spiking, periodic bursting, and irregular bursting (Hindmarsh & Rose, 1984; Rajagopal et al., 2019). The exact firing pattern depends on the model parameters, the initial conditions, and the intensity of the applied current. In particular, changes in the slow-variable parameters may strongly affect the duration of bursts, the number of spikes within each burst, and the length of the silent intervals.

Another important feature of the model is its ability to exhibit chaotic dynamics (Rajagopal et al., 2019). In a chaotic firing regime, the intervals between spikes or bursts vary irregularly even though the model is deterministic. Such behavior is useful in nonlinear dynamics studies because it allows the analysis of complex neuronal responses without requiring a highly detailed ionic-channel formulation.

The model is also widely used in neuronal synchronization studies (Dahasert et al., 2012; Malik & Mir, 2020; Rajagopal et al., 2019; Reháč & Lynnyk, 2021). When multiple Hindmarsh–Rose neurons are coupled, their firing activity may become synchronized in terms of spike timing, bursting phase, or oscillation frequency. Synchronization is particularly important in the study of information transmission and coordinated activity in biological neural systems (Malik & Mir, 2020; Reháč & Lynnyk, 2021).

The main advantage of the Hindmarsh–Rose model is its ability to reproduce rich neuronal dynamics using only three state variables. It offers a useful balance between computational simplicity and behavioral diversity. Although it is less biologically detailed than the Hodgkin–Huxley model, it can reproduce spiking, bursting, adaptation, irregular firing, and chaotic behavior with lower mathematical complexity (Rajagopal et al., 2019; Selverston et al., 2000).

Its main limitation is that the model variables are phenomenological rather than direct representations of specific ionic channel conductances. Therefore, the model is more suitable for studying firing dynamics, synchronization, and nonlinear behavior than for detailed physiological interpretation. Nevertheless, its relatively simple structure and wide range of firing patterns make it widely used in computational neuroscience and nonlinear neural-system analysis.

3.4. Izhikevich Neuron Model

The Izhikevich neuron model was introduced by Eugene M. Izhikevich in 2003 (Izhikevich, 2003). The model was developed to combine the biological richness of conductance-based neuron models with the computational simplicity of Integrate-and-Fire-type formulations. It can reproduce a wide range of neuronal firing behaviors while requiring only two state variables and a simple reset mechanism.

The model is defined by the coupled differential equations given in Eq. (11).

$$\begin{aligned}\frac{dv}{dt} &= 0.04v^2 + 5v + 140 - u + I \\ \frac{du}{dt} &= a(bv - u)\end{aligned}\tag{11}$$

The model is complemented by the auxiliary reset condition given in Eq. (12).

$$\text{if } v \geq 30 \text{ mV}, \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases}\tag{12}$$

In these equations, v represents the membrane potential and u is the membrane recovery variable. The input current is denoted by I . The quadratic term in the membrane equation determines the rapid rise of the membrane potential during spike generation, while the recovery variable provides negative feedback and contributes to repolarization and adaptation (Izhikevich, 2003).

The parameters a , b , c , and d determine the firing characteristics of the neuron. The parameter a defines the time scale of the recovery variable. Smaller values of a produce slower recovery dynamics. The parameter b determines the sensitivity of the recovery variable to subthreshold changes in the membrane potential. The parameter c specifies the reset value of the membrane potential after a spike, whereas d determines the amount by which the recovery variable increases following spike generation (Izhikevich, 2003, 2004).

When the membrane potential reaches 30 mV, the model treats this event as a spike and applies the reset rule. The value of 30 mV is not intended to represent the full shape of a biological action potential. Instead, it serves as a numerical threshold for identifying spike occurrence. The reset mechanism approximates the rapid repolarization and recovery processes that follow an action potential.

One of the most important advantages of the Izhikevich model is its ability to reproduce a large variety of firing patterns by changing only a small set of parameters. These patterns include regular spiking, fast spiking, intrinsic bursting, chattering, low-threshold spiking, tonic spiking, tonic bursting, phasic spiking, and phasic bursting (Izhikevich, 2003, 2004). The model can also exhibit spike-frequency adaptation, rebound responses, and delayed firing.

The model may behave as either an integrator or a resonator depending on the selected parameters (Izhikevich, 2004; Vazquez, 2010). An integrator neuron accumulates incoming inputs over time and fires when their combined effect exceeds a threshold. A resonator neuron, in contrast, responds more strongly to inputs with particular temporal or frequency characteristics. This flexibility allows the Izhikevich model to represent different classes of cortical neurons.

The Izhikevich model can also reproduce bursting and chaotic behavior (Izhikevich, 2004; Nobukawa, Nishimura, Yamanishi, & Liu, 2015). Bursting occurs when periods of rapid spike generation alternate with relatively silent intervals. Chaotic responses may appear for appropriate parameter values and

input conditions, producing irregular firing despite deterministic model equations. These characteristics make the model useful in studies of nonlinear neuronal dynamics and neural synchronization.

Compared with Hodgkin–Huxley, Morris–Lecar, and Hindmarsh–Rose models, the Izhikevich model has relatively low computational cost. According to Izhikevich, it can reproduce many cortical firing patterns using approximately 13 floating-point operations per simulation step, whereas more detailed models may require substantially more computations (Izhikevich, 2004). This efficiency makes the model suitable for the simulation of large-scale spiking neural networks.

Several modifications of the original model have been proposed. Wang and Wang introduced short-term depression and delay-related mechanisms into the Izhikevich model (Wang & Wang, 2009). Their study also addressed the possibility that the simulated spike peak may exceed biologically plausible values by limiting the spike apex to approximately 30 mV. Such modifications aim to improve biological plausibility and network-level behavior while preserving the simplicity of the original formulation (Kane, Marcelino, Spirkin, & Chang, 2025; Wang & Wang, 2009).

The main strength of the Izhikevich model is therefore its balance between computational efficiency and behavioral richness. It is considerably simpler than detailed conductance-based models but can reproduce a much wider range of firing patterns than basic Integrate-and-Fire models. Its main limitation is that its variables and parameters do not directly correspond to specific ionic conductances. Consequently, the model is more appropriate for representing neuronal firing behavior than for detailed investigation of membrane physiology.

3.5. Integrate-and-Fire Neuron Model

The Integrate-and-Fire neuron model is one of the earliest and simplest mathematical descriptions of neuronal activity. It was introduced by Louis Lapicque in 1907 and later became an important foundation for simplified spiking neuron models (Abbott, 1999; Lapicque, 1907). The model does not attempt to reproduce detailed ionic channel dynamics. Instead, it represents the neuron as an electrical element that accumulates input over time and generates a spike when its membrane potential reaches a predefined threshold.

Lapicque represented the neuronal membrane using an electrical circuit containing a capacitor and resistive elements (Lapicque, 1907). In this interpretation, the capacitor stores electrical charge and therefore represents the membrane potential. The applied current gradually charges the capacitor until the voltage reaches the firing threshold. Once the threshold is exceeded, the neuron emits a spike and the membrane potential is reset to a resting value.

The general membrane equation is given by Eq. (13).

$$C_m \frac{dV(t)}{dt} = I(t) \quad (13)$$

where C_m is the membrane capacitance, $V(t)$ is the membrane potential, and $I(t)$ is the total input current applied to the neuron. Integrating both sides gives Eq. (14).

$$V(t) = V(t_0) + \frac{1}{C_m} \int_{t_0}^t I(\tau) d\tau \quad (14)$$

This equation indicates that the membrane potential is determined by the accumulated input current. Each incoming stimulus contributes to the membrane potential, and the neuron fires when the accumulated value reaches the threshold.

The firing condition is commonly expressed as Eq. (15).

$$\text{if } V(t) \geq V_{\text{th}}, \begin{cases} \text{generate a spike} \\ V(t) \leftarrow V_{\text{reset}} \end{cases} \quad (15)$$

where V_{th} is the firing threshold and V_{reset} is the membrane reset value. Some formulations also include a refractory period after spike generation, during which the membrane potential is held at the reset value or the neuron is prevented from firing again.

The model describes two fundamental features of neuronal behavior: gradual integration of subthreshold inputs and rapid spike generation after threshold crossing (Burkitt, 2006; Hill, 1936). Inputs that are individually too weak to generate a spike may still produce firing when they arrive sufficiently close together in time. Therefore, the Integrate-and-Fire neuron acts as a temporal accumulator of incoming signals.

For a constant input current I_0 , the membrane potential is given by Eq. (16).

$$V(t) = V(0) + \frac{I_0}{C_m} t \quad (16)$$

If the initial membrane potential is equal to the reset value, the time required to reach the threshold is given by Eq. (17).

$$T_{\text{fire}} = \frac{C_m(V_{\text{th}} - V_{\text{reset}})}{I_0} \quad (17)$$

Accordingly, the firing frequency can be approximated by Eq. (18).

$$f = \frac{1}{T_{\text{fire}} + T_{\text{ref}}} \quad (18)$$

where T_{ref} denotes the refractory period. This relationship shows that stronger input currents result in shorter firing intervals and higher firing frequencies.

The main advantage of the Integrate-and-Fire model is its mathematical simplicity. It requires only the update of a single state variable and a threshold comparison. Therefore, it is suitable for the simulation of large neural networks and for studies in which detailed biological dynamics are not required (Abbott, 1999; Burkitt, 2006).

However, the standard Integrate-and-Fire model has important limitations. Since it assumes perfect accumulation of input, the membrane potential does not decay when the input current is removed. This behavior is less realistic than that of biological neurons, whose membrane potentials gradually return toward the resting level. In addition, the model cannot naturally reproduce complex firing patterns such as bursting, adaptation, rebound firing, or chaotic activity.

Despite these limitations, the Integrate-and-Fire model remains useful because it captures the basic relationship among input integration, threshold crossing, spike generation, and reset. It also forms the basis of the Leaky Integrate-and-Fire model, in which a leakage term is added to represent the gradual decay of membrane potential over time.

3.6. Leaky Integrate-and-Fire Neuron Model

The Leaky Integrate-and-Fire neuron model extends the basic Integrate-and-Fire formulation by introducing a leakage mechanism. In the standard Integrate-and-Fire model, the membrane potential accumulates incoming current without any natural decay. In contrast, the Leaky Integrate-and-Fire model assumes that part of the accumulated membrane potential is continuously lost over time. This behavior provides a more realistic approximation of the passive electrical properties of biological neuronal membranes (Burkitt, 2006; Gerstner et al., 2014).

The model can be represented by a resistor-capacitor circuit. The capacitor models the ability of the membrane to store electrical charge, whereas the resistor represents the leakage of charge through the membrane. The membrane potential dynamics are described by Eq. (19).

$$C_m \frac{dV(t)}{dt} = -\frac{V(t) - V_{\text{rest}}}{R_m} + I(t) \quad (19)$$

where C_m is the membrane capacitance, R_m is the membrane resistance, $V(t)$ is the membrane potential, V_{rest} is the resting potential, and $I(t)$ is the total input current.

The membrane time constant is defined by Eq. (20).

$$\tau_m = R_m C_m \quad (20)$$

Using this definition, the membrane equation can be rewritten as Eq. (21).

$$\tau_m \frac{dV(t)}{dt} = -[V(t) - V_{\text{rest}}] + R_m I(t) \quad (21)$$

The leakage term, $-[V(t) - V_{\text{rest}}]$, causes the membrane potential to decay toward the resting potential when no input is applied. Therefore, unlike the ideal Integrate-and-Fire model, the neuron does not permanently retain previously accumulated input.

For a constant input current I_0 , the membrane potential response is given by Eq. (22).

$$V(t) = V_{\text{rest}} + R_m I_0 + [V(0) - V_{\text{rest}} - R_m I_0] e^{-t/\tau_m} \quad (22)$$

This equation shows that the membrane potential approaches the steady-state value defined by Eq. (23).

$$V_{\infty} = V_{\text{rest}} + R_m I_0 \quad (23)$$

with a rate determined by the membrane time constant τ_m . A small time constant causes the membrane potential to decay rapidly, whereas a large time constant allows the neuron to retain the effect of previous inputs for a longer period.

The spike-generation mechanism is defined by a threshold and reset rule given in Eq. (24).

$$\text{if } V(t) \geq V_{\text{th}}, \begin{cases} \text{generate a spike} \\ V(t) \leftarrow V_{\text{reset}} \end{cases} \quad (24)$$

Here, V_{th} is the firing threshold and V_{reset} is the reset potential. A refractory period may also be included after spike generation. During this interval, the neuron is either prevented from firing or its membrane potential is held at the reset value (Burkitt, 2006; Gerstner et al., 2014).

The LIF neuron integrates excitatory and inhibitory inputs over time. Excitatory inputs increase the membrane potential, whereas inhibitory inputs decrease it. If multiple excitatory inputs arrive within a sufficiently short interval, their combined effect may raise the membrane potential above the threshold. However, if the interval between inputs is too long, the leakage mechanism reduces the contribution of earlier inputs before later inputs arrive. Consequently, the model naturally represents temporal integration.

For numerical simulation, the continuous-time equation may be discretized using the Euler method, as shown in Eq. (25).

$$V[k + 1] = V[k] + \frac{\Delta t}{\tau_m} [- (V[k] - V_{\text{rest}}) + R_m I[k]] \quad (25)$$

where Δt is the simulation time step and k is the discrete-time index. After each update, the membrane potential is compared with the threshold. If the threshold is reached, a spike is generated and the membrane potential is reset.

A simplified discrete formulation is also frequently used, as given in Eq. (26).

$$V[k + 1] = \lambda V[k] + I[k] \quad (26)$$

where λ is the leakage factor satisfying, $0 \leq \lambda < 1$. In this representation, a smaller λ corresponds to stronger leakage, whereas a value close to one corresponds to slower decay. The simplified equation preserves the basic integration, leakage, threshold, and reset mechanisms while reducing the number of arithmetic operations.

The main advantage of the LIF model is its low mathematical complexity. It requires only one principal state variable, a leakage operation, input accumulation, threshold comparison, and reset. Therefore, it is widely used in large-scale spiking neural network simulations and applications in which computational efficiency is more important than detailed biological realism (Burkitt, 2006; Gerstner et al., 2014).

However, the standard LIF model cannot naturally reproduce many complex neuronal behaviors. Firing patterns such as bursting, spike-frequency adaptation, rebound firing, and chaotic activity generally require additional state variables or modified LIF formulations. Moreover, the generated spike is represented as an abstract event rather than as the complete waveform of a biological action potential.

Despite these limitations, the LIF model provides a practical balance between biological interpretation and computational simplicity. Compared with the basic Integrate-and-Fire model, it more realistically represents the decay of membrane potential. Compared with Hodgkin–Huxley, Morris–Lecar, Hindmarsh–Rose, and Izhikevich models, it has substantially lower mathematical complexity. For this reason, it remains one of the most commonly used neuron models in spiking neural network research.

4. Neuronal Firing Patterns

Neurons can respond to external stimulation through different temporal firing patterns. These patterns are determined by the membrane dynamics of the neuron, the intensity and duration of the stimulus, and the parameters of the selected neuron model. Firing patterns are important because they influence how information is represented, transmitted, and processed in a neural system.

Neuronal responses can broadly be classified into single-spike-based and burst-based patterns. In single-spike-based responses, isolated action potentials are generated at particular times. In burst-based responses, multiple spikes occur in rapid succession and are followed by relatively silent intervals. Burst firing

may improve robustness against noise and unreliable synaptic transmission, although it generally requires more neural activity than single-spike communication. Bursting may also support selective communication through resonance-related mechanisms.

4.1. Tonic Spiking

Tonic spiking refers to the regular generation of individual spikes during a sustained stimulus. A representative tonic spiking response is shown in Figure 1. When the applied input exceeds the firing threshold, the neuron begins to emit spikes repeatedly and continues firing until the stimulus is removed.

The firing frequency generally depends on the stimulus intensity and the parameters of the neuron model. Stronger stimulation commonly leads to shorter interspike intervals and therefore higher firing rates. Since tonic spiking continues throughout the stimulus period, it can represent both the presence and approximate intensity of a sustained input.

All neuron models discussed in this chapter can exhibit tonic spiking under suitable parameter and input conditions (Izhikevich, 2004). In simplified models such as Integrate-and-Fire and Leaky Integrate-and-Fire, tonic spiking emerges when the membrane potential repeatedly reaches the threshold, resets, and begins integrating again. In more detailed models, repetitive firing results from interactions among membrane potential, ionic conductances, and recovery variables.

Tonic spiking is particularly suitable for rate-based representation because information can be encoded through the number of spikes generated within a defined time interval.

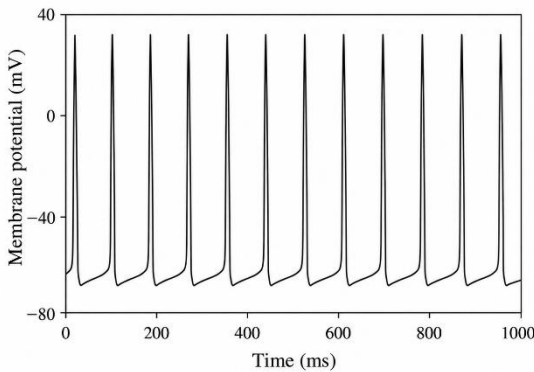


Figure 1. Representative tonic spiking response.

4.2. Tonic Bursting

Tonic bursting is characterized by the regular generation of spike groups during a sustained stimulus, as illustrated in Figure 2. Instead of producing isolated spikes, the neuron repeatedly emits bursts, each containing several closely spaced spikes.

A tonic bursting response generally consists of two alternating phases: an active phase in which several spikes are generated and a silent phase during which the neuron temporarily stops firing. This pattern commonly arises from the interaction of fast spike-generation dynamics with a slower adaptation or recovery variable.

Among the models considered in this chapter, the Hodgkin–Huxley, Hindmarsh–Rose, Morris–Lecar, and Izhikevich models can generate bursting behavior under appropriate conditions (Hu et al., 2016; Izhikevich, 2004). The standard Integrate-and-Fire and Leaky Integrate-and-Fire models do not naturally produce tonic bursting because they contain no additional slow state variable. Bursting in these models requires an extended formulation.

Compared with tonic spiking, tonic bursting may provide greater robustness in noisy systems because several spikes represent the same response event. However, this potential advantage is obtained at the cost of increased neural activity and energy consumption.

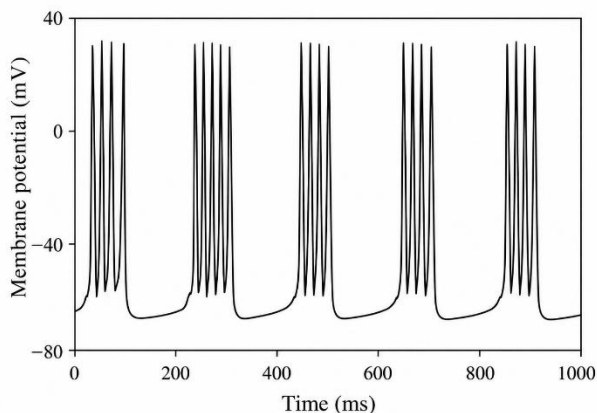


Figure 2. Representative tonic bursting response.

4.3. Phasic Spiking

Phasic spiking refers to the generation of a single spike at the beginning of a stimulus. A representative example is shown in Figure 3. After the initial response, the neuron does not continue firing even if the stimulus remains active.

This behavior is suitable for detecting changes or stimulus onset rather than representing the full duration of an input. A neuron exhibiting phasic spiking acts

as a transition detector because it responds strongly to the beginning of stimulation but adapts to a constant input.

Phasic spiking can be generated by several neuron models, including Hodgkin–Huxley, Morris–Lecar, Hindmarsh–Rose, and Izhikevich models (Izhikevich, 2004). The standard Integrate-and-Fire model does not naturally exhibit phasic spiking because a constant suprathreshold input causes repeated threshold crossings. Additional adaptation or inhibitory mechanisms are required to obtain a purely phasic response.

Since only one spike is generated, phasic spiking can provide an energy-efficient representation of stimulus onset.

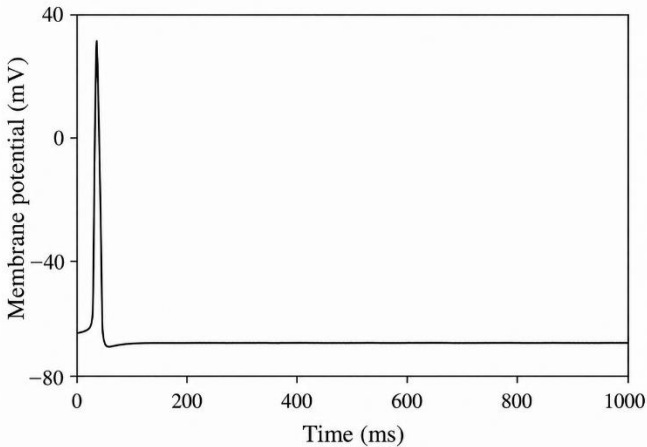


Figure 3. Representative phasic spiking response.

4.4. Phasic Bursting

Phasic bursting is similar to phasic spiking, but the neuron generates a short burst instead of a single spike when the stimulus begins, as illustrated in Figure 4. After this initial burst, firing stops even though the stimulus may continue.

This pattern combines onset detection with the reliability of burst-based transmission. The multiple spikes in the burst may increase the probability that the response is successfully transmitted across unreliable synapses. However, phasic bursting requires more neural activity than phasic spiking.

The Izhikevich model is particularly well known for reproducing phasic bursting through suitable parameter selection (Izhikevich, 2004). More detailed conductance-based models may also generate this behavior under appropriate conditions.

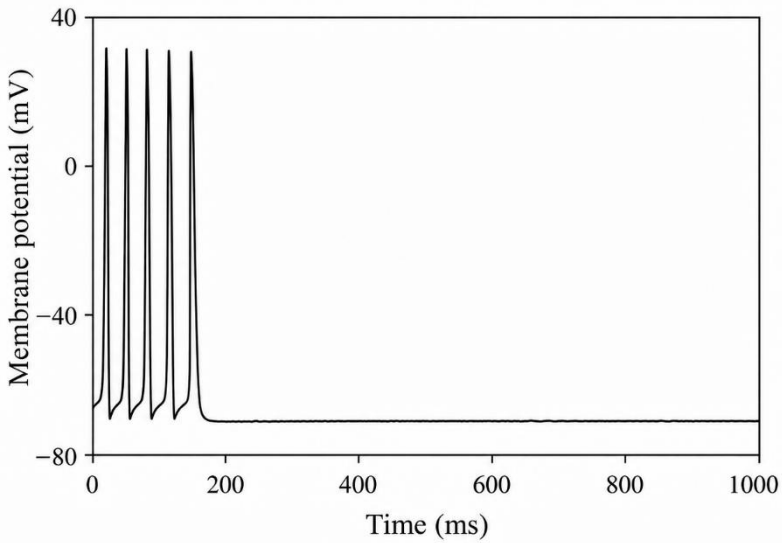


Figure 4. Representative phasic bursting response.

4.5. Spike-Frequency Adaptation

Spike-frequency adaptation occurs when a neuron initially fires rapidly in response to a sustained stimulus, but its firing rate gradually decreases over time. The stimulus remains active, yet the intervals between successive spikes become progressively longer. This behavior is illustrated in Figure 5.

This behavior commonly results from a slow recovery or adaptation process that becomes stronger after each spike. The adaptation variable suppresses subsequent spike generation and reduces the firing frequency.

The Hindmarsh–Rose and Izhikevich models can reproduce spike-frequency adaptation through their slow variables and recovery dynamics (Hindmarsh & Rose, 1984; Izhikevich, 2003, 2004). Conductance-based models may also exhibit adaptation through slowly activated ionic currents. The basic Integrate-and-Fire and Leaky Integrate-and-Fire models do not naturally generate adaptation unless an additional adaptive threshold or recovery variable is introduced.

Spike-frequency adaptation can allow a neuron to respond strongly to a new stimulus while reducing its activity during prolonged stimulation.

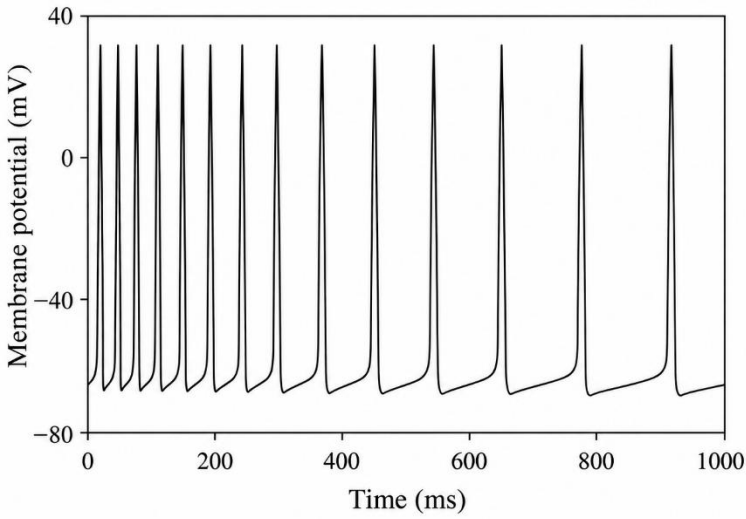


Figure 5. Representative spike-frequency adaptation response.

4.6. Fast Spiking

Fast spiking refers to the generation of spikes at a high and relatively stable frequency. Neurons with this response pattern can maintain short interspike intervals during sustained stimulation, as shown in Figure 6.

The Izhikevich model can represent fast-spiking neurons by selecting appropriate values for its recovery parameters (Izhikevich, 2003, 2004). Morris–Lecar and Hindmarsh–Rose models can also generate high-frequency responses under suitable conditions (Hu et al., 2016).

Fast spiking is generally associated with rapid signal processing and precise temporal control. However, the high firing rate implies greater neural activity compared with slower firing patterns.

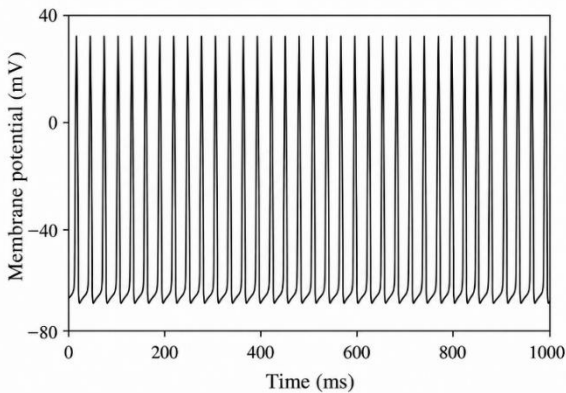


Figure 6. Representative fast-spiking response.

4.7. Delayed Firing

In delayed firing, the neuron does not immediately generate a spike after the stimulus begins. Instead, a measurable delay occurs before the first action potential.

This delay may result from competing membrane processes, slow activation variables, or temporary inhibitory effects. The neuron initially remains below threshold and begins firing only after the internal dynamics evolve sufficiently.

The Izhikevich model can reproduce delayed firing through suitable parameter configurations (Izhikevich, 2004). More detailed models can also exhibit this pattern depending on channel activation and recovery dynamics.

Delayed firing can provide a temporal coding mechanism because information may be represented by the latency between stimulus onset and the first spike. A representative response is shown in Figure 7.

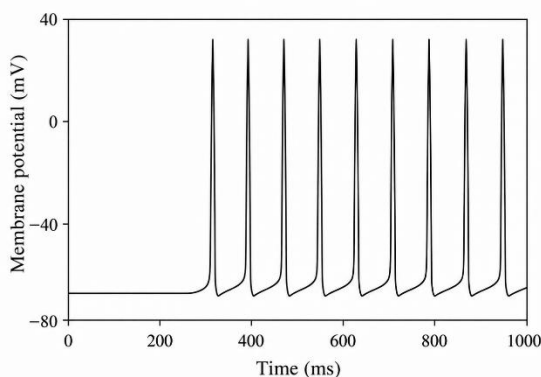


Figure 7. Representative delayed-firing response.

4.8. Rebound Spiking and Rebound Bursting

Rebound responses occur after the removal of an inhibitory stimulus, as shown in Figures 8(a) and 8(b). During inhibition, the membrane potential becomes hyperpolarized. When the inhibitory input ends, the neuron may generate either a single spike or a burst.

A single action potential generated after inhibition is called rebound spiking, whereas a group of spikes is called rebound bursting. These patterns arise because hyperpolarization changes the state of recovery variables or ion channels, producing a transient increase in excitability after inhibition ends.

The Izhikevich model can reproduce both rebound spiking and rebound bursting through appropriate parameter selection (Izhikevich, 2004). Conductance-based models may also generate these responses through voltage-dependent ionic mechanisms.

Rebound firing illustrates that neural information processing depends not only on excitatory input but also on the timing and removal of inhibition.

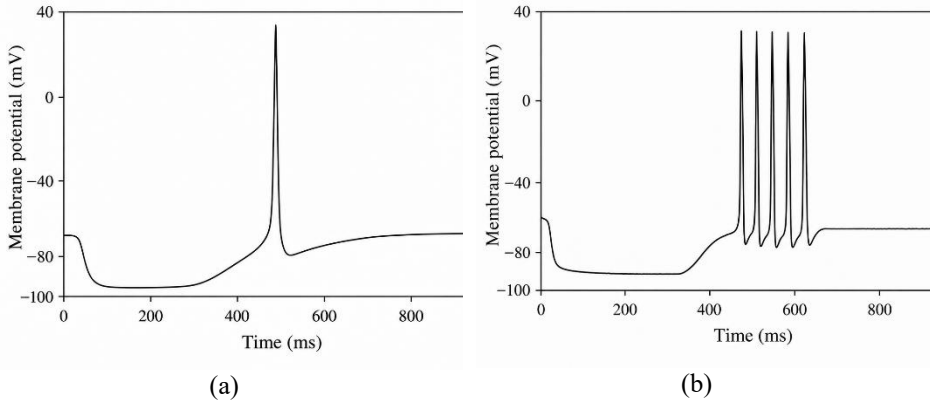


Figure 8. (a) Representative rebound-spiking response (b) Representative rebound-bursting response.

4.9. Class 1 Excitability

Hodgkin classified neuronal excitability into three main classes according to the relationship between applied current and firing response.

Class 1 neurons can generate repetitive spikes at arbitrarily low frequencies when the input current slightly exceeds the firing threshold. As the stimulus intensity increases, the firing frequency increases gradually.

This behavior makes Class 1 neurons suitable for rate coding because the firing frequency can represent the magnitude of the stimulus. All neuron models discussed in this chapter can exhibit Class 1-like behavior under appropriate formulations and parameter selections (Izhikevich, 2004).

The key feature of Class 1 excitability is the continuous transition from rest to repetitive firing.

4.10. Class 2 Excitability

Class 2 neurons begin repetitive firing at a nonzero frequency when the stimulus exceeds the firing threshold. Unlike Class 1 neurons, the firing frequency does not increase smoothly from values close to zero.

The firing rate of a Class 2 neuron generally changes within a comparatively narrow range. Therefore, its response may be less suitable for representing stimulus intensity through firing rate. Instead, the timing and phase of the spikes may carry greater importance (Izhikevich, 2007).

The Hodgkin–Huxley, Morris–Lecar, Hindmarsh–Rose, and Izhikevich models can exhibit Class 2 behavior under suitable conditions (Hu et al., 2016;

Izhikevich, 2004). The standard Integrate-and-Fire model does not naturally reproduce the oscillatory onset mechanism associated with Class 2 excitability.

4.11. Class 3 Excitability

Class 3 neurons generate one or a small number of spikes at stimulus onset but do not produce sustained repetitive firing under constant stimulation.

This behavior resembles a phasic response and is useful for detecting stimulus onset, rapid transitions, or sudden changes. The response emphasizes the occurrence of a new event rather than the duration or magnitude of a sustained input.

Morris–Lecar and other biologically inspired models can exhibit Class 3 behavior under suitable parameter settings (Hu et al., 2016; Liu et al., 2014). Simplified threshold-based models generally require additional adaptation or inhibitory mechanisms to reproduce this response.

4.12. Importance of Firing Pattern Selection

No single firing pattern is optimal for every neural application. Tonic spiking is suitable for sustained input representation and rate coding, whereas phasic spiking is more appropriate for onset detection. Bursting can improve robustness, but it requires more spikes and therefore greater activity. Adaptation emphasizes changes in stimulation, while rebound responses enable the processing of inhibitory events.

The selected neuron model directly affects which firing patterns can be reproduced. Detailed and multi-variable models generally support a wider range of behaviors, whereas simplified models offer lower computational complexity. Therefore, model selection should consider both the required neural dynamics and the intended level of abstraction, which form the basis for the comparative evaluation presented in the following section.

5. Comparative Evaluation of Neuron Models

Neuron models differ considerably in terms of biological plausibility, mathematical complexity, computational requirements, state-variable structure, and ability to reproduce neuronal firing behaviors. Detailed conductance-based models explicitly represent ionic mechanisms, whereas phenomenological and threshold-based models describe neuronal activity using increasingly simplified mathematical formulations. These differences directly affect the level of biological interpretation and the computational effort required for simulation or implementation.

The Hodgkin–Huxley model provides the highest level of biological detail among the models considered in this chapter. It explicitly represents sodium, potassium, and leakage currents together with voltage-dependent gating

variables. This structure enables the model to reproduce the main phases of the action potential and to relate neuronal activity directly to ionic mechanisms. However, the presence of several coupled nonlinear differential equations results in very high mathematical and computational complexity.

The Morris–Lecar model reduces the complexity of conductance-based modeling by representing neuronal activity with two principal state variables. It preserves calcium, potassium, and leakage currents while simplifying the underlying channel dynamics. The model can reproduce oscillatory responses, spiking, bursting-like activity, and different excitability classes. Although it is less complex than the Hodgkin–Huxley model, its nonlinear activation functions still require relatively high computational effort.

The Hindmarsh–Rose model provides a phenomenological representation of neuronal dynamics using three coupled state variables. Its interaction between fast and slow subsystems enables the generation of repetitive spiking, bursting, adaptation, irregular firing, and chaotic behavior. Compared with conductance-based models, its mathematical structure is simpler, but the model variables do not directly correspond to specific ionic channel conductances.

The Izhikevich model combines relatively low computational complexity with a broad range of neuronal firing behaviors. It uses two state variables together with a reset mechanism and can reproduce regular spiking, fast spiking, bursting, adaptation, rebound responses, and delayed firing. Its main limitation is that its parameters primarily control firing behavior rather than representing specific physiological mechanisms.

The Integrate-and-Fire model provides the simplest representation among the models considered in this chapter. It describes input accumulation, threshold crossing, spike generation, and reset using a single principal state variable. Its low mathematical complexity makes it computationally efficient. However, the model does not include membrane leakage and cannot naturally reproduce complex firing behaviors such as bursting, adaptation, rebound firing, or chaotic activity.

The Leaky Integrate-and-Fire model extends the basic Integrate-and-Fire formulation by introducing membrane leakage. This modification allows the membrane potential to gradually return toward its resting value while preserving a simple mathematical structure. The LIF model therefore provides a more realistic representation of temporal integration than the basic IF model while maintaining very low computational complexity. Nevertheless, the standard LIF formulation still cannot naturally reproduce many complex firing patterns without additional mechanisms.

Table 1 summarizes the principal characteristics of the neuron models considered in this chapter.

Table 1. Comparative characteristics of commonly used spiking neuron models

Neuron model	State variables	Biological plausibility	Mathematical complexity	Main firing capabilities	Main limitation
Hodgkin–Huxley	4	Very high	Very high	Action-potential dynamics, repetitive spiking, conductance-based responses	High computational cost
Morris–Lecar	2	High	Medium to high	Oscillation, spiking, bursting-like activity, Class 1–3 excitability	Nonlinear activation functions
Hindmarsh–Rose	3	Medium	Medium	Spiking, bursting, adaptation, irregular and chaotic firing	State variables are phenomenological
Izhikevich	2	Medium	Low	Regular spiking, fast spiking, bursting, adaptation, rebound and delayed firing	Limited physiological interpretation
Integrate-and-Fire	1	Low	Very low	Threshold-based repetitive spiking	No membrane leakage or complex firing dynamics
Leaky Integrate-and-Fire	1	Low to medium	Very low	Temporal integration and repetitive spiking	Cannot naturally generate complex firing patterns

Biological plausibility and computational efficiency generally present a trade-off. Models with explicit ionic mechanisms provide stronger physiological interpretation but require more state variables and nonlinear calculations. In contrast, simplified models reduce computational requirements by representing only the essential mechanisms of integration, threshold crossing, leakage, spike generation, and reset. This simplification improves computational efficiency but reduces the range of neuronal behaviors that can be reproduced.

Another important distinction concerns the interpretation of the model variables. In the Hodgkin–Huxley and Morris–Lecar models, the equations are directly associated with ionic currents and membrane conductances. In contrast, the Hindmarsh–Rose and Izhikevich models use more abstract state or recovery variables to reproduce observed firing dynamics. The Integrate-and-Fire and Leaky Integrate-and-Fire models simplify neuronal behavior further by using membrane potential as the principal state variable.

Overall, the models considered in this chapter represent a spectrum extending from biologically detailed conductance-based formulations to computationally efficient threshold-based abstractions. Hodgkin–Huxley and Morris–Lecar retain strong physiological interpretation, Hindmarsh–Rose and Izhikevich provide richer firing dynamics with reduced complexity, and IF and LIF offer the simplest mathematical structures. These differences form the basis for application-oriented model selection, which is discussed in the following section.

6. Selection of Neuron Models for SNN Applications

The selection of an appropriate neuron model is an important design step in spiking neural network applications because different models offer different balances between biological plausibility, firing-pattern diversity, mathematical complexity, and computational efficiency (Ahmed et al., 2014; Izhikevich, 2004; Yamazaki et al., 2022). Therefore, neuron model selection should be performed according to the main objective of the application rather than by assuming that a single model is suitable for all SNN systems (Izhikevich, 2004).

When the primary objective is to represent neuronal membrane physiology in detail, conductance-based models such as the Hodgkin–Huxley model are more appropriate. This model explicitly represents sodium, potassium, and leakage currents together with voltage-dependent gating variables (Hodgkin & Huxley, 1952; Johnson & Chartier, 2017). As a result, it can provide detailed information about the mechanisms underlying action-potential generation. However, this biological detail is obtained at the cost of relatively high mathematical complexity and computational requirements (Izhikevich, 2004; Petousakis et al., 2023). Therefore, Hodgkin–Huxley-type models are more suitable for studies in which physiological interpretation is more important than computational simplicity.

The Morris–Lecar model may be preferred when a reduced conductance-based description is sufficient. Compared with the Hodgkin–Huxley model, it uses fewer state variables while preserving important calcium, potassium, and leakage dynamics (Morris & Lecar, 1981). This makes the model useful for studies related to excitability, oscillatory behavior, and nonlinear neuronal dynamics (Hu et al., 2016; Liu et al., 2014). However, its nonlinear activation functions and differential equations still result in higher computational requirements than simplified threshold-based neuron models (Hu et al., 2016; Leigh et al., 2022).

For applications focusing on complex firing behavior, bursting, synchronization, or nonlinear dynamics, the Hindmarsh–Rose model provides a useful alternative. Its three-variable structure enables the generation of periodic spiking, bursting, adaptation, irregular firing, and chaotic responses (Hindmarsh & Rose, 1984; Rajagopal et al., 2019). Therefore, it is particularly suitable for studies in which the temporal organization of spikes and bursts is more important than the direct physiological interpretation of individual ion channels. The model has also been widely investigated in synchronization studies (Dahasert et al., 2012; Malik & Mir, 2020; Rehák & Lynnyk, 2021).

The Izhikevich model provides a different compromise between biological behavior and computational efficiency. It can reproduce a wide variety of firing patterns, including regular spiking, fast spiking, bursting, adaptation, rebound responses, and delayed firing, while using only two state variables and a reset mechanism (Izhikevich, 2003, 2004). This combination makes the model suitable

for applications requiring diverse firing behavior without the computational cost of detailed conductance-based models (Izhikevich, 2004). Therefore, it can be considered an effective choice for relatively large spiking neural networks in which neuronal firing diversity remains important.

Integrate-and-Fire and Leaky Integrate-and-Fire models are more suitable when computational simplicity and network scalability are the main design requirements (Burkitt, 2006; Gerstner et al., 2014). The basic Integrate-and-Fire model represents the accumulation of input current, threshold crossing, spike generation, and reset using a single principal state variable (Abbott, 1999; Burkitt, 2006; Lapicque, 1907). Its main limitation is the absence of membrane leakage, which causes accumulated input to remain in the system until a spike occurs (Burkitt, 2006).

The Leaky Integrate-and-Fire model improves this behavior by introducing a leakage mechanism that gradually reduces the membrane potential toward its resting value (Burkitt, 2006; Gerstner et al., 2014). Although the standard LIF model cannot naturally reproduce complex behaviors such as bursting, rebound firing, or spike-frequency adaptation, it provides a useful representation of temporal integration with very low mathematical complexity (Burkitt, 2006; Gerstner et al., 2014). For this reason, LIF neurons are particularly suitable for applications in which large populations of neurons must be simulated or implemented efficiently (Ahmed et al., 2014; Schuman et al., 2017).

The required firing behavior should also be considered during model selection. Applications based mainly on tonic spiking or temporal integration can often be implemented using simple IF or LIF neurons, whereas applications requiring bursting, adaptation, delayed firing, rebound responses, or multiple excitability classes may require models with richer internal dynamics, such as Morris–Lecar, Hindmarsh–Rose, or Izhikevich (Hu et al., 2016; Izhikevich, 2004; Rajagopal et al., 2019). Similarly, studies focusing on detailed membrane behavior may require conductance-based models even when their computational requirements are higher (Hodgkin & Huxley, 1952; Morris & Lecar, 1981; Petousakis et al., 2023).

Consequently, neuron model selection in SNN applications can be viewed as a trade-off among biological interpretation, firing-pattern capability, computational complexity, and scalability (Ahmed et al., 2014; Izhikevich, 2004; Yamazaki et al., 2022). Detailed conductance-based models are appropriate when physiological mechanisms are the main focus. Phenomenological models are suitable when nonlinear firing diversity is required, while simplified threshold-based models are more advantageous when low computational cost and large-scale implementation are the main priorities. Therefore, the selected neuron model should be consistent with the intended application, the required neuronal dynamics, and the available computational resources.

7. Hardware Implementation Considerations

The implementation of spiking neuron models on digital hardware requires a careful balance between computational accuracy, resource utilization, numerical complexity, and real-time processing requirements (Isik, 2023; Pham et al., 2021; Schuman et al., 2017). Although biologically detailed models can reproduce neuronal behavior more accurately, their mathematical structure may increase the cost of hardware implementation. Therefore, the selection of a neuron model for hardware-oriented SNN applications should consider not only biological and computational properties but also the arithmetic operations, memory requirements, and update mechanisms required by the model (Isik, 2023; Pham et al., 2021).

The Hodgkin–Huxley model is one of the most demanding models from a hardware implementation perspective. Its formulation includes several coupled differential equations, voltage-dependent gating variables, and nonlinear functions (Hodgkin & Huxley, 1952; Johnson & Chartier, 2017). These operations generally require a relatively large number of arithmetic units and may involve multiplication, division, and exponential function evaluation. As a result, direct digital implementation may require considerable FPGA logic resources, memory, and computational time (Isik, 2023; Schuman et al., 2017). Approximation methods, lookup tables, or simplified arithmetic structures can reduce this cost, but these techniques may also introduce additional design complexity or numerical approximation errors.

The Morris–Lecar model has a lower state-variable count than the Hodgkin–Huxley model, but it still includes nonlinear activation functions and continuous-time differential equations (Morris & Lecar, 1981; Hu et al., 2016). In particular, the evaluation of hyperbolic functions may increase implementation complexity. Therefore, hardware realizations of the Morris–Lecar model generally benefit from numerical approximation methods or lookup-table-based implementations (Ghanbarpour et al., 2024; Hu et al., 2016; Leigh et al., 2022). This model may still be suitable for hardware systems when nonlinear neuronal dynamics are required and the available resources are sufficient.

The Hindmarsh–Rose model can provide a more practical alternative for representing complex neuronal dynamics in digital hardware. Its equations mainly include polynomial terms and interactions among three state variables (Hindmarsh & Rose, 1984; Rajagopal et al., 2019). Compared with detailed conductance-based models, this structure may simplify numerical implementation. However, the presence of multiple state variables and nonlinear operations still increases the number of arithmetic operations required at each simulation step. Hardware realizations of the Hindmarsh–Rose model have demonstrated its applicability to programmable platforms and synchronization studies (Dahasert et al., 2012). Consequently, the model can be useful for

hardware studies focusing on bursting, synchronization, and nonlinear behavior, but it is generally more resource-demanding than simplified threshold-based models.

The Izhikevich model offers a favorable balance between firing-pattern diversity and implementation complexity. It uses two state variables and a simple reset condition while reproducing several biologically relevant firing behaviors (Izhikevich, 2003, 2004). Its membrane equation includes a quadratic term, which requires additional arithmetic compared with basic IF or LIF formulations. Nevertheless, the overall number of operations remains relatively low compared with conductance-based neuron models (Izhikevich, 2004). Hardware implementations of the Izhikevich model have also been reported using analog and mixed-signal approaches (Rangan, Ghosh, Aparin, & Cauwenberghs, 2010). This characteristic makes the Izhikevich model suitable for hardware systems that require richer firing behavior without the high computational cost of detailed physiological models.

The Integrate-and-Fire model is one of the simplest neuron models for digital implementation. It mainly requires input accumulation, threshold comparison, spike generation, and reset operations (Abbott, 1999; Burkitt, 2006; Lapicque, 1907). Since only one principal state variable is used, memory requirements are low and the neuron update can be implemented using a small number of arithmetic and control elements. However, the absence of leakage limits its ability to represent temporal decay in the membrane potential (Burkitt, 2006).

The Leaky Integrate-and-Fire model is particularly attractive for hardware-oriented SNN implementations because it preserves the simple structure of the Integrate-and-Fire model while adding temporal decay (Burkitt, 2006; Gerstner et al., 2014). In its discrete form, the membrane update can be implemented using addition, multiplication or shift-based leakage, threshold comparison, and reset operations. If suitable fixed-point representations and simplified leakage factors are selected, the multiplication operation may also be reduced to shift-and-add structures. These properties make LIF-type neurons well suited to scalable FPGA and neuromorphic implementations (Isik, 2023; Pham et al., 2021; Schuman et al., 2017).

The hardware-oriented characteristics of the considered neuron models are summarized in Table 2.

Table 2. Hardware-oriented comparison of spiking neuron models

Neuron model	State variables	Nonlinear operations	Relative computational complexity	Fixed-point suitability	Hardware implementation suitability
Hodgkin–Huxley	4	High	Very high	Moderate	Low
Morris–Lecar	2	High	High	Moderate	Low–medium

Hindmarsh–Rose	3	Medium	Medium	Good	Medium
Izhikevich	2	Medium	Low–medium	Good	High
Integrate-and-Fire	1	Very low	Very low	Very good	Very high
Leaky Integrate-and-Fire	1	Low	Very low	Very good	Very high

Numerical representation is another important design consideration in digital SNN implementations (Isik, 2023; Pham et al., 2021). Floating-point arithmetic provides a wide dynamic range and simplifies numerical scaling, but it generally requires more hardware resources than fixed-point arithmetic. Fixed-point representation is therefore often preferred in FPGA and embedded implementations where resource efficiency is important (Isik, 2023). The required word length should be selected carefully because insufficient precision may alter membrane dynamics or spike timing, whereas excessive precision unnecessarily increases logic and memory usage.

Parallelism is one of the major advantages of hardware platforms such as FPGAs for SNN implementation (Isik, 2023; Pham et al., 2021; Schuman et al., 2017). Multiple neuron and synapse updates can be processed simultaneously when sufficient resources are available. Alternatively, a smaller number of arithmetic units can be shared among several neurons through time-multiplexed architectures. The first approach provides higher processing speed but increases resource usage, whereas the second reduces hardware cost at the expense of additional processing cycles (Isik, 2023; Pham et al., 2021).

Consequently, neuron models with lower mathematical complexity are generally more suitable for large-scale and resource-constrained hardware implementations (Isik, 2023; Pham et al., 2021; Schuman et al., 2017). Conductance-based models provide stronger biological interpretation but require more arithmetic operations and hardware resources. Models such as Hindmarsh–Rose and Izhikevich provide intermediate solutions with greater firing-pattern diversity, while IF and LIF models offer the lowest implementation complexity (Burkitt, 2006; Dahasert et al., 2012; Izhikevich, 2004; Rangan et al., 2010). Among these alternatives, the LIF model provides a particularly practical balance between temporal neuronal behavior, computational simplicity, and hardware efficiency, making it a common choice for FPGA-based and neuromorphic SNN systems (Isik, 2023; Pham et al., 2021; Schuman et al., 2017).

8. Conclusion

Spiking neural networks provide a computational framework in which information is represented and transmitted through discrete spikes distributed over time. Unlike conventional artificial neural networks, SNNs directly

incorporate temporal dynamics into neuronal communication. This characteristic makes the selection of an appropriate neuron model an important design decision because the chosen model determines how membrane integration, spike generation, recovery, adaptation, bursting, and other firing behaviors are represented.

This chapter reviewed the biological foundations of spike generation and examined six commonly used neuron models: Hodgkin–Huxley, Morris–Lecar, Hindmarsh–Rose, Izhikevich, Integrate-and-Fire, and Leaky Integrate-and-Fire. These models represent different levels of abstraction. The Hodgkin–Huxley model provides detailed physiological interpretation through ionic currents and voltage-dependent gating variables, whereas the Morris–Lecar model offers a reduced conductance-based formulation. The Hindmarsh–Rose and Izhikevich models provide rich firing dynamics with lower complexity, while the IF and LIF models focus on efficient threshold-based spike generation.

The comparative evaluation of these models shows that biological plausibility and computational simplicity generally form a trade-off. Detailed models provide stronger physiological interpretation and can reproduce a broad range of nonlinear responses, but they require more state variables and numerical calculations. Simplified models are easier to analyze and simulate, although they cannot naturally reproduce many complex firing behaviors without additional mechanisms.

Neuronal firing patterns provide another important criterion for model evaluation. Tonic and phasic responses, bursting, adaptation, delayed firing, rebound activity, and different excitability classes represent distinct forms of neural information processing. Models containing conductance-based mechanisms or slow recovery variables generally support a wider range of these behaviors, whereas basic IF and LIF formulations mainly represent temporal integration and repetitive threshold-based firing.

The selection of a neuron model should therefore be based on the requirements of the intended SNN application. Studies focusing on membrane physiology may require biologically detailed conductance-based models, whereas applications emphasizing nonlinear firing dynamics may benefit from phenomenological models such as Hindmarsh–Rose or Izhikevich. When computational simplicity, scalability, or real-time processing is a primary requirement, threshold-based models such as IF and LIF provide more practical alternatives.

Hardware implementation introduces additional design considerations. Arithmetic complexity, state-variable count, numerical representation, memory usage, and available parallelism can significantly influence the feasibility of implementing a neuron model on digital platforms. Detailed conductance-based formulations generally require more computational resources because of their nonlinear differential equations, whereas simpler models can be realized with

fewer arithmetic operations. Fixed-point representation, approximation techniques, and time-multiplexed processing can further reduce implementation cost when appropriate.

Among the models considered in this chapter, the LIF model offers a particularly simple structure for hardware-oriented SNN systems because it combines temporal integration and membrane leakage with low mathematical complexity. The Izhikevich model provides a useful alternative when greater firing-pattern diversity is required while maintaining moderate computational cost. However, no single neuron model can be considered universally superior, since different applications impose different requirements in terms of biological detail, neuronal dynamics, computational efficiency, and implementation resources.

Overall, understanding the characteristics, advantages, and limitations of different neuron models is essential for the effective design of spiking neural networks. Model selection should not be based solely on biological realism or mathematical simplicity, but rather on an appropriate balance among the required firing behavior, application objective, computational constraints, and implementation platform.

References

- Abbott, L. F. (1999). Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Research Bulletin*, 50(5–6), 303–304.
- Ahmed, F. Y., Yusob, B., & Hamed, H. N. A. (2014). Computing with spiking neuron networks: A review. *International Journal of Advances in Soft Computing & Its Applications*, 6(1).
- Alqahtani, F. M., Adwan, S., Ahmad, M. Y., Karman, S. B., Majed, R., & Majed, L. A. (2025, May). Optimized ICA-based approach for accurate retinal vessel segmentation. In *2025 IEEE 15th Symposium on Computer Applications & Industrial Electronics (ISCAIE)* (pp. 242–248). IEEE.
- Bezugam, S. S., Shaban, A., & Suri, M. (2023, May). Neuromorphic recurrent spiking neural networks for EMG gesture classification and low power implementation on Loihi. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1–5). IEEE.
- Burkitt, A. N. (2006). A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input. *Biological Cybernetics*, 95(1), 1–19.
- Dahasert, N., Öztürk, İ., & Kiliç, R. (2012). Experimental realizations of the HR neuron model with programmable hardware and synchronization applications. *Nonlinear Dynamics*, 70(4), 2343–2358.
- Dayan, P., & Abbott, L. F. (2003). Theoretical neuroscience: Computational and mathematical modeling of neural systems. *Journal of Cognitive Neuroscience*, 15(1), 154–155.
- Fang, H., Shrestha, A., Zhao, Z., Li, Y., & Qiu, Q. (2019). An event-driven neuromorphic system with biologically plausible temporal dynamics. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (pp. 1–8). IEEE.
- Ferster, D., & Spruston, N. (2005). Cracking the neuronal code. *Science*, 270(5237), 756–757.
- Garcia-Palencia, O., Fernandez, J., Shim, V., Kasabov, N. K., Wang, A., & Alzheimer's Disease Neuroimaging Initiative. (2025). Spiking neural networks for multimodal neuroimaging: A comprehensive review of current trends and the NeuCube brain-inspired architecture. *Bioengineering*, 12(6), 628.
- Gerstner, W. (1999). Spiking neurons. In W. Maass & C. M. Bishop (Eds.), *Pulsed neural networks* (pp. 3–54).
- Gerstner, W., Kistler, W. M., Naud, R., & Paninski, L. (2014). *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press.

- Ghanbarpour, G., Assaad, M., & Ghanbarpour, M. (2024). New model for Wilson and Morris–Lecar neuron models: Validation and digital implementation on FPGA. *IEEE Access*, 12, 154751–154759.
- Gütig, R. (2014). To spike, or when to spike? *Current Opinion in Neurobiology*, 25, 134–139.
- Hill, A. V. (1936). Excitation and accommodation in nerve. *Proceedings of the Royal Society of London. Series B-Biological Sciences*, 119(814), 305–355.
- Hindmarsh, J. L., & Rose, R. M. (1984). A model of neuronal bursting using three coupled first order differential equations. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 221(1222), 87–102.
- Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4), 500–544.
- Hu, X., Liu, C., Liu, L., Ni, J., & Li, S. (2016). An electronic implementation for Morris–Lecar neuron model. *Nonlinear Dynamics*, 84(4), 2317–2332.
- Isik, M. (2023). A survey of spiking neural network accelerator on FPGA. *arXiv preprint arXiv:2307.03910*.
- Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6), 1569–1572.
- Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks*, 15(5), 1063–1070.
- Izhikevich, E. M. (2007). *Dynamical systems in neuroscience*. MIT Press.
- Johnson, M. G., & Chartier, S. (2017). Spike neural models (Part I): The Hodgkin–Huxley model. *The Quantitative Methods for Psychology*, 13(2), 105–119.
- Kane, A., Marcelino, F., Spirkin, A., & Chang, Y. (2025, September). An improved Izhikevich neuron model with time delay and adaptive parameters for high-performance spiking neural networks. In *2025 IEEE High Performance Extreme Computing Conference (HPEC)* (pp. 1–6). IEEE.
- Kim, C. G., Hwang, Y. J., & Kamyod, C. (2022). A study of profanity effect in sentiment analysis on natural language processing using ANN. *Journal of Web Engineering*, 21(3), 751–766.
- Lapicque, L. (1907). Recherches quantitatives sur l’excitation électrique des nerfs traitée comme une polarisation. *Journal de Physiologie et de Pathologie Générale*, 9, 620–635.
- Lee, J. Y. (2025). Neural transmission and auditory encoding in afferent spiral ganglion neurons. *The Journal of the Acoustical Society of Korea*, 44(3), 240–248.

- Leigh, A. J., Heidarpur, M., & Mirhassani, M. (2022). A high-accuracy digital implementation of the Morris–Lecar neuron with variable physiological parameters. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(10), 4138–4142.
- Lin, J., Chen, H., Li, S., Liu, Y., Li, X., & Yu, B. (2019). Accurate prediction of potential druggable proteins based on genetic algorithm and Bagging-SVM ensemble classifier. *Artificial Intelligence in Medicine*, 98, 35–47.
- Liu, C., Liu, X., & Liu, S. (2014). Bifurcation analysis of a Morris–Lecar neuron model. *Biological Cybernetics*, 108(1), 75–84.
- Liu, T., Liu, Z., Lin, F., Jin, Y., Quan, G., & Wen, W. (2017, November). MT-Spike: A multilayer time-based spiking neuromorphic architecture with temporal error backpropagation. In *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (pp. 450–457). IEEE.
- Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9), 1659–1671.
- Malik, S. A., & Mir, A. H. (2020). Synchronization of Hindmarsh Rose neurons. *Neural Networks*, 123, 372–380.
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115–133.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26.
- Morris, C., & Lecar, H. (1981). Voltage oscillations in the barnacle giant muscle fiber. *Biophysical Journal*, 35(1), 193–213.
- Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (pp. 807–814).
- Narayan, S. (1997). The generalized sigmoid activation function: Competitive supervised learning. *Information Sciences*, 99(1–2), 69–82.
- Nobukawa, S., Nishimura, H., Yamanishi, T., & Liu, J. Q. (2015). Analysis of chaotic resonance in Izhikevich neuron model. *PLoS ONE*, 10(9), e0138919.
- Perez, C., & Gutierrez, E. (2025, May). Analysis of calibration requirements in image recognition tasks with time-encoded ANNs. In *2025 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1–5). IEEE.
- Petousakis, K.-E., Apostolopoulou, A. A., & Poirazi, P. (2023). The impact of Hodgkin–Huxley models on dendritic research. *The Journal of Physiology*, 601(15), 3091–3102.

- Pfeiffer, M., & Pfeil, T. (2018). Deep learning with spiking neurons: Opportunities and challenges. *Frontiers in Neuroscience*, 12, 409662.
- Pham, Q. T., Nguyen, T. Q., Hoang, P. C., Dang, Q. H., Nguyen, D. M., & Nguyen, H. H. (2021, October). A review of SNN implementation on FPGA. In *2021 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)* (pp. 1–6). IEEE.
- Qiumei, Z., Dan, T., & Fenghua, W. (2019). Improved convolutional neural network based on fast exponentially linear unit activation function. *IEEE Access*, 7, 151359–151367.
- Rahman, N. A., & Yusoff, N. (2025). Modulated spike-time dependent plasticity (STDP)-based learning for spiking neural network (SNN): A review. *Neurocomputing*, 618, 129170.
- Rajagopal, K., Khalaf, A. J. M., Parastesh, F., Moroz, I., Karthikeyan, A., & Jafari, S. (2019). Dynamical behavior and network analysis of an extended Hindmarsh–Rose neuron model. *Nonlinear Dynamics*, 98(1), 477–487.
- Rangan, V., Ghosh, A., Aparin, V., & Cauwenberghs, G. (2010, August). A subthreshold aVLSI implementation of the Izhikevich simple neuron model. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology* (pp. 4164–4167). IEEE.
- Rashid, T. (2016). *Make your own neural network*. CreateSpace Independent Publishing Platform.
- Rasamoelina, A. D., Adjailia, F., & Sinčák, P. (2020, January). A review of activation function for artificial neural network. In *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMII)* (pp. 281–286). IEEE.
- Rehák, B., & Lynnyk, V. (2021). Synchronization of a network composed of stochastic Hindmarsh–Rose neurons. *Mathematics*, 9(20), 2625.
- Sabbella, H., Mukherjee, A., Kandappu, T., Dey, S., Pal, A., Misra, A., & Ma, D. (2025). The promise of spiking neural networks for ubiquitous computing: A survey and new perspectives. *arXiv preprint arXiv:2506.01737*.
- Schuman, C. D., Potok, T. E., Patton, R. M., Birdwell, J. D., Dean, M. E., Rose, G. S., & Plank, J. S. (2017). A survey of neuromorphic computing and neural networks in hardware. *arXiv preprint arXiv:1705.06963*.
- Selverston, A. I., Rabinovich, M. I., Abarbanel, H. D. I., Elson, R., Szűcs, A., Pinto, R. D., Huerta, R., & Varona, P. (2000). Reliable circuits from irregular neurons: A dynamical approach to understanding central pattern generators. *Journal of Physiology-Paris*, 94(5–6), 357–374. [https://doi.org/10.1016/S0928-4257\(00\)01101-3](https://doi.org/10.1016/S0928-4257(00)01101-3).

- Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, 6(12), 310–316.
- Stevens, C. F. (1979). The neuron. *Scientific American*, 241(3), 54–65.
- Tian, Y. (2020). Artificial intelligence image recognition method based on convolutional neural network algorithm. *IEEE Access*, 8, 125731–125744.
- Vazquez, R. (2010). Izhikevich neuron model and its application in pattern recognition. *Australian Journal of Intelligent Information Processing Systems*, 11(1), 35–40.
- Wang, H., & Wang, H. (2009, December). Improvement of Izhikevich's neuronal and neural network model. In *2009 International Conference on Information Engineering and Computer Science* (pp. 1–4). IEEE.
- Yamazaki, K., Vo-Ho, V. K., Bulsara, D., & Le, N. (2022). Spiking neural networks and their applications: A review. *Brain Sciences*, 12(7), 863.
- Zhang, M., Gu, C., He, Y., Lin, S., & Zheng, X. (2025, May). An applied study of image recognition using convolutional neural networks. In *2025 IEEE 5th International Conference on Electronic Technology, Communication and Information (ICETCI)* (pp. 589–596). IEEE.

CHAPTER 9

Android Malware Detection via Generative AI

Bekir Berkcan¹

¹ Res. Asst. Dr., Department of Electrical and Electronics Engineering, Faculty of Engineering, Artvin Çoruh University, Artvin, Türkiye. ORCID: 0000-0002-5695-1513

1. Introduction

Rapid technological advances have also influenced the development of artificial intelligence (AI), leading to the emergence of models capable of generating new and meaningful data samples based on structures learned from existing data. Generative AI (GAI) refers to computational techniques that can produce seemingly new and meaningful content, such as text, images, or audio, from training data (Feuerriegel, Hartmann, Janiesch, & Zschech, 2024). Whereas traditional discriminative models focus on tasks such as classification or prediction, generative models learn the probability distribution of data and can produce similar yet original content. These models also offer capabilities relevant to cybersecurity, a persistent concern across technological eras. Through their ability to model relationships in high-dimensional data and learn representations, GAI models can support not only original content generation but also numerous cybersecurity tasks, including anomaly detection, identity spoofing and fraud detection, the generation of malware variants, cyber threat intelligence generation, and secure code generation.

As in nearly all other areas of cybersecurity, GAI is also advancing Android malware detection. Android malware refers to any malicious software that targets devices running the Android operating system with the aim of gaining unauthorized access, exfiltrating data, or disrupting the system (Karakaya & Ulu, 2024). Generative language models such as GPT can support the interpretation of relationships within Android APK files and generate descriptions of application behavior. Related transformer encoders, such as BERT and CodeBERT, learn contextual representations of text or code; their use for feature extraction does not itself constitute generative modeling. Together, these approaches can help interpret information from manifest files, permissions, API calls, and code snippets.

Android applications contain multilayered and heterogeneous data structures, including manifest files, DEX code, permission structures, API calls, and user interface components. Manually analyzing these structures is difficult. Consequently, Android malware detection systems generally rely on static, dynamic, or hybrid analysis techniques. Static analysis examines application code, whereas dynamic analysis observes behaviors that emerge during execution. However, these methods may be insufficient against advanced evasion techniques such as code obfuscation, anti-emulation, and polymorphic malware. In particular, limitations in static and dynamic analysis can create serious gaps in mobile security systems when dealing with obfuscated applications and sophisticated malware families (Rashid et al., 2025). Automated models capable of learning and generalizing are therefore needed, especially for large-scale malware analysis.

Integrating generative AI into Android malware detection can improve detection accuracy and enable systems to adapt to new threats. For example, synthetically generated malware variants can provide a valuable source of training data for classification models. Nevertheless, the use of GAI in this specific context has not yet sufficiently matured in the literature. Many machine-learning-based Android malware detection models, particularly those relying on static data, have limitations that reduce their reliability in real-world applications (Y. Liu, Tantithamthavorn, Li, & Liu, 2022). This situation indicates inconsistencies and a lack of standards in the architectures and evaluation metrics used.

This chapter systematically reviews and comparatively analyzes studies that use GAI models for Android malware detection. It also presents current trends and directions for future research.

2. Role of Generative AI in Malware Detection

Malware targeting the Android operating system is becoming increasingly diverse and complex. Traditional signature-based methods and purely discriminative machine learning models may struggle to detect previously unseen malware types (Guerra-Manzanares, Luckner, & Bahsi, 2022). Generative AI models offer innovative solutions to this problem, providing substantial advantages in adaptive learning, behavioral analysis, and data enrichment. Their contributions can be considered under these three themes.

Adaptation to New Malware

The first contribution of generative AI models is their ability to support adaptation to new malware. The mobile malware threat landscape continues to evolve. For example, Kaspersky reported detecting 1,315,405 unique malicious or potentially unwanted mobile installation packages in 2023 (Kivva, 2024). This volume illustrates the scale of the detection challenge, including previously unseen malware variants. Conventional machine-learning-based detection systems often struggle to identify new variants that are absent from their training data.

Adaptive learning refers to a model's ability to adjust to new malware behaviors and patterns over time. Generative AI models support this adaptation through their ability to generate new samples from learned distributions. In particular, generative models such as generative adversarial networks (GANs) and variational autoencoders (VAEs) offer flexibility in recognizing emerging threats because they can capture data distributions and derive samples from them, rather than merely classify existing data.

Behavioral Analysis: Proactive Detection of Abnormal Behavior

Another contribution of generative AI is behavioral analysis, particularly the proactive detection of abnormal behavior. Dynamic analysis, or behavioral analysis, is vital for malware detection. Whereas static analysis examines fixed application characteristics such as code and permissions, dynamic analysis observes runtime behavior, including system calls, memory and CPU usage, and network traffic, to identify abnormal activity (Thangaveloo, Wong, Chiew, & Abdullah, 2020). Generative AI can serve as a powerful tool for anomaly detection in dynamic analysis. By learning and modeling the statistical distribution of normal user or system behavior, generative models can proactively identify activities that deviate from this profile. This makes it possible to raise an alert when a malicious application exhibits unusual behavior, potentially before it causes harm.

Related deep learning approaches to anomaly detection use reconstruction or sequence prediction to learn normal system behavior (Milosevic & Huang, 2019). Reconstruction errors or deviations from predicted behavior can indicate potentially anomalous activity. Milosevic and Huang (2019) combined two separate models in an ensemble: a conventional autoencoder that reconstructs measurements and an LSTM that predicts the next measurement using the preceding 20 seconds of observations. Their framework analyzed Android resource usage, including CPU, memory, and battery measurements, to identify deviations from normal behavior (Milosevic & Huang, 2019). This example illustrates complementary reconstruction and prediction approaches to behavioral anomaly detection; a conventional autoencoder or an LSTM predictor is not, by itself, evidence of generative modeling. Such approaches can complement generative methods when identifying unusual runtime behavior.

Data Enrichment

The final contribution is data enrichment. In AI-based malware detection, model performance depends heavily on the quantity and diversity of training data. However, obtaining balanced and comprehensive datasets in real-world settings is difficult: some malware families have very few available samples, whereas others have thousands. This imbalance and scarcity make accurate classification particularly challenging for rare malware families (Ban, Yi, & Cho, 2023). Generative AI addresses this problem by enriching datasets through synthetic data generation and improving models' generalization capacity. Generative models can create new, realistic samples based on existing malware samples. These synthetic samples increase diversity during training and help models maintain strong performance on previously unseen examples.

Academic studies have demonstrated concrete benefits from augmenting training datasets with synthetic data. In particular, GANs and conditional variational autoencoders (CVAEs) have been used to augment Android malware

data. For example, a study by Stalin and Mekoya (2024) reported performance improvements ranging from 1.5% to 7% in Android malware detection using Wasserstein-GAN-based data augmentation (Stalin & Mekoya, 2024). In that study, a GAN used malicious application features as input to generate realistic new samples. A deep learning model trained with these synthetic data achieved an F1 score of 0.975 in some test scenarios, representing a substantial performance gain compared with training on the original data.

These findings demonstrate the important role of generative AI in addressing insufficient training data. Synthetic data both expand the range of malware samples from which classification algorithms can learn and enrich the distribution of dataset content, reducing the risk of overfitting.

3. Generative Models and Related Representation Models

Generative Adversarial Networks (GANs)

A generative adversarial network (GAN) is a machine learning model in which two neural network components are trained with opposing objectives. These components are called the generator and the discriminator. The generator attempts to produce synthetic data resembling real data, while the discriminator attempts to distinguish real data from the synthetic data produced by the generator. During training, the generator learns to produce data realistic enough to deceive the discriminator, while the discriminator learns to identify generated samples. Through this adversarial training process, the generator is trained to produce new samples that closely resemble the original data distribution (Goodfellow et al., 2014).

GANs have been used to generate new variants of existing malware that can evade detection systems. In particular, attackers can use a GAN's generator to create adversarial malware samples designed to bypass detection. Studies have demonstrated the use of GANs to evade deep-learning-based classifiers trained on mobile malware, showing that GAN-generated malware variants can pass through detection systems unnoticed (Kc, Sapkota, & Adhikari, 2024).

GANs have also been used defensively to generate synthetic data that strengthen the training of malware detection models. GAN-based data augmentation has been explored particularly for imbalanced datasets, such as those containing relatively few malware samples (Kc et al., 2024).

GAN-based methods have also been explored for Android malware intrusion detection and classification. Shirazi and Shaikh proposed an approach that combines GAN-generated synthetic samples with zero-shot learning to support the classification of previously unseen malware variants. This approach illustrates the potential of generative models to augment training data for intrusion detection when labeled examples are unavailable for every malware class (Shirazi & Shaikh, 2023).

Variational Autoencoders (VAEs)

A variational autoencoder (VAE) is a probabilistic generative model that learns to encode and reconstruct data through latent variables. Unlike conventional autoencoders that map inputs to deterministic latent representations, VAEs use an encoder to approximate a probability distribution over latent variables for each input. Training combines a reconstruction objective with regularization toward a chosen prior distribution, commonly a multidimensional standard normal distribution (Kingma & Welling, 2013).

A VAE consists of two components. The encoder transforms the input into a low-dimensional latent representation characterized by a parameterized probability distribution, with means and variances. The decoder maps vectors sampled from the latent space back into the original data domain to reconstruct the input. In this way, a VAE learns a compact representation that captures the underlying characteristics of the data while also enabling the generation of new samples by sampling from the latent prior (Kingma & Welling, 2013). In Android malware detection, VAE-based models have been explored for modeling benign behavior and identifying anomalous application executions (Vella & Colombo, 2021).

The latent space of VAEs can be used to capture malware characteristics. Instead of working directly with raw data from an Android application's code or behavioral traces, such as API call sequences and system logs, a VAE can summarize these high-dimensional data as a low-dimensional latent vector. Because this vector retains important features of the original data, it can provide more meaningful input to classification algorithms. Studies have shown improved detection performance when VAE-derived latent features are combined with traditional machine learning models (Ajayi, Barakat, & McGarry, 2025).

Another important use of VAEs is the detection of abnormal behavior. A VAE trained on benign application data can assign anomaly scores to executions that deviate from learned patterns. Reconstruction-based scores can help identify potentially suspicious behavior. For example, Vella and Colombo's SpotCheck framework evaluates VAE-based anomaly detection using Android system call traces and process memory dumps. It flags suspicious applications for further malware analysis, supporting the identification of unusual behavior that signature-based methods may miss (Vella & Colombo, 2021).

Like GANs, VAEs can also be used for data augmentation. Conditional VAE models can synthesize new samples belonging to a specific Android malware family.

Transformer-Based Generative and Representation Models

Transformer-based models consist of encoder and/or decoder blocks containing multi-head attention layers. When trained on large datasets, they learn

contextual relationships and can extract rich representation vectors from text, code, or visual data. Generative language models produce new content, whereas encoder-based approaches such as BERT and the transformer component of Chimera are discussed here as related representation methods used in malware classifiers, rather than as examples of generative modeling.

The transformer architecture originated in natural language processing and revolutionized sequential data processing through its self-attention mechanism. Transformer models offer a powerful approach to interpreting the static characteristics of Android applications. For example, permissions and textual descriptions in an application's `AndroidManifest` file, or its decompiled source code, can be represented as sequences suitable for transformer models. A model such as BERT processes these sequences to produce contextual vectors for each token, such as an API name, a permission name, or a word in the code. These contextual representations can capture linguistic cues that distinguish malicious applications from benign ones (Alshomrani, Albeshri, Alturki, Alallah, & Alsulami, 2024).

Transformer encoders can also model dynamic application behavior represented as sequences of runtime events. For example, de Oliveira and Sassi's Chimera framework uses a transformer encoder to learn representations from Android system call sequences. These dynamic representations are combined with static permission and intent features and representations learned from DEX images in a multimodal malware classifier. Chimera demonstrates the use of transformer-based sequence modeling within hybrid Android malware detection (de Oliveira & Sassi, 2020).

Generative language models can support Android malware detection through code reasoning and the generation of behavioral explanations. For example, Qian et al. proposed LAMD, a context-driven framework that extracts security-relevant code and program structures through static analysis, then uses tier-wise LLM reasoning to progress from function-level behavior summaries to API intent analysis and application-level maliciousness judgments. The framework generates human-readable explanations alongside its predictions and applies factual consistency verification to reduce hallucinations in intermediate summaries (Qian, Zheng, He, Yang, & Cavallaro, 2025).

4. Generative AI Techniques for Android Malware Detection

4.1. Static Analysis with Generative Models

Static analysis is a widely used approach for Android malware detection, as it enables the examination of Android application packages (APKs) without executing the applications. This process involves extracting relevant information from the `AndroidManifest.xml` file, Dalvik Executable (DEX) code, and application resources. Common static features include requested permissions,

API calls referenced in the code, opcode sequences, application components, and embedded URLs. These features provide valuable information for distinguishing malicious applications from benign ones. For example, Arp et al.'s Drebin framework combines features extracted from the manifest and disassembled DEX code, including permissions, application components, API calls, and network addresses, in a joint feature space for malware detection (Arp, Spreitzenbarth, Hübner, Gascon, & Rieck, 2014).

Generative AI models can enhance static analysis by generating synthetic feature representations and addressing limitations associated with insufficient or imbalanced training data. In particular, GANs can learn the underlying distribution of extracted malware features and generate additional samples to improve the training of classification models. For example, S. Li et al. (2024) proposed GMADV, an Android malware classification and adversarial training framework that transforms APK-derived file data into RGB Markov images. The framework uses a GMM-GAN-based augmentation mechanism to increase sample diversity and a VGG13 network for feature extraction and classification. The generated image representations are incorporated into adversarial training to strengthen malware classification (S. Li et al., 2024).

Large language models (LLMs) have also introduced new opportunities for analyzing static features. Rather than relying exclusively on conventional feature vectors, LLM-assisted approaches can generate semantic descriptions of application functionality and potentially suspicious behavior. Zhao et al. (2025) proposed AppPoet, which extracts permissions, API information, URLs, and other static features from APK files. The framework uses an LLM to generate functional descriptions and behavioral summaries from multiple feature views. These texts are converted into embedding vectors and used to train a multilayer perceptron (MLP) classifier. The authors reported an accuracy of 97.15% and an F1 score of 97.21% in their experimental evaluation (Zhao, Wu, & Meng, 2025).

Despite these advantages, static analysis remains vulnerable to code obfuscation, packing, and dynamically loaded malicious components (Arp et al., 2014; S. Li et al., 2024). Furthermore, generated feature or image representations do not necessarily correspond to valid executable applications, while LLM-generated descriptions may contain inaccuracies (Zhao et al., 2025). Therefore, the effectiveness of generative approaches depends on the quality of extracted features, the reliability of generated representations, and rigorous evaluation using independent malware samples.

4.2. Dynamic Analysis with Generative Models

Dynamic analysis examines Android applications during execution, typically within an instrumented emulator or on a real device. Runtime observations can include system calls, invoked APIs, network activity, process memory, and CPU or battery usage (Milosevic & Huang, 2019; Vella & Colombo, 2021). These

observations provide evidence of application behavior, complementing the structural information available through static analysis. The quality of collected traces depends on the execution environment and the inputs used to exercise the application. Alzaylaee et al.'s DL-Droid study demonstrated the importance of stateful input generation and code coverage when collecting behavioral features on real Android devices (Alzaylaee, Yerima, & Sezer, 2020).

Generative models can support dynamic analysis by learning the distribution of benign runtime behavior and identifying observations that fit this distribution poorly. VAEs are particularly relevant because they combine probabilistic latent representations with reconstruction-based modeling. Vella and Colombo's SpotCheck framework applies VAE-based anomaly detection to system call traces and process memory dumps collected during Android application execution. Its experimental evaluation used 2,000 benign and 1,000 malicious applications, with malware samples reserved for testing. Suspicious executions are identified through reconstruction-probability-based anomaly scores and submitted for further malware analysis. This approach treats deviations as evidence requiring investigation, rather than assigning a definitive malware label solely from an anomaly score (Vella & Colombo, 2021).

GAN-based architectures have also been investigated for learning representations of dynamic application behavior. Jan et al. (2025) proposed a deep convolutional generative adversarial network (DCGAN) approach trained on behavioral logs from more than 10,000 Android applications, emphasizing resource-consumption patterns associated with application activity. The learned features were used to distinguish benign from malicious applications. The authors reported 96.5% accuracy, 97.0% precision, 96.1% recall, and a 96.5% F1 score in their evaluation (Jan et al., 2025). This study provides an Android-specific example of using an adversarially trained architecture to extract discriminative information from behavioral logs.

Despite these opportunities, dynamic analysis can miss malicious actions that are not triggered within the observation period. Delayed execution, limited interaction coverage, and differences between emulated and real-device environments can restrict the behaviors available to a model (Alzaylaee et al., 2020; Milosevic & Huang, 2019). Anomalous behavior may also arise from legitimate application activity, making threshold selection and false-positive control essential. Evaluation should compare generative methods with appropriate non-generative baselines: in SpotCheck, KPCA achieved higher F1 scores than the evaluated VAE configurations (Vella & Colombo, 2021). Accordingly, reported performance should be interpreted in relation to the collected runtime features, execution coverage, and test conditions, with deployment also accounting for monitoring overhead and response latency.

4.3. Hybrid Approaches

Hybrid approaches combine static evidence from an APK with observations collected during application execution. Features such as permissions, API references, and call graphs can be analyzed alongside executed API calls, system calls, and network activity. These complementary views can be integrated through feature fusion or coordinated detection stages (Nasser, Hasan, & Humaidi, 2024; Zhang, Su, Liu, & Yang, 2025). An illustrative use case is correlating code that accesses sensitive information with observed outbound network traffic. Such correlation can strengthen an investigation, although identifying data exfiltration requires evidence connecting the accessed information to the transmitted content; the presence of networking code or traffic alone is insufficient.

Recent Android studies illustrate different ways of organizing hybrid analysis. Zhang et al.'s MPDroid represents static information using function call graphs and dynamic information using API call graphs. Graph convolutional networks, modality alignment, and multimodal fusion combine these views during pre-training, while downstream detection can operate using static features alone (Zhang et al., 2025). Nasser et al.'s DL-AMDet instead employs a staged architecture: a CNN-BiLSTM model analyzes permissions and API calls, and applications classified as non-malicious undergo additional system-call-based anomaly detection using deep autoencoders (Nasser et al., 2024). These frameworks provide useful architectural examples for integrating generative components. However, multimodal fusion and conventional autoencoder reconstruction do not, by themselves, establish that a framework uses a generative model.

Generative models could extend these architectures by augmenting training data while preserving relationships between static and dynamic features. As a relevant data-generation example, T. Li et al. (2025) introduced BS-GAN, which combines Borderline-SMOTE with a GAN to address class imbalance, followed by feature processing and an attention-based stacking ensemble (T. Li et al., 2025). Their evaluation used Drebin features, so the study supports generative augmentation of static representations rather than demonstrating joint static–dynamic generation. Extending this idea to hybrid analysis would require paired observations from the same application and validation that synthetic combinations remain consistent. For instance, generated network activity should be compatible with the application's communication capabilities and observed execution context.

Hybrid systems also face practical limitations. Dynamic traces may omit untriggered behavior, while collecting and aligning multiple feature types increases analysis cost (Zhang et al., 2025). Staged analysis, as in DL-AMDet, offers one way to avoid additional execution when static analysis already

identifies malware (Nasser et al., 2024). For generative extensions, evaluation should separately measure the contribution of feature fusion and synthetic augmentation, using static-only, dynamic-only, and hybrid baselines. Real applications should remain separate from generated training samples in the test set, and comparisons should report false positives, execution coverage, and analysis time alongside detection performance.

5. Discussion, Conclusion, and Future Directions

Advantages of Generative AI

Generative AI can enhance Android malware detection by improving data diversity, representation learning, and adaptation to previously unseen malware variants. Unlike conventional signature-based methods, which primarily rely on previously observed malicious patterns, generative models can learn the underlying distribution of malware features and produce synthetic samples that expand the training space. For example, Stalin and Mekoya used Wasserstein GANs to augment Android malware datasets and reported performance improvements of approximately 1.5–7%, with an F1-score of up to 0.975. Similarly, the GMADV framework employed GAN-based variant amplification and adversarial training to increase intra-family diversity, achieving F1-scores of at least 95% across three Android malware datasets. These findings suggest that generative augmentation can improve generalization when available malware samples are limited or imbalanced (S. Li et al., 2024; Stalin & Mekoya, 2024).

However, generative AI should not be described as universally more accurate than conventional machine-learning methods. Its main advantage is better expressed as greater flexibility and potential adaptability: synthetic-data generation, latent-space learning, and semantic modeling can complement discriminative classifiers and help them cope with evolving malware distributions. Recent LLM-assisted approaches reinforce this point. AppPoet, for instance, combines Android permissions, APIs, URLs, and other static features with LLM-generated behavioral descriptions and reported 97.15% accuracy and a 97.21% F1-score (Zhao et al., 2025).

Anomaly Detection and Real-Time Threat Hunting

Generative and reconstruction-based models are particularly relevant to behavioral anomaly detection because they can model patterns associated with normal application activity and identify observations that deviate from the learned distribution. Dynamic Android analysis can exploit runtime information such as system calls, network activity, resource usage, and inter-process behavior, providing signals that may not be visible through static APK inspection alone. The MaDroid study, for example, emphasizes runtime system-call sequences as useful representations for mobile anomaly detection (Duan, Liu, Cai, Sun, & Chen, 2024).

A practical example is DL-AMDet, which combines static detection with a deep-autoencoder-based dynamic anomaly detector. Applications are executed in an emulated environment, their system calls are collected during runtime, and reconstruction error is used to identify anomalous behavior. Importantly, the framework performs dynamic analysis selectively after the initial static stage, reducing unnecessary analysis and therefore limiting detection time and computational complexity. The combined architecture reported an accuracy of 99.935% under its experimental setting (Nasser et al., 2024).

Such mechanisms can also support continuous threat hunting by flagging behavioral deviations for further investigation rather than depending exclusively on known malware signatures. Nevertheless, “real-time” detection requires more than high classification accuracy: behavioral events must be collected, processed, and evaluated with sufficiently low latency. TDBAMLA, for example, uses LSTM networks and attention mechanisms to analyze temporal patterns in system calls and network interactions, illustrating the relevance of dynamic monitoring to timely threat identification (Misalkar & Harshavardhanan, 2025). End-to-end detection delay should nevertheless be evaluated separately from accuracy and recall under the intended deployment conditions.

Challenges and Limitations

Despite these advantages, deploying generative AI for mobile malware detection introduces significant computational constraints. Model size, memory usage, inference latency, and energy consumption become especially important when analysis is expected to run directly on smartphones. These limitations are most pronounced for transformer and LLM-based components. MobileLLM, for example, was explicitly designed around sub-billion-parameter architectures because conventional LLMs are difficult to deploy efficiently on resource-constrained mobile hardware (Z. Liu et al., 2024). A 2025 empirical study of mobile LLM deployment similarly found that only relatively small models could be executed successfully on high-end mobile devices in its experiments, while meaningful on-device inference could require more than 30 seconds in some configurations (Yan & Ding, 2025). Measurements of small language models on edge hardware further demonstrate an explicit trade-off among model accuracy, inference latency, and energy consumption (Islam et al., 2025). These measurements concern general LLM applications and edge workloads; malware-detection pipelines require task-specific latency and energy evaluation.

For Android malware detection, this means that computationally expensive generative models may be better suited to cloud-assisted, edge-assisted, or staged architectures, while lightweight models perform initial on-device screening. Hybrid designs already illustrate this principle by avoiding expensive dynamic analysis when an earlier detection stage provides sufficient evidence (Nasser et al., 2024).

False positives are another important concern. Anomaly detectors identify deviations from learned normality, but unusual behavior is not necessarily

malicious; consequently, reconstruction thresholds and behavioral baselines must be carefully calibrated. This is particularly relevant in Android environments, where legitimate applications may exhibit highly diverse runtime behavior. Hybrid systems attempt to mitigate this problem by combining complementary evidence rather than relying on a single anomalous feature source (Asmitha, Vinod, Rafidha Rehiman, Raveendran, & Conti, 2025; Nasser et al., 2024).

Finally, generative models introduce a dual-use security problem. GANs can strengthen detectors through augmentation and adversarial training, but similar techniques can also generate evasive malware variants. Recent Android research has demonstrated GAN-based manipulation of opcode distributions to produce adversarial malware examples intended to evade learned detectors. Independent problem-space attacks such as EvadeDroid have likewise achieved high evasion rates against several Android malware classifiers, demonstrating that adversarial robustness remains a practical concern rather than merely a theoretical one (Ahmad et al., 2025; Bostani & Moonsamy, 2024).

Insights from the Literature

The reviewed literature indicates that generative models can contribute to both static and dynamic analysis. Their usefulness depends on the feature representation and the role they perform within the detection pipeline. GAN-based methods have also produced promising results from static or APK-derived representations: WGAN-based augmentation has been applied to image representations of Android malware features, while GMADV generates variants from APK-derived file representations for classification and adversarial training. Likewise, CVAE-based work has demonstrated that generative augmentation can improve Android malware family classification under class imbalance (Ban et al., 2023; S. Li et al., 2024; Stalin & Mekoya, 2024).

Conversely, reconstruction-based models are particularly natural for dynamic anomaly detection because runtime system calls and behavioral sequences can be modeled according to their deviation from normal patterns. DL-AMDet provides a reconstruction-based example in which a deep autoencoder operates on dynamically collected system-call information (Nasser et al., 2024), while SpotCheck specifically evaluates VAE-based dynamic anomaly detection (Vella & Colombo, 2021). Thus, a more defensible interpretation is that GANs and VAEs can contribute to both static and dynamic detection, but their function differs according to the available representation: they are commonly used for data augmentation and latent representation learning in static settings, while reconstruction-based approaches are especially useful for behavioral anomaly detection.

Recent hybrid studies illustrate the potential benefits of combining complementary static and dynamic feature sources. A 2025 hybrid multimodal Android malware framework fused models trained on static and dynamic

attributes and reported an F1-score of 99.97% on CICMaldroid2020 (Asmitha et al., 2025). DL-AMDet similarly reported 99.935% accuracy when its static and dynamic components were combined, while other recent multimodal work explicitly argues that each individual modality has inherent limitations (Nasser et al., 2024; Zhang et al., 2025). These results make hybrid approaches particularly promising, although numerical results across studies should not be compared directly without considering differences in datasets, train/test protocols, class distributions, feature representations, and malware collection periods.

Conclusion and Future Directions

This chapter examined the role of generative AI in Android malware detection across static, dynamic, and hybrid analysis. The reviewed approaches illustrate several complementary contributions: GANs can augment training data, VAEs can model application behavior for anomaly detection, and LLMs can generate semantic descriptions and explanations from security-relevant application features. Hybrid architectures provide opportunities to connect information extracted from application code with observations collected during execution. Together, these approaches offer useful directions for improving detection systems, although their effectiveness depends on the data, feature representations, and evaluation conditions. Conclusions about their advantages should therefore remain specific to the experimental settings in which they were evaluated.

Future research should prioritize realistic and reproducible evaluation. Models should be tested on independently collected applications, including samples from later periods and malware families excluded from training, to assess their ability to handle evolving threats. Synthetic augmentation should be evaluated separately from other architectural changes, with generated samples confined to training and checked for diversity, label consistency, and correspondence to plausible application behavior. Comparisons with conventional methods should report false-positive rates and computational costs alongside detection metrics. For hybrid systems, evaluation should also examine execution coverage and consistency between static features and runtime observations.

Further work should develop efficient architectures suitable for practical Android deployment, balancing detection capability with memory usage, energy consumption, and response latency. Integrating generative models into staged or multimodal analysis pipelines is a promising research direction, particularly when additional runtime analysis can be targeted to applications requiring further investigation. LLM-assisted systems should connect their explanations to verifiable code or execution evidence, while model updates should be evaluated against changing application behavior and adversarial manipulation. Progress in these areas could help translate experimental results into reliable and maintainable mobile security tools.

References

- Ahmad, N., Saleem Rana, A., Jalil Hadi, H., Bashir Hussain, F., Chakrabarti, P., Alshara, M. A., & Chakrabarti, T. (2025). GEAAD: Generating evasive adversarial attacks against Android malware defense. *Scientific Reports*, 15(1), 11867. <https://doi.org/10.1038/s41598-025-96392-x>
- Ajayi, B., Barakat, B., & McGarry, K. (2025). *Leveraging VAE-derived latent spaces for enhanced malware detection with machine learning classifiers*. arXiv preprint arXiv:2503.20803. Retrieved from <https://arxiv.org/abs/2503.20803>
- Alshomrani, M., Albeshri, A., Alturki, B., Alallah, F. S., & Alsulami, A. A. (2024). Survey of transformer-based malicious software detection systems. *Electronics*, 13(23), 4677. <https://www.mdpi.com/2079-9292/13/23/4677>
- Alzaylaee, M. K., Yerima, S. Y., & Sezer, S. (2020). DL-Droid: Deep learning based Android malware detection using real devices. *Computers & Security*, 89, 101663. <https://doi.org/10.1016/j.cose.2019.101663>
- Arp, D., Spreitzenbarth, M., Hübner, M., Gascon, H., & Rieck, K. (2014). Drebin: Effective and explainable detection of Android malware in your pocket. In *Proceedings of the Network and Distributed System Security Symposium (NDSS 2014)*. <https://www.ndss-symposium.org/ndss2014/ndss-2014-programme/drebin-effective-and-explainable-detection-android-malware-your-pocket/>
- Asmitha, K. A., Vinod, P., Rafidha Rehiman, K. A., Raveendran, N., & Conti, M. (2025). Android malware defense through a hybrid multi-modal approach. *Journal of Network and Computer Applications*, 233, 104035. <https://doi.org/10.1016/j.jnca.2024.104035>
- Ban, Y., Yi, J. H., & Cho, H. (2023). Augmenting Android malware using conditional variational autoencoder for the malware family classification. *Computer Systems Science and Engineering*, 46(2), 2215–2230. <https://www.techscience.com/csse/v46n2/51665>
- Bostani, H., & Moonsamy, V. (2024). EvadeDroid: A practical evasion attack on machine learning for black-box Android malware detection. *Computers & Security*, 139, 103676. <https://doi.org/10.1016/j.cose.2023.103676>
- de Oliveira, A. S., & Sassi, R. J. (2020). *Chimera: An Android malware detection method based on multimodal deep learning and hybrid analysis*. TechRxiv preprint. <https://doi.org/10.36227/techrxiv.13359767.v1>
- Duan, G., Liu, H., Cai, M., Sun, J., & Chen, H. (2024). MaDroid: A maliciousness-aware multifeatured dataset for detecting Android malware. *Computers & Security*, 144, 103969. <https://doi.org/10.1016/j.cose.2024.103969>

- Feuerriegel, S., Hartmann, J., Janiesch, C., & Zschech, P. (2024). Generative AI. *Business & Information Systems Engineering*, 66(1), 111–126. <https://link.springer.com/article/10.1007/s12599-023-00834-7>
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27. <https://proceedings.neurips.cc/paper/2014/hash/f033ed80deb0234979a61f95710dbe25-Abstract.html>
- Guerra-Manzanares, A., Luckner, M., & Bahsi, H. (2022). Concept drift and cross-device behavior: Challenges and implications for effective Android malware detection. *Computers & Security*, 120, 102757. <https://www.sciencedirect.com/science/article/pii/S0167404822001523>
- Islam, M. R., Deng, B., Dhar, N., Nguyen, T. N., He, S., Shi, Y., & Suo, K. (2025). Characterizing and understanding energy footprint and efficiency of small language model on edges. In *Proceedings of the 2025 IEEE 22nd International Conference on Mobile Ad-Hoc and Smart Systems (MASS)* (pp. 363–371). IEEE. <https://doi.org/10.1109/MASS66014.2025.00059>
- Jan, S., Ali, G., Ul Rehman, S., Syed, T. A., Jilani, A. K., Khan, M. Y., & Ali, J. (2025). Advancing Android malware detection: Innovative approach in behavioral log analysis. In R. K. Hamdan (Ed.), *Sustainable data management* (Studies in Big Data, Vol. 170, pp. 611–621). Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-031-83915-3_50
- Karakaya, A., & Ulu, A. (2024). Dynamic malware detection approach based on API calls: Machine learning and ensemble learning models. *International Journal of Information Security Science*, 13(4), 1–20. <https://dergipark.org.tr/en/download/article-file/4044979>
- Kc, B., Sapkota, S., & Adhikari, A. (2024). Generative adversarial networks in anomaly detection and malware detection: A comprehensive survey. *Advances in Artificial Intelligence Research*, 4(1), 18–35. https://www.researchgate.net/publication/383575896_Generative_Adversarial_Networks_in_Anomaly_Detection_and_Malware_Detection_A_Comprehensive_Survey
- Kingma, D. P., & Welling, M. (2013). *Auto-encoding variational Bayes*. arXiv preprint arXiv:1312.6114. Retrieved from <https://arxiv.org/abs/1312.6114>
- Kivva, A. (2024, February 26). *The mobile malware threat landscape in 2023*. Securelist, Kaspersky. Retrieved from <https://securelist.com/mobile-malware-report-2023/111964/>
- Li, S., Tang, Z., Li, H., Zhang, J., Wang, H., & Wang, J. (2024). GMADV: An Android malware variant generation and classification adversarial training

- framework. *Journal of Information Security and Applications*, 84, 103800. <https://doi.org/10.1016/j.jisa.2024.103800>
- Li, T., Yao, Q., Li, J., Li, Q., Wang, R., Jia, C., & Xiao, Y. (2025). Malware detection model based on stacking ensemble technique. *Applied Soft Computing*, 180, 113338. <https://doi.org/10.1016/j.asoc.2025.113338>
- Liu, Y., Tantithamthavorn, C., Li, L., & Liu, Y. (2022). Explainable AI for Android malware detection: Towards understanding why the models perform so well? In *Proceedings of the 2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE)* (pp. 169–180). IEEE. <https://ieeexplore.ieee.org/abstract/document/9978981>
- Liu, Z., Zhao, C., Iandola, F., Lai, C., Tian, Y., Fedorov, I., . . . Chandra, V. (2024). MobileLLM: Optimizing sub-billion parameter language models for on-device use cases. In *Proceedings of the 41st International Conference on Machine Learning* (Proceedings of Machine Learning Research, Vol. 235, pp. 32431–32454). PMLR. <https://proceedings.mlr.press/v235/liu24ce.html>
- Milosevic, N., & Huang, J. (2019). *Deep learning guided Android malware and anomaly detection*. arXiv preprint arXiv:1910.10660. Retrieved from <https://arxiv.org/abs/1910.10660>
- Misalkar, H. D., & Harshavardhanan, P. (2025). TDBAMLA: Temporal and dynamic behavior analysis in Android malware using LSTM and attention mechanisms. *Computer Standards & Interfaces*, 92, 103920. <https://doi.org/10.1016/j.csi.2024.103920>
- Nasser, A. R., Hasan, A. M., & Humaidi, A. J. (2024). DL-AMDet: Deep learning-based malware detector for Android. *Intelligent Systems with Applications*, 21, 200318. <https://doi.org/10.1016/j.iswa.2023.200318>
- Qian, X., Zheng, X., He, Y., Yang, S., & Cavallaro, L. (2025). *LAMD: Context-driven Android malware detection and classification with LLMs*. arXiv preprint arXiv:2502.13055. Retrieved from <https://arxiv.org/abs/2502.13055>
- Rashid, M. U., Qureshi, S., Abid, A., Alqahtany, S. S., Alqazzaz, A., ul Hassan, M., . . . Shaikh, A. (2025). Hybrid Android malware detection and classification using deep neural networks. *International Journal of Computational Intelligence Systems*, 18(1), 52. <https://link.springer.com/article/10.1007/s44196-025-00783-x>
- Shirazi, S. A. R., & Shaikh, M. (2023). A novel approach to Android malware intrusion detection using zero-shot learning GANs. *Sir Syed University Research Journal of Engineering & Technology*, 13(2). <https://doi.org/10.33317/ssurj.584>
- Stalin, K., & Mekoya, M. B. (2024). *Improving Android malware detection through data augmentation using Wasserstein generative adversarial networks*.

- Thangaveloo, R., Wong, W. J., Chiew, K. L., & Abdullah, J. (2020). DATDroid: Dynamic analysis technique in Android malware detection. *International Journal on Advanced Science, Engineering and Information Technology*, 10(2), 536–541. https://www.researchgate.net/publication/340919935_DATDroid_Dynamic_Analysis_Technique_in_Android_Malware_Detection
- Vella, M., & Colombo, C. (2021). SpotCheck: On-device anomaly detection for Android. In *Proceedings of the 13th International Conference on Security of Information and Networks (SIN 2020)*. ACM. <https://doi.org/10.1145/3433174.3433591>
- Yan, X., & Ding, Y. (2025). Are we there yet? A measurement study of efficiency for LLM applications on mobile devices. In *Proceedings of the 2nd International Workshop on Foundation Models for Cyber-Physical Systems & Internet of Things* (pp. 19–24). ACM. <https://doi.org/10.1145/3722565.3727192>
- Zhang, S., Su, H., Liu, H., & Yang, W. (2025). MPDroid: A multimodal pre-training Android malware detection method with static and dynamic features. *Computers & Security*, 150, 104262. <https://doi.org/10.1016/j.cose.2024.104262>
- Zhao, W., Wu, J., & Meng, Z. (2025). AppPoet: Large language model based Android malware detection via multi-view prompt engineering. *Expert Systems with Applications*, 262, 125546. <https://doi.org/10.1016/j.eswa.2024.125546>

CHAPTER 10

Deep Learning-Based Damage Detection and Classification in Historic Masonry Structures Implications for Post- Earthquake Assessment

Bekir Berkcan¹

¹ Res. Asst. Dr., Department of Electrical and Electronics Engineering, Faculty of Engineering, Artvin Çoruh University, Artvin, Türkiye. ORCID: 0000-0002-5695-1513

1. Introduction

Historic masonry structures constitute an essential part of architectural and cultural heritage, reflecting the construction techniques and traditions of past civilizations. These structures, including religious monuments, towers, and fortifications, are often characterized by heterogeneous materials, complex geometries, and structural systems that differ from modern construction practices. Their limited tensile resistance and inadequate connections between structural components make them particularly vulnerable to seismic actions, which may cause cracking, masonry separation, out-of-plane wall failures, and partial or complete collapse. Identifying potential failure mechanisms is therefore essential for assessing seismic vulnerability and supporting appropriate conservation strategies (D'Ayala & Speranza, 2003). Furthermore, ICOMOS (2003) emphasizes the importance of reliable structural assessment methods that respect the architectural and historical significance of cultural heritage structures.

Post-earthquake damage assessment traditionally relies on visual inspections conducted by structural engineers and conservation specialists. Although these inspections provide valuable information, they can be time-consuming, labor-intensive, and difficult to perform in damaged or inaccessible areas. In addition, damage interpretation may vary depending on the inspector's experience. These limitations have encouraged the development of automated image-based methods for identifying and documenting damage in masonry structures (Wang et al., 2018; Dais et al., 2021).

Recent advances in artificial intelligence, particularly deep learning (DL), have introduced promising approaches for automated damage detection and classification. Convolutional neural networks (CNNs) can extract relevant visual features directly from images, reducing the need for manually designed features. Previous studies have demonstrated their potential for identifying cracks and different forms of surface deterioration in masonry structures (Wang et al., 2018; Dais et al., 2021; Hacıfendioğlu et al., 2023). Beyond classification, object detection and semantic segmentation enable damage localization and pixel-level identification, respectively. These methods provide increasingly detailed information about visible defects, although their effectiveness depends on image quality, training data, and surface characteristics (ElBehairy et al., 2026). Importantly, not all defects investigated in these studies were necessarily caused by earthquakes.

Despite these developments, identifying visible damage does not necessarily establish the seismic damage state or structural safety of a building. Recent investigations have therefore explored the integration of deep learning with engineering-oriented assessment. Hacıfendioğlu et al. (2024) employed image segmentation to estimate damage levels in earthquake-affected masonry structures, while Pereira et al. (2024) investigated the prediction of residual drift

capacity using crack patterns. Fraternali et al. (2026) further examined the integration of CNN-based damage recognition with mechanical models for preliminary post-earthquake assessment. These studies indicate the potential of deep learning to support structural evaluation beyond conventional visual damage identification.

However, several challenges remain, particularly for historic masonry structures. Limited heritage-specific datasets, variations in materials and construction techniques, complex surface textures, and pre-existing deterioration may restrict model generalization. Furthermore, models trained on conventional masonry or laboratory specimens may not perform reliably on historic structures under real post-earthquake conditions. Distinguishing visible surface deterioration from earthquake-induced structural damage also requires careful interpretation and integration with engineering knowledge.

In this context, this chapter presents a critical review of deep learning-based damage detection and classification methods and their implications for post-earthquake assessment of historic masonry structures. Particular attention is given to CNN-based classification, transfer learning, object detection, and semantic segmentation. Existing studies are compared in terms of their methodologies, datasets, damage categories, performance measures, and limitations. The review distinguishes studies directly involving historic masonry from those providing transferable insights from conventional masonry and discusses the relationship between visual damage identification and structural assessment. Finally, current challenges and future research directions are examined to support the development of more reliable and practically applicable approaches for preserving historic masonry heritage.

2. Historic Masonry Structures and Earthquake-Induced Damage

2.1. Characteristics of Historic Masonry Structures

Historic masonry structures represent an important component of architectural heritage and include churches, mosques, minarets, towers, castles, and other monumental buildings constructed using traditional masonry techniques. These structures are commonly composed of natural stone, fired clay bricks, adobe blocks, and mortar, with construction materials and techniques varying according to geographical location, historical period, and local architectural traditions. Unlike modern engineered buildings, historic masonry structures often exhibit irregular geometries, heterogeneous materials, and complex structural configurations that significantly influence their mechanical behavior (Roca et al., 2010).

The mechanical performance of masonry structures is primarily governed by the interaction between masonry units and mortar joints. Masonry generally exhibits relatively high compressive strength but limited tensile capacity, making

it vulnerable to cracking and separation under lateral loading. Furthermore, variations in material properties, wall thickness, bonding patterns, and construction quality can produce nonuniform stress distributions throughout the structure. The absence of effective connections between perpendicular walls or between walls and horizontal diaphragms may further increase susceptibility to seismic damage. D'Ayala and Speranza (2003) demonstrated that the seismic vulnerability of historic masonry buildings is strongly associated with their structural configuration and the potential activation of different collapse mechanisms.

Historic masonry buildings may also include arches, vaults, domes, and slender structural elements, whose geometric characteristics can influence their seismic response. Large openings, irregular wall arrangements, and heavy architectural components may contribute to localized stress concentrations and complex deformation patterns. Moreover, structural modifications, previous restoration interventions, material deterioration, and accumulated damage can alter the original mechanical behavior of these structures. As emphasized by Roca et al. (2010), understanding the actual structural condition of historic masonry buildings requires consideration of their construction history, material characteristics, and structural organization. These characteristics are particularly important when interpreting visible damage patterns and developing reliable post-earthquake assessment methods.

2.2. Typical Seismic Damage Mechanisms

Earthquake-induced damage in historic masonry structures is primarily associated with their limited tensile resistance, nonlinear material behavior, structural irregularities, and connection deficiencies. The distribution and severity of damage depend on several factors, including ground-motion characteristics, masonry quality, structural geometry, boundary conditions, and the interaction between structural elements. Consequently, similar masonry structures may exhibit different damage patterns even when subjected to comparable seismic actions (D'Ayala & Speranza, 2003; Lagomarsino, 2012).

Seismic damage mechanisms in masonry walls are commonly classified into in-plane and out-of-plane failure mechanisms. In-plane mechanisms occur when seismic forces act approximately parallel to the wall surface and may produce diagonal shear cracks, horizontal sliding along mortar joints, flexural cracking, or localized crushing. Diagonal cracking is frequently associated with shear-dominated behavior, whereas horizontal cracks and rocking mechanisms may develop under flexural actions. Out-of-plane mechanisms occur when seismic forces act perpendicular to the wall surface, potentially causing wall bending, separation from adjacent structural components, overturning, or partial collapse. The vulnerability to these mechanisms is strongly influenced by the effectiveness of wall-to-wall and wall-to-floor connections (D'Ayala & Speranza, 2003).

In monumental masonry structures, earthquake damage may also affect architectural components such as façades, bell towers, arches, vaults, and domes. Their structural response can involve localized cracking, separation between connected elements, instability, and progressive failure. Lagomarsino (2012), in an investigation of churches damaged during the 2009 L'Aquila earthquake, emphasized the importance of identifying damage mechanisms associated with individual architectural components. Similarly, Ruggieri (2021) investigated seismic damage in 14 historic masonry churches following the 2020 Cosenza earthquake, illustrating how structural geometry, masonry organization, boundary conditions, and previous interventions can influence observed damage patterns.

The identification of these mechanisms is essential because visible cracks do not necessarily indicate the same structural condition. Crack orientation, distribution, width, and location may provide valuable information about the underlying failure mechanism; however, their interpretation requires knowledge of the structural system and loading conditions. In addition, pre-existing cracks and material deterioration should be distinguished from newly developed earthquake-induced damage whenever sufficient evidence is available. Therefore, the classification of seismic damage in historic masonry structures should consider not only visible surface deterioration but also its potential relationship with structural behavior and stability.

2.3. Conventional Post-Earthquake Damage Assessment

Post-earthquake damage assessment of historic masonry structures is essential for identifying structural deterioration, evaluating potential safety concerns, and supporting conservation and restoration decisions. Conventional assessment methods generally rely on visual inspections conducted by structural engineers, architects, and cultural heritage specialists. These inspections typically involve documenting visible cracks, masonry displacements, material loss, partial collapses, and other indicators of structural distress. The collected observations are subsequently interpreted in relation to the building's construction characteristics, structural configuration, and possible failure mechanisms (Lagomarsino et al., 2019).

Several established approaches are available for evaluating earthquake-induced damage in masonry structures. The European Macroseismic Scale (EMS-98), developed under the editorship of Grünthal (1998), describes five damage grades for masonry buildings, ranging from negligible or slight damage (Grade 1) to destruction (Grade 5). These grades are associated with observable physical damage, including cracking, partial structural failures, and collapse. Although such classifications provide a useful framework for describing earthquake effects, the assignment of a damage grade does not, by itself, constitute a complete structural safety assessment of an individual historic building.

For monumental structures, more specialized damage assessment procedures have also been developed. Lagomarsino et al. (2019) investigated post-earthquake damage assessment methodologies for masonry churches and proposed rapid and detailed assessment forms based on the identification of structural macroelements and associated seismic damage mechanisms. Such approaches provide a more systematic interpretation of observed damage and support the prioritization of further investigations and conservation interventions.

In addition to visual inspection, conventional structural assessment may involve historical documentation, geometric surveys, material characterization, nondestructive testing, structural monitoring, and numerical analysis. According to ICOMOS (2003), the diagnosis of architectural heritage structures should integrate historical, qualitative, and quantitative approaches to establish the causes and significance of structural deterioration. Roca et al. (2010) also highlighted the role of analytical and numerical methods in understanding the structural response of historic masonry constructions, while acknowledging the difficulties associated with material characterization, computational modeling, and limited knowledge of existing structural conditions.

Despite their importance, conventional inspection methods present several practical limitations. Detailed surveys can be time-consuming and may require access to structurally unstable or difficult-to-reach areas. Furthermore, visual damage interpretation may vary among inspectors, particularly when distinguishing between surface deterioration, pre-existing defects, and earthquake-induced structural damage. These challenges highlight the potential value of automated image analysis for supporting damage documentation and preliminary classification. However, automated identification of visible damage should be regarded as complementary to, rather than a replacement for, engineering-based structural assessment. This distinction provides an important foundation for examining the application of deep learning methods to post-earthquake damage detection and classification in historic masonry structures.

3. Deep Learning Methods for Damage Detection

3.1. Review Approach and Literature Selection

This chapter adopts a focused critical narrative review approach, emphasizing deep learning methods relevant to damage detection and classification in historic masonry structures. Particular attention is given to studies involving convolutional neural networks, transfer learning, object detection, and image segmentation. Research directly examining historic masonry structures is prioritized, while studies involving conventional masonry structures are considered where they provide transferable methodological insights. The reviewed methods are organized according to their primary functions: image classification, damage localization, and pixel-level segmentation. This task-based

framework supports the comparison of their capabilities, limitations, and potential contributions to post-earthquake damage assessment.

3.2. Convolutional Neural Networks and Transfer Learning

Convolutional Neural Networks (CNNs) are among the most widely investigated deep learning architectures for image-based damage detection. CNNs employ convolutional layers to automatically learn hierarchical visual features, including edges, textures, shapes, and complex spatial patterns. These characteristics make them suitable for identifying structural defects such as cracks, material deterioration, and surface irregularities in masonry structures.

Several CNN architectures, including AlexNet, VGG, ResNet, and MobileNet, have been employed in structural damage recognition applications. Wang et al. (2018) investigated CNN-based classification of multiple surface damage categories in historic masonry structures, demonstrating the feasibility of automatically distinguishing cracks, efflorescence, spalling, and intact surfaces. Similarly, Dais et al. (2021) applied CNN-based approaches to masonry crack classification and segmentation, highlighting the potential of deep learning under complex visual conditions.

One important challenge in training CNN models for historic masonry damage detection is the limited availability of sufficiently large and annotated datasets. Historic structures exhibit considerable variations in material composition, surface texture, geometry, and deterioration patterns, which may reduce the effectiveness of models trained on general-purpose image datasets. Transfer learning provides a potential solution by adapting models previously trained on large datasets to more specific damage detection tasks. This approach can reduce training requirements and improve feature extraction when only limited domain-specific data are available. Dais et al. (2021) demonstrated the benefits of transfer learning for masonry crack identification. Nevertheless, the applicability of transferred models to different historic structures requires careful validation because variations in construction materials and environmental conditions may affect prediction performance.

3.3. Object Detection Models

Unlike image classification methods, which assign labels to entire images or image patches, object detection models identify both the category and spatial location of damage. These models typically represent detected regions using bounding boxes, enabling the localization of cracks, spalling, and other visible defects.

Two commonly investigated object detection approaches are the You Only Look Once (YOLO) family and Faster Region-Based Convolutional Neural Networks (Faster R-CNN). YOLO-based models perform object localization and classification within a unified detection framework, making them suitable for

applications requiring rapid image processing (Redmon et al., 2016). In contrast, Faster R-CNN employs a region proposal network followed by classification and bounding-box refinement, providing a framework for detecting visually complex objects (Ren et al., 2015).

In cultural heritage applications, object detection models have been investigated for automatically locating deterioration and damage on historic building surfaces. ElBehairy et al. (2026) reviewed the application of YOLO and Faster R-CNN architectures to cultural heritage damage detection, emphasizing their potential for identifying damaged regions while highlighting differences in computational requirements and localization capabilities. For post-earthquake inspections of historic masonry structures, these methods may support the rapid identification and documentation of visible defects. However, bounding-box localization does not precisely describe crack geometry or establish the structural significance of the detected damage.

3.4. Semantic Segmentation Models

Semantic segmentation extends image-based damage analysis by assigning a class label to individual pixels rather than classifying an entire image or identifying rectangular regions. This capability is particularly relevant to masonry crack analysis, where irregular crack paths and narrow damaged regions may be difficult to represent accurately using bounding boxes.

U-Net is a widely adopted segmentation architecture based on an encoder–decoder structure with skip connections that combine contextual information with spatial details (Ronneberger et al., 2015). Other approaches, such as DeepLabV3+, employ multiscale feature extraction to improve segmentation accuracy, particularly around complex object boundaries (Chen et al., 2018).

Dais et al. (2021) investigated pixel-level crack segmentation on masonry surfaces using deep learning architectures, demonstrating the applicability of segmentation for identifying complex crack patterns. Loverdos and Sarhosis (2022) further examined deep learning-based segmentation of masonry walls using architectures including U-Net and DeepLabV3+, highlighting their potential for identifying masonry units and supporting crack localization and quantification.

For historic masonry structures, segmentation models can provide detailed spatial information about visible damage and may facilitate the estimation of crack extent and distribution. However, their performance depends on image quality, annotation accuracy, and the availability of representative training data. Furthermore, converting pixel-level predictions into physically meaningful crack measurements requires appropriate geometric calibration or additional spatial information.

3.5. Performance Evaluation Metrics

The performance of deep learning models for damage detection and classification is commonly assessed using quantitative evaluation metrics. For image classification, accuracy, precision, recall, and the F1-score are frequently employed to measure the correctness and reliability of predictions. Precision evaluates the proportion of predicted positive instances that are correctly identified, whereas recall measures the proportion of actual positive instances successfully detected. The F1-score combines precision and recall into a single measure, making it particularly useful when damage classes are imbalanced.

For object detection, mean Average Precision (mAP) is widely used to evaluate detection performance across different classes and localization thresholds. In image segmentation, Intersection over Union (IoU) and pixel-level F1-score are commonly used to assess the agreement between predicted damage regions and manually annotated reference masks. These metrics provide complementary information regarding model performance and should be selected according to the specific damage assessment task (ElBehairy et al., 2026).

However, reported performance values cannot always be directly compared across studies because of differences in datasets, damage categories, annotation procedures, and validation strategies. For example, high classification accuracy does not necessarily indicate precise crack localization or reliable estimation of structural damage severity. In addition, evaluation using images from the same structure may overestimate performance when compared with testing on previously unseen buildings. Therefore, the assessment of deep learning models for historic masonry damage detection should consider both conventional performance metrics and their generalization across different structures, materials, and post-earthquake conditions.

Overall, CNN-based classification, object detection, and semantic segmentation provide complementary capabilities for the automated analysis of visible damage in historic masonry structures. Their integration with engineering-based assessment procedures may improve post-earthquake damage documentation and support more systematic structural evaluations. Nevertheless, the relationship between automatically detected visual defects and the actual seismic damage state of historic masonry buildings requires further investigation.

4. Deep Learning-Based Damage Detection and Classification

4.1. Crack Identification and Classification

Crack identification is one of the most extensively investigated applications of deep learning in masonry damage assessment. Cracks represent important visual indicators of structural distress and may provide information about the distribution and development of damage in historic masonry structures. However, their appearance can vary considerably depending on masonry materials, surface

conditions, structural characteristics, and environmental factors. These variations create challenges for conventional image-processing approaches and have encouraged the development of deep learning-based classification methods.

Wang et al. (2018) developed a convolutional neural network (CNN)-based approach for identifying and classifying multiple types of surface damage in historic masonry structures. The proposed method employed AlexNet and GoogLeNet architectures with a sliding-window strategy to distinguish four categories: intact surfaces, cracks, efflorescence, and spalling. Using images collected from the Forbidden City Wall in China, the study reported a classification accuracy of 94.3%. The results demonstrated the potential of CNNs for automatically recognizing different surface conditions in historic masonry. Nevertheless, the training and testing data originated from a single heritage site, which limits conclusions regarding the generalizability of the proposed approach to other masonry materials and construction techniques.

Transfer learning has also been investigated to address the limited availability of annotated masonry damage images. Katsigiannis et al. (2023) examined several pre-trained CNN architectures for classifying cracked and non-cracked brick masonry façades. Their dataset consisted of 700 images, divided into training, validation, and testing subsets. Different training strategies, including data augmentation and fine-tuning, were investigated. The best-performing configurations achieved classification accuracy and F1-scores of up to 100%. Although these results indicate the effectiveness of transfer learning under the examined experimental conditions, the relatively limited dataset highlights the importance of evaluating such models on independent buildings and different masonry surfaces.

More recently, Bozzano et al. (2026) investigated automated crack detection in historic churches using the Xception architecture and transfer learning. The model was initially trained using concrete crack images and subsequently evaluated on church surfaces characterized by frescoes, plaster, and decorative elements. The authors observed a considerable reduction in performance when the model trained on concrete images was applied directly to church images. After fine-tuning with church-specific data, the model achieved an F1-score of 91.08% for frescoed vault surfaces. This study highlights the importance of domain-specific training data and demonstrates that models developed for conventional construction materials may not perform equally well on historic masonry surfaces.

Overall, these investigations indicate that CNN-based models can effectively identify visible cracks and distinguish between different surface conditions. However, crack classification results should not be interpreted as direct measurements of seismic damage severity. The structural significance of detected

cracks depends on additional factors, including their location, geometry, distribution, and relationship with possible failure mechanisms.

4.2. Damage Localization

While image classification determines whether a particular type of damage is present, damage localization identifies its position within an image. This capability is particularly important for inspecting large masonry façades and architectural components where multiple damaged regions may occur simultaneously.

Wang et al. (2019) proposed an automatic damage detection framework for historic masonry buildings using Faster R-CNN with a ResNet101 backbone. The method focused on two types of surface deterioration: efflorescence and spalling. The proposed detection model achieved a mean Average Precision (mAP) of 95.0% under the reported evaluation conditions. The researchers also developed smartphone-based and webcam-assisted inspection systems to demonstrate the feasibility of automated on-site damage detection. Although the investigation was not specifically concerned with earthquake-induced damage, it provided an important example of the practical application of deep learning-based localization in historic masonry structures.

Object detection methods have also been investigated for identifying cracks on complex historic masonry surfaces. Pratibha et al. (2024) applied a YOLOv5-based detection model to masonry crack images collected from historic sites in India. The study employed approximately 4,000 annotated images and reported an mAP at an Intersection over Union threshold of 0.5 of approximately 92%. The results indicate the potential of YOLO-based approaches for rapid crack localization, particularly when damage documentation must be performed across numerous structural surfaces.

An alternative approach was presented by Hacıfendioğlu et al. (2023), who developed a framework combining pre-trained CNN models, Gradient-weighted Class Activation Mapping (Grad-CAM), and K-means clustering. Their CAM-K-SEG approach was designed to identify and localize crack regions in historical masonry structures. The study also employed U-Net for comparison, distinguishing the broader localization capabilities of CAM-K-SEG from the more detailed pixel-level segmentation provided by U-Net. This distinction is important because identifying the approximate position of a crack does not necessarily provide accurate information about its dimensions or geometric characteristics.

These investigations demonstrate that deep learning-based localization can support the documentation of visible damage in historic masonry structures. However, localization performance depends on the quality of annotated images, the complexity of surface textures, and the ability of models to distinguish

structural defects from visually similar features. Furthermore, bounding-box predictions provide limited information about crack continuity and physical dimensions, which may be essential for subsequent engineering assessment.

4.3. Damage Segmentation and Quantification

Image segmentation provides a more detailed representation of structural damage by identifying affected regions at the pixel level. In historic masonry structures, this capability is particularly relevant for examining irregular crack patterns and estimating their spatial distribution. Compared with image classification and object detection, segmentation can provide more precise geometric information, although it generally requires more detailed annotations.

Dais et al. (2021) investigated CNN-based crack classification and segmentation on masonry surfaces using transfer learning. The authors developed a dataset containing images with complex backgrounds and different crack patterns. Their approach achieved 95.3% accuracy for patch-level crack classification and an F1-score of 79.6% for pixel-level segmentation. These results demonstrate that successful classification does not necessarily imply equally accurate identification of crack boundaries. The study also emphasized the benefits of transfer learning for improving crack detection on masonry surfaces.

Loverdos and Sarhosis (2022) investigated automated brick segmentation and crack detection using several deep learning architectures, including U-Net, DeepLabV3+, LinkNet, and Feature Pyramid Networks. Their study focused on the identification of masonry units and the geometric characterization of cracks. The results highlighted the potential of segmentation methods for producing digital representations of masonry surfaces and supporting the automated documentation of structural features.

Similarly, Minh Dang et al. (2022) developed a deep learning-based framework for masonry crack segmentation and real-life crack length measurement. The study compared several segmentation architectures, including U-Net, DeepLabV3+, and Feature Pyramid Networks, using a dataset of masonry crack images. An important contribution was the estimation of physical crack length through the identification of brick dimensions, illustrating how pixel-level predictions can be transformed into quantitative damage measurements when appropriate geometric reference information is available.

A particularly relevant investigation concerning earthquake-induced damage was conducted by Pantoja-Rosero et al. (2022). The authors proposed a topology-oriented loss function, referred to as TOPO-Loss, for segmenting cracks in earthquake-damaged masonry buildings. Their approach employed TeraNet and focused on improving the continuity of detected cracks, especially when training annotations were imprecise. The study introduced the Cracks Per Patch

metric to evaluate crack continuity and provided a dataset containing images of earthquake-affected masonry and urban environments. These contributions are important because discontinuous segmentation predictions may incorrectly represent connected cracks and complicate subsequent interpretation of damage patterns.

Although segmentation models can provide detailed representations of crack geometry, several limitations remain. Pixel-level measurements are influenced by image resolution, viewing angle, annotation accuracy, and geometric calibration. Furthermore, the proportion of a surface classified as damaged does not necessarily correspond directly to the structural severity of earthquake-induced damage. Consequently, segmentation outputs should be interpreted alongside structural information and appropriate engineering criteria.

4.4. Applications in Historic Masonry Structures

Historic masonry presents distinctive visual challenges, including irregular masonry arrangements, weathered textures, decorative elements, previous repairs, and complex geometries that can obscure cracks or produce false detections. Studies by Wang et al. (2018, 2019), Hacıfendioğlu et al. (2023), and Bozzano et al. (2026) illustrate the applicability of deep learning to heritage surfaces, while emphasizing the need for context-specific validation.

However, these applications do not all concern earthquake-induced damage. Some investigations focus on efflorescence or spalling without establishing a seismic cause. Conversely, Pantoja-Rosero et al. (2022) addressed earthquake-damaged masonry, but their study was not restricted to historic heritage buildings. This distinction between historic masonry and earthquake-damaged masonry is important when interpreting the applicability of existing results.

Accordingly, models intended for post-earthquake inspection of historic masonry should be tested across structures, materials, and damage conditions. Automated classification, localization, and segmentation may support damage documentation, but the determination of structural damage states requires additional engineering knowledge and consideration of possible failure mechanisms.

5. Post-Earthquake Damage Assessment

5.1. Damage Types and Damage Severity

Post-earthquake damage assessment of historic masonry structures involves identifying visible damage and evaluating its potential implications for structural performance and stability. As discussed in previous sections, earthquakes may produce various damage patterns, including diagonal and horizontal cracking, separation between masonry elements, out-of-plane wall deformation, material loss, and partial or complete collapse. However, these manifestations do not

necessarily correspond to the same level of structural damage. Their significance depends on the structural configuration, masonry properties, loading conditions, and failure mechanisms involved (D'Ayala & Speranza, 2003).

A distinction must therefore be established between damage type and damage severity. Damage type describes the physical manifestation of deterioration, such as cracking, spalling, or displacement, whereas damage severity refers to the extent of structural impairment resulting from these phenomena. The European Macroseismic Scale (EMS-98) provides a widely recognized framework for describing earthquake damage through five grades, ranging from negligible or slight damage to destruction (Grünthal, 1998). Nevertheless, damage grades based on observable effects should not be considered equivalent to a comprehensive evaluation of structural safety or residual load-bearing capacity.

Recent developments in deep learning have encouraged the investigation of automated approaches for identifying damage severity from visual information. Hacıfendioğlu et al. (2024) proposed a deep learning-based methodology for estimating damage levels in masonry structures following earthquakes. Their approach employed a U-Net segmentation model to identify damaged regions and calculate a damage index based on the ratio between detected damaged surface area and the total visible structural surface area. The model achieved approximately 92% precision and 93% recall in damage detection. These findings suggest that automated segmentation can support the quantitative characterization of visible earthquake damage. However, the resulting image-based damage index should be distinguished from an engineering-based assessment of structural capacity.

The assessment of damage severity in historic masonry structures is particularly challenging because their irregular geometries, heterogeneous materials, and distinctive architectural features may influence both damage development and visual interpretation. Accordingly, automated damage classification should be considered within the broader context of structural characteristics and possible seismic failure mechanisms.

5.2. Image-Based Damage Indicators

Image-based damage assessment methods attempt to extract quantitative information from visible damage patterns. Common indicators include crack length, width, orientation, density, spatial distribution, and the proportion of surface area affected by damage. Deep learning-based classification and segmentation techniques provide opportunities to estimate these characteristics automatically, potentially reducing the time required for post-earthquake inspections.

Hacıfendioğlu et al. (2024) investigated the use of segmented damage regions to construct an image-based damage index for masonry structures. By

normalizing the detected damaged area with respect to the visible structural surface, their approach provided a quantitative representation of damage extent. Such an indicator may facilitate preliminary comparisons of surface conditions, although its interpretation depends on the viewing perspective, image quality, and the relationship between visible damage and the underlying structural condition.

Other studies have investigated crack geometry as a potential indicator of seismic damage. Diaz et al. (2025) developed a deep learning-based approach for estimating damage levels in confined masonry walls by analyzing crack patterns and their relationship with lateral deformation. Their methodology combined crack segmentation, crack length quantification, and experimentally derived relationships between crack measurements and structural drift. The resulting damage index was used to estimate damage levels associated with the observed deformation response.

The study by Diaz et al. (2025) is particularly relevant because it moves beyond the identification of cracks toward the interpretation of visual damage through experimentally established structural response parameters. Nevertheless, the investigation focused on confined masonry walls, which differ structurally from many historic unreinforced masonry buildings. Consequently, direct application of the proposed relationships to historic masonry structures would require further experimental validation and consideration of differences in construction techniques and structural behavior.

Overall, image-based indicators provide a potentially useful connection between automated damage detection and quantitative assessment. However, their engineering significance depends on whether the extracted visual features can be reliably associated with deformation, stiffness degradation, residual capacity, or other structural performance parameters.

5.3. Relationship Between Visual Damage and Structural Condition

One of the main challenges in automated post-earthquake assessment is establishing a reliable relationship between visible damage and the actual structural condition of masonry buildings. Although crack patterns can provide information about possible failure mechanisms, visual observations alone may not adequately describe the remaining strength, stiffness, or displacement capacity of a damaged structure.

Pereira et al. (2024) investigated the prediction of residual drift capacity in damaged unreinforced masonry walls using an ensemble of convolutional neural networks. Their approach employed numerically generated crack patterns and represented crack characteristics through crack width exceedance curves. These curves were associated with drift loss, defined in relation to the displacement capacity of corresponding undamaged walls. The resulting CNN-based

framework demonstrated the potential to estimate residual deformation capacity from quantified crack patterns.

An important contribution of this study is the incorporation of mechanically consistent information into the learning process. Rather than relying solely on visual damage categories, the proposed methodology establishes a relationship between crack characteristics and structural response obtained through numerical simulations. Nevertheless, the applicability of such predictions to historic masonry structures depends on the representativeness of the numerical models, material assumptions, and structural configurations considered during training.

More recently, Fraternali et al. (2026) investigated an artificial intelligence-assisted methodology for the preliminary post-earthquake assessment of masonry structures. The study employed a dataset of 250 photographs collected during inspections of buildings affected by earthquakes in Italy. A CNN-based detection framework was used to identify earthquake-induced damage patterns, including different crack orientations and masonry spalling. The detected damage information was subsequently incorporated, through engineering-based interpretation, into simplified no-tension mechanical models to support preliminary evaluations of the stability and residual seismic capacity of masonry components.

This approach illustrates the potential advantages of integrating automated visual damage recognition with structural mechanics. Unlike methods that rely exclusively on image-based damage indices, the proposed framework incorporates mechanical considerations into the assessment process. However, the relatively limited image dataset and the preliminary nature of the methodology indicate that further validation is necessary before such approaches can be applied routinely to diverse historic masonry buildings.

Together, these investigations suggest an emerging research direction in which deep learning-based damage detection serves as an input to engineering-oriented structural evaluation. Such integration may contribute to more informative post-earthquake assessments, particularly when visual observations are combined with structural geometry, material characteristics, and mechanically meaningful performance indicators.

5.4. Limitations of Automated Damage-State Classification

Despite recent advances, several limitations restrict the use of deep learning models for automatically determining the seismic damage states of historic masonry structures. A fundamental issue is that visible surface damage does not always provide sufficient evidence of the overall structural condition. Similar crack patterns may arise from different loading histories or failure mechanisms, while significant structural weaknesses may exist without clearly observable surface damage.

The relationship between segmentation accuracy and damage-state classification also requires careful consideration. A model may successfully identify cracked regions while providing insufficient information about their structural significance. For example, the proportion of damaged pixels does not necessarily correspond to the loss of load-bearing capacity or seismic resistance. Similarly, a damage classification model trained using laboratory-derived damage levels may not be directly applicable to historic masonry structures with different geometries and mechanical properties.

The investigation by Diaz et al. (2025) illustrates some of these challenges. Although their proposed method successfully established relationships between crack measurements and experimentally observed deformation, its damage-level classification performance varied across severity categories, with lower reliability reported for severe damage and collapse conditions. This finding emphasizes the importance of evaluating models across all damage levels rather than relying exclusively on overall accuracy.

Furthermore, uncertainty in image acquisition, annotation procedures, and structural interpretation can influence automated predictions. Variations in illumination, camera position, image resolution, surface occlusion, and crack visibility may affect the quality of extracted damage indicators. For historic masonry structures, additional uncertainty arises from architectural complexity and the difficulty of distinguishing earthquake-induced damage from pre-existing defects.

Therefore, automated damage-state classification should be regarded primarily as a decision-support capability rather than a replacement for expert structural assessment. Reliable post-earthquake evaluation requires the integration of visual information with engineering knowledge, structural investigations, and, where appropriate, analytical or numerical methods. Future developments should focus on establishing validated relationships between deep learning outputs and structural performance indicators, particularly for historic masonry buildings subjected to actual seismic events.

6. Comparative Discussion of Existing Studies

6.1. Comparison of Deep Learning Approaches

The reviewed methods address different prediction tasks and should not be ranked solely by their reported accuracy. Image classification can recognize damage categories but does not provide precise crack geometry (Wang et al., 2018; Katsigiannis et al., 2023). Object detection locates damaged regions but commonly relies on bounding boxes that cannot fully represent irregular defects (Wang et al., 2019). Segmentation offers pixel-level detail, although precise crack boundaries and continuity remain challenging (Dais et al., 2021; Pantoja-Rosero et al., 2022).

These approaches therefore provide complementary capabilities. Classification is suitable for initial screening, detection for locating visible defects, and segmentation for more detailed surface characterization. As the studies differ in datasets, damage categories, and evaluation protocols, model selection should be driven by inspection objectives, annotation availability, and the intended use of the outputs in subsequent structural assessment.

6.2. Datasets and Evaluation Strategies

The characteristics of training and evaluation datasets play a critical role in determining the reliability of deep learning models. Existing studies vary considerably in terms of dataset size, image acquisition conditions, damage categories, and annotation procedures.

Several investigations have relied on relatively limited or site-specific datasets. For example, Wang et al. (2018) used images from the Forbidden City Wall, while Katsigiannis et al. (2023) investigated crack classification using 700 masonry façade images. Although high performance was reported, such datasets may not adequately represent the diversity of materials and architectural characteristics encountered in historic masonry structures.

Transfer learning and data augmentation have been employed to address limited training data. However, their effectiveness depends on the similarity between the original training domain and the intended application. Bozzano et al. (2026) demonstrated this challenge by observing reduced performance when a model trained on concrete crack images was applied to historic church surfaces. Subsequent fine-tuning with church-specific images substantially improved classification performance.

Another important concern is the comparability of evaluation metrics. Accuracy and F1-score are commonly reported for classification, while mAP is used for object detection and Intersection over Union (IoU) or pixel-level F1-score for segmentation. Consequently, performance values obtained from different tasks cannot be interpreted as equivalent measures of model effectiveness.

Moreover, random image-level or patch-level dataset splitting may produce optimistic results when visually similar samples from the same structure appear in both training and testing subsets. Evaluating models on independent buildings and previously unseen masonry types would provide stronger evidence of their generalization capabilities.

6.3. Generalization to Historic Masonry Structures

Generalizing deep learning models to historic masonry structures remains challenging due to irregular masonry arrangements, varying mortar compositions, decorative elements, previous repairs, and pre-existing

deterioration. These characteristics may reduce the performance of models trained on conventional masonry or concrete surfaces. Bozzano et al. (2026) demonstrated the importance of domain-specific fine-tuning for crack detection in historic churches, while Wang et al. (2018, 2019) highlighted the need to validate models developed for individual heritage sites across different structures.

Although Dais et al. (2021) and Pantoja-Rosero et al. (2022) introduced valuable approaches for improving masonry crack segmentation and continuity, their findings cannot be directly generalized to historic structures with different construction characteristics. Therefore, reliable real-world applications require diverse heritage-specific datasets, cross-building validation, and systematic evaluation under varying environmental and image acquisition conditions.

6.4. Visual Damage Detection versus Structural Damage Assessment

Deep learning methods can identify visible cracks and surface deterioration, but these outputs alone cannot establish seismic damage severity or residual structural capacity. Recent studies have explored approaches linking visual damage to structural performance. Hacıfendioğlu et al. (2024) developed an image-based damage index using segmented regions, while Pereira et al. (2024) related crack patterns to residual drift capacity through mechanically consistent numerical data. Similarly, Diaz et al. (2025) associated crack measurements with experimentally derived deformation-based damage levels in confined masonry walls. Fraternali et al. (2026) further integrated CNN-based damage recognition with simplified mechanical models for preliminary post-earthquake assessment.

Despite these advances, differences in structural typology, datasets, and assessment objectives limit the direct transferability of existing findings to historic unreinforced masonry. Approaches based on laboratory specimens, numerical simulations, or confined masonry therefore require further validation under realistic post-earthquake conditions.

Table 1 summarizes selected studies in terms of their objectives, methodological contributions, and limitations.

The reported performance results are not directly comparable because the studies employ different datasets, prediction tasks, and evaluation procedures. Furthermore, the identified limitations primarily concern study scope and applicability rather than necessarily reflecting methodological weaknesses acknowledged by the authors. Overall, reliable post-earthquake assessment requires heritage-specific datasets, cross-site validation, uncertainty analysis, and the integration of visual damage indicators with mechanically meaningful structural performance measures.

Table 1. Comparative summary of selected deep learning studies relevant to masonry damage detection and post-earthquake assessment.

Study	Application and approach	Main contribution or reported result	Main limitation
Wang et al. (2018)	Historic masonry; CNN classification	Four surface categories; 94.3% accuracy	Single heritage site
Wang et al. (2019)	Historic masonry; Faster R-CNN	Efflorescence and spalling detection; 95.0% mAP	Not specific to seismic damage
Dais et al. (2021)	Masonry; CNN classification and segmentation	95.3% classification accuracy; 79.6% segmentation F1	No structural damage-state estimation
Pantoja-Rosero et al. (2022)	Earthquake-damaged masonry; TernaUSNet and TOPO-Loss	Improved crack continuity	No direct structural capacity estimation
Katsigiannis et al. (2023)	Brick masonry; CNN and transfer learning	Up to 100% accuracy and F1-score	Limited dataset diversity
Hacıfendioğlu et al. (2024)	Earthquake-affected masonry; U-Net	Image-based damage index	Visible damage extent is not equivalent to structural safety
Pereira et al. (2024)	Damaged masonry walls; CNN ensemble	Prediction of residual drift capacity	Reliance on numerical simulations
Diaz et al. (2025)	Confined masonry; deep learning and crack measurements	Deformation-related damage estimation	Limited transferability to historic unreinforced masonry
Bozzano et al. (2026)	Historic churches; Xception and transfer learning	91.08% F1-score on frescoed vault surfaces	Strong dependence on target-domain data
Fraternali et al. (2026)	Post-earthquake masonry; CNN and mechanical models	Preliminary structural assessment	Further validation and broader datasets required

7. Challenges and Future Research Directions

Despite advances in deep learning-based damage detection and classification, several challenges remain in applying these methods to the post-earthquake assessment of historic masonry structures. Their practical implementation requires improvements in dataset quality, model generalization, damage quantification, and integration with engineering-based structural assessment.

A major limitation is the scarcity of large, diverse, and accurately annotated datasets representing earthquake damaged historic masonry structures. Existing studies often rely on images from individual heritage sites, conventional masonry surfaces, or laboratory specimens, which may not capture the diversity of construction materials, architectural characteristics, and seismic damage patterns. Moreover, pixel level annotation requires considerable expertise and effort (Dais et al., 2021; ElBehairy et al., 2026). Future research should prioritize standardized, publicly accessible heritage-specific datasets with consistent damage annotations and, where available, structural information.

Model generalization across different historic structures remains another challenge. Variations in surface textures, decorative elements, and construction techniques may reduce the performance of models trained on conventional materials. Bozzano et al. (2026) highlighted the importance of domain-specific adaptation when transferring crack detection models from concrete surfaces to historic churches. Future studies should therefore investigate transfer learning, domain adaptation, and data augmentation strategies, supported by cross-building and cross-site validation.

Accurate damage segmentation and quantification also require further investigation. Narrow cracks, irregular geometries, and varying illumination conditions can affect segmentation reliability. Pantoja-Rosero et al. (2022) demonstrated the importance of preserving crack continuity through topology-oriented learning. Future approaches should incorporate geometric characteristics, crack connectivity, and spatial measurement techniques, including photogrammetry and three-dimensional reconstruction, to improve damage quantification.

Integrating deep learning with structural engineering represents a particularly important research direction. Although most existing approaches focus on visible defects, Pereira et al. (2024) investigated relationships between crack patterns and residual drift capacity, while Fraternali et al. (2026) combined CNN-based damage recognition with simplified mechanical models. These developments suggest the potential of hybrid frameworks linking image analysis with structural performance. However, their applicability to historic masonry requires further validation using representative structural configurations, material properties, and real earthquake damage data.

Finally, practical implementation requires reliable and interpretable predictions under real-world conditions. Future investigations should address uncertainty estimation, explainable artificial intelligence, false-negative detections, and the distinction between pre-existing deterioration and earthquake-induced damage. Computational efficiency and integration with inspection workflows should also be considered, particularly for mobile and unmanned vehicle-based aerial applications. Ultimately, automated methods should

complement rather than replace the expertise of structural engineers and cultural heritage specialists.

8. Conclusion

This chapter presented a critical review of deep learning-based methods for the detection and classification of post-earthquake damage in historic masonry structures. The reviewed studies demonstrated the potential of convolutional neural networks, transfer learning, object detection, and semantic segmentation techniques for identifying cracks, classifying surface deterioration, localizing damaged regions, and generating detailed damage maps. These approaches offer promising opportunities to improve the efficiency and consistency of visual damage documentation, particularly in historic structures characterized by complex geometries and heterogeneous materials.

The comparative analysis highlighted that different deep learning approaches provide complementary capabilities. While image classification methods enable the rapid identification of visible damage, object detection and segmentation techniques offer increasingly detailed spatial information. However, reported performance results vary considerably depending on the datasets, damage categories, and evaluation procedures employed. Furthermore, the limited availability of heritage-specific datasets and differences between conventional and historic masonry surfaces continue to restrict model generalization.

An important finding of this review is that accurate identification of visible damage does not necessarily provide a reliable assessment of seismic damage severity or structural safety. Recent investigations integrating image-based damage indicators with deformation-related parameters and mechanical models represent promising developments toward more comprehensive post-earthquake assessment. Nevertheless, the applicability of these approaches to diverse historic masonry structures requires further experimental and field-based validation.

Future research should prioritize the development of representative earthquake-damage datasets, robust cross-site evaluation strategies, improved geometric damage quantification, and the integration of deep learning with engineering-based structural assessment. Ultimately, deep learning should be considered a complementary decision-support tool rather than a replacement for expert structural evaluation. Its responsible integration into post-earthquake inspection procedures may contribute to more systematic damage documentation, informed conservation decisions, and the long-term preservation of historic masonry heritage.

References

- Bozzano, M., Eid, M. O., Cattari, S., & Federici, B. (2026). Crack detection in churches using deep learning. *Applied Geomatics*, 18, 94. <https://doi.org/10.1007/s12518-026-00743-w>
- Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *Proceedings of the European Conference on Computer Vision (ECCV)*. https://doi.org/10.1007/978-3-030-01234-2_49
- Dais, D., Bal, İ. E., Smyrou, E., & Sarhosis, V. (2021). Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. *Automation in Construction*, 125, 103606. <https://doi.org/10.1016/j.autcon.2021.103606>
- D'Ayala, D., & Speranza, E. (2003). Definition of collapse mechanisms and seismic vulnerability of historic masonry buildings. *Earthquake Spectra*, 19(3), 479–509. <https://doi.org/10.1193/1.1599896>
- Diaz, M., Lopez, L., Amancio, M., Inocente, I., Salinas, J., Isuhuaylas, S., Flores, E., & Moscoso, E. (2025). Crack-based estimation of seismic damage level in confined masonry walls in the Lima Metropolitan Area using deep learning techniques. *Applied Sciences*, 15(11), 5875. <https://doi.org/10.3390/app15115875>
- ElBehairy, A., Abu El-Nasr, N. A., Grimberg, P., & Said, L. A. (2026). A comprehensive review of deep learning methods in damage classification, detection, and segmentation of cultural heritage sites. *Journal of Cultural Heritage*, 78, 228-237. <https://doi.org/10.1016/j.culher.2026.01.015>
- Fraternali, F., Etteyeb, H., Lato, A., Nazifi Charandabi, R., Spagnuolo, M., Olivieri, C., Fabbrocino, F., Ciaramella, A., & Amendola, A. (2026). On the use of artificial intelligence and no-tension models in the post-earthquake preliminary assessment of masonry structures. *Applications in Engineering Science*, 26, 100332. <https://doi.org/10.1016/j.apples.2026.100332>
- Grünthal, G. (Ed.). (1998). *European Macroseismic Scale 1998 (EMS-98)*. Cahiers du Centre Européen de Géodynamique et de Séismologie, Vol. 15.
- Hacıfendioğlu, K., Altunışık, A. C., & Abdioğlu, T. (2023). Deep learning-based automated detection of cracks in historical masonry structures. *Buildings*, 13(12), 3113. <https://doi.org/10.3390/buildings13123113>
- Hacıfendioğlu, K., Özgan, K., Adanur, S., Altunışık, A. C., Demirer, B., & Günaydın, M. (2024). Estimation of damage levels in masonry structures following earthquake impact using deep learning-based segmentation method. *Journal of Earthquake and Tsunami*, 18(2). <https://doi.org/10.1142/S1793431123500367>

- ICOMOS. (2003). *Principles for the analysis, conservation and structural restoration of architectural heritage*. International Council on Monuments and Sites.
- Katsigiannis, S., Seyedzadeh, S., Agapiou, A., & Ramzan, N. (2023). Deep learning for crack detection on masonry façades using limited data and transfer learning. *Journal of Building Engineering*, 76, 107105. <https://doi.org/10.1016/j.jobbe.2023.107105>
- Lagomarsino, S. (2012). Damage assessment of churches after L'Aquila earthquake (2009). *Bulletin of Earthquake Engineering*, 10(1), 73–92. <https://doi.org/10.1007/s10518-011-9307-x>
- Lagomarsino, S., Cattari, S., Ottonelli, D., & Giovinazzi, S. (2019). Earthquake damage assessment of masonry churches: Proposal for rapid and detailed forms and derivation of empirical vulnerability curves. *Bulletin of Earthquake Engineering*, 17, 3327–3364. <https://doi.org/10.1007/s10518-018-00542-8>
- Loverdos, D., & Sarhosis, V. (2022). Automatic image-based brick segmentation and crack detection of masonry walls using machine learning. *Automation in Construction*, 140, 104389. <https://doi.org/10.1016/j.autcon.2022.104389>
- Minh Dang, L., Wang, H., Li, Y., Nguyen, L. Q., Nguyen, T. N., Song, H. K., & Moon, H. (2022). Deep learning-based masonry crack segmentation and real-life crack length measurement. *Construction and Building Materials*, 359, 129438. <https://doi.org/10.1016/j.conbuildmat.2022.129438>
- Pantoja-Rosero, B. G., Oner, D., Kozinski, M., Achanta, R., Fua, P., Perez-Cruz, F., & Beyer, K. (2022). TOPO-Loss for continuity-preserving crack detection using deep learning. *Construction and Building Materials*, 344, 128264. <https://doi.org/10.1016/j.conbuildmat.2022.128264>
- Pereira, M., D'Altri, A. M., de Miranda, S., & Glisic, B. (2024). Crack pattern-based machine learning prediction of residual drift capacity in damaged masonry walls. *Computer-Aided Civil and Infrastructure Engineering*, 39(24), 3685–3699. <https://doi.org/10.1111/mice.13212>
- Pratibha, K., Mishra, M., Ramana, G. V., & Lourenço, P. B. (2024). Deep learning-based YOLO network model for detecting surface cracks during structural health monitoring. In Y. Endo & T. Hanazato (Eds.), *Structural analysis of historical constructions: SAHC 2023* (RILEM Bookseries, Vol. 47, pp. 179–187). Springer. https://doi.org/10.1007/978-3-031-39603-8_15
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788. <https://doi.org/10.1109/CVPR.2016.91>

- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 91–99.
- Roca, P., Cervera, M., Gariup, G., & Pelà, L. (2010). Structural analysis of masonry historical constructions: Classical and advanced approaches. *Archives of Computational Methods in Engineering*, 17, 299-325 (2010). <https://doi.org/10.1007/s11831-010-9046-1>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 234-241. https://doi.org/10.1007/978-3-319-24574-4_28
- Ruggieri, N. (2021). Damage assessment of historical masonry churches subjected to moderate intensity seismic shaking. *Sustainability*, 13(9), 4710. <https://doi.org/10.3390/su13094710>
- Wang, N., Zhao, Q., Li, S., Zhao, X., & Zhao, P. (2018). Damage classification for masonry historic structures using convolutional neural networks based on still images. *Computer-Aided Civil and Infrastructure Engineering*, 33(12), 1073-1089. <https://doi.org/10.1111/mice.12411>
- Wang, N., Zhao, X., Zhao, P., Zhang, Y., Zou, Z., & Ou, J. (2019). Automatic damage detection of historic masonry buildings based on mobile deep learning. *Automation in Construction*, 103, 53-66. <https://doi.org/10.1016/j.autcon.2019.03.003>

CHAPTER 11

Comparative Evaluation of Polarization and Magnification in Petrographic Thin Section Classification

Rukiye Uzun Arslan¹

Mualla Toksan²

¹ Assoc. Prof., Department of Electrical and Electronics Engineering, Faculty of Engineering, Zonguldak Bülent Ecevit University, 67100 Zonguldak, Türkiye
ORCID:0000-0002-2082-8695

² Lecturer (Corr. Auth.), Department of Mining and Mineral Extraction, Çaycuma Vocational School, Zonguldak Bülent Ecevit University, 67900 Çaycuma, Zonguldak, Türkiye, ORCID: 0000-0001-8682-485X

1 Introduction

Petrographic thin-section images contain optical and textural information that can support rock description. The University of Toronto Scarborough Virtual Petrography Resource provides an educational collection of thin sections and hand samples intended for examining mineral composition and texture (Daxberger and Kennedy, 2026). A classification model trained on such images addresses a narrower task than a complete petrographic interpretation: it predicts the labels represented in its training data. Its classification score alone does not demonstrate mineral identification, quantitative mineralogy, or geological interpretation.

Previous research provides a methodological basis for comparing plane-polarized light (PPL) and crossed-polarized light (XPL) representations. Ma et al. (2025) investigated feature fusion of these image modalities for lithology identification. Their findings motivate the evaluation of combined representations, although the direction and magnitude of a fusion-associated difference remain dependent on the implemented model and dataset. The present comparison therefore examines the consistency of the observed differences across configurations without presupposing a classification benefit from optical complementarity.

The comparison includes two established architectural families. Residual networks employ residual connections, as represented here by ResNet50 (He et al., 2016). Swin Transformer constructs hierarchical image representations through shifted attention windows (Liu et al., 2021). These architectural characteristics provide a basis for comparative evaluation, while relative classification performance is assessed from the implemented models under the documented partition and training conditions.

The principal research question concerns variation in classification performance between ResNet50 and Swin-Tiny within the five-class, point-partitioned experiments when polarization and magnification conditions are evaluated individually, pooled as training samples, or combined through feature concatenation. A secondary question concerns whether feature fusion yields higher scores than the corresponding unfused input configurations. The analytical framework accommodates positive differences, negligible differences, and decreases, without assuming an advantage from either additional input images or the Transformer architecture.

2 Materials and methods

2.1 Data source and class labels

The images were obtained from Virtual Petrography, an open educational resource of the University of Toronto Scarborough, credited to Heidi Daxberger and Kirsten Kennedy (Daxberger and Kennedy, 2026). The website describes the

Virtual Petrography Lab as a database of geological thin sections and hand samples and specifies a Creative Commons Attribution 4.0 International license. Images with available rock-class names were selected. The retained targets were dolostone, limestone, quartz sandstone, olivine basalt, and schist.

The primary criteria for class selection were the availability of a sufficient number of distinct physical imaging locations, complete image coverage across polarization and objective magnification conditions at the same locations, and the feasibility of assigning imaging locations to separate training, validation, and test groups. Based on these criteria, five rock classes were selected: dolostone, limestone, quartz sandstone, olivine basalt, and schist.

A common feature of the five selected classes is the availability of all four imaging conditions (4× PPL, 4× XPL, 10× PPL, and 10× XPL) at each physical XY imaging location. Each condition comprises 36 angular images acquired at approximately 10° intervals from 0° to 350°, yielding 144 images per imaging location across the four conditions. Imaging the same physical location under different polarization states, magnifications, and rotation angles enables comparisons of imaging conditions within the same sample area. This approach aims to support more controlled comparative experiments by reducing the extent to which variability between imaging locations confounds performance differences across imaging conditions.

The number of distinct imaging locations was another criterion for class selection. The dataset contains 35 physical imaging locations: 14 for dolostone, five for limestone, four for quartz sandstone, four for olivine basalt, and eight for schist. This distribution allowed at least one imaging location from each rock class to be assigned to each of the validation and test sets. Other rock types with insufficient imaging locations were excluded because their inclusion could have required treating angular images from the same physical location as independent samples.

The selection of the five rock classes was therefore based on rock-type diversity, the experimental integrity of the dataset, and its suitability for comparative analyses. The availability of four imaging conditions and 36 angular images per condition at the same physical locations enables controlled investigation of the effects of imaging parameters on classification performance, while location-based data partitioning reduces the risk of data leakage arising from related images.

The retained files encode an imaging-point identifier, a magnification label, a polarization label, and a nominal angle. Each point contributes 36 angles, from 0° to 350° at 10° intervals, for each of the four combinations 4x PPL, 4x XPL, 10x PPL and 10x XPL. Thus, each point contributes 144 images. Table 1 describes the subset used in the completed experiments. Point identifiers

distinguish recorded views; they do not establish independent physical specimens.

Table 1 Class composition and recorded point assignments

Class	Points	Images	Training points	Validation points	Test points
Dolostone	14	2016	10	2	2
Limestone	8	1152	6	1	1
Quartz sandstone	4	576	2	1	1
Olivine basalt	4	576	2	1	1
Schist	8	1152	6	1	1
Total	38	5472	26	6	6

2 2 Data organization and image metadata

The retained five-class subset is organized into rock-class directories, each containing four imaging conditions defined by two magnification labels and two polarization channels. The directory hierarchy includes only the conditions used in the completed experiments (Figure 1).

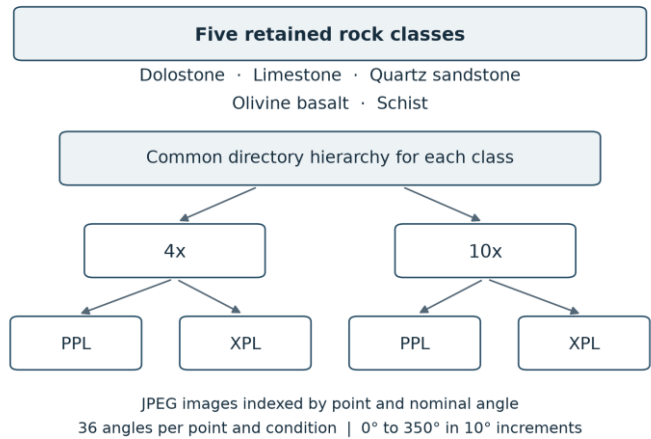


Figure 1 Directory organization of the experimental subset. Each of the five class directories contains the 4x and 10x branches, each with PPL and XPL images indexed by point and nominal angle. Only conditions included in the completed comparison are shown.

Image identifiers in the retained subset contain fields describing the rock class, point identifier, magnification, polarization, and nominal angle. The example represents dolostone point P00018 under 4× PPL at a nominal angle of 0.0°. The two compatibility fields contain zeros and are not interpreted as measured spatial coordinates. The spatial coordinate fields were introduced to ensure code and data compatibility in subsequent studies. Point identifiers establish correspondence for data partitioning and tuple assembly but do not establish independent specimen identity or pixel-level registration (Figure 2).

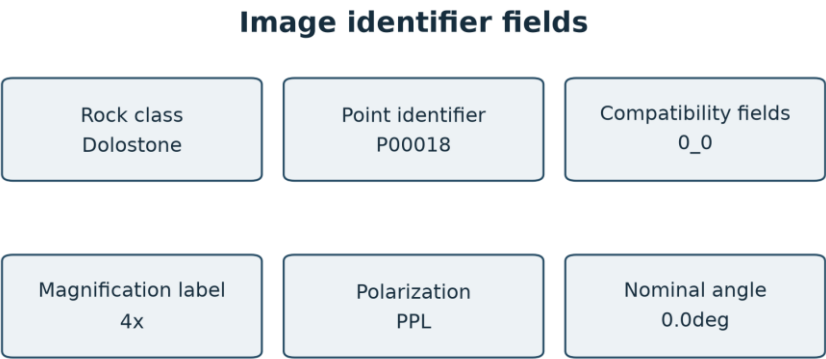


Figure 2 Interpretation of identifier fields for a retained dolostone image from point P00018.

2 3 Illustrative thin-section image set

Actual thin-section images represent all five retained rock classes under the four input conditions. For each class, the lowest-numbered test point in the recorded partition is displayed at a fixed nominal angle of 0° (Figures 3–7).

Dolostone is a carbonate sedimentary rock composed predominantly of dolomite [CaMg(CO₃)₂], commonly formed through the diagenetic replacement of limestone by magnesium-rich fluids. Petrographically, it typically exhibits a crystalline mosaic of dolomite rhombs, with variable preservation of the original depositional fabric (Tucker & Wright, 1990). In plane-polarized light (PPL), dolomite is generally colorless and non-pleochroic, with relief that varies with crystal orientation; under crossed polars (XPL), it's very high birefringence produces high-order interference colors (Figure 3)(Deer, Howie, & Zussman, 2013).

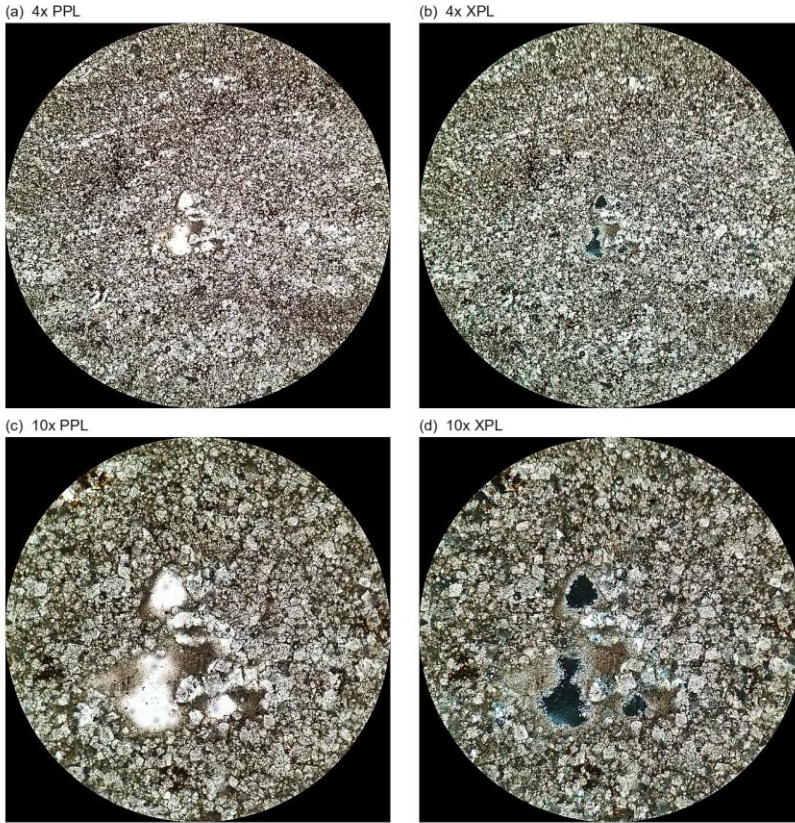


Figure 3 Thin-section images labelled dolostone from point P00018 at nominal angle 0 degrees. (a) 4x PPL; (b) 4x XPL; (c) 10x PPL; (d) 10x XPL. (Daxberger and Kennedy, 2026)

Limestone is a carbonate sedimentary rock composed predominantly of calcite (CaCO_3), formed through the accumulation of carbonate skeletal material, carbonate precipitation, or a combination of these processes. Its petrographic fabric may include bioclasts, ooids, and other carbonate grains within a fine-grained micritic matrix or sparry calcite cement (Tucker & Wright, 1990). In plane-polarized light (PPL), calcite is generally colorless and non-pleochroic, with relief that varies with crystal orientation. Under crossed polars (XPL), individual calcite crystals exhibit very high birefringence and high-order interference colors and may display lamellar twinning (Figure 4) (Deer, Howie, & Zussman, 2013).

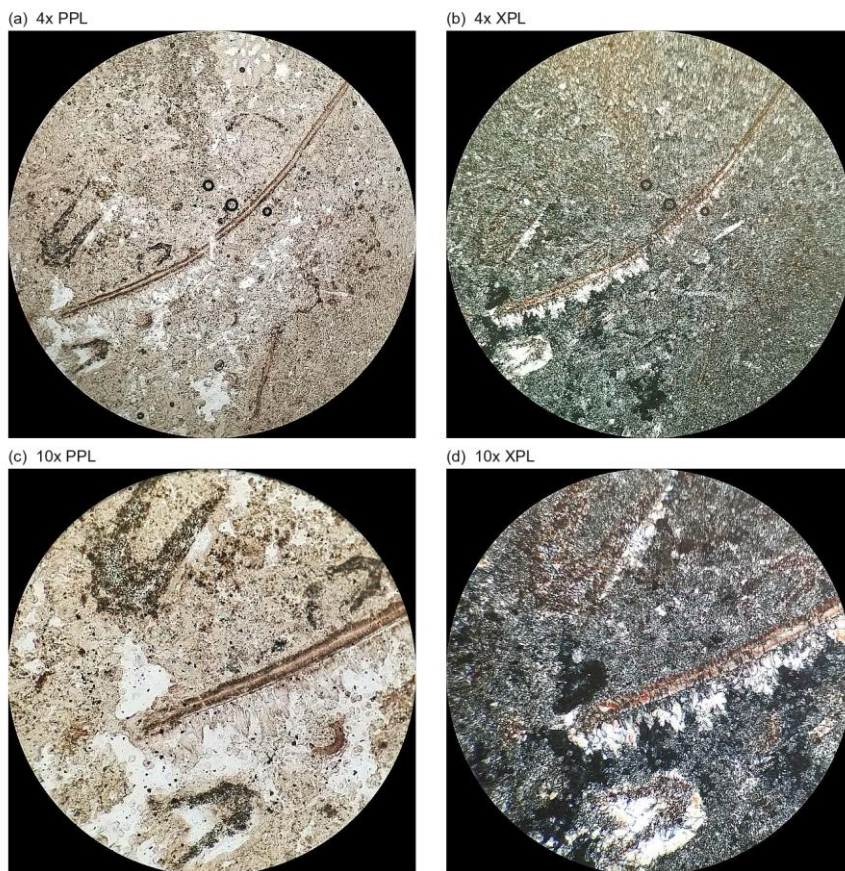


Figure 4 Thin-section images labelled limestone from point P00027 at nominal angle 0 degrees. (a) 4x PPL; (b) 4x XPL; (c) 10x PPL; (d) 10x XPL. (Daxberger and Kennedy, 2026)

Quartz sandstone is a siliciclastic sedimentary rock dominated by sand-sized quartz grains, formed through the deposition and subsequent lithification of quartz-rich sediment. Its petrographic fabric is characterized by grain size, sorting, roundness, and the distribution of intergranular matrix and cement, which may include silica or carbonate minerals (Boggs, 2009). In plane-polarized light (PPL), quartz is colorless and non-pleochroic, with low relief and no cleavage. Under crossed polars (XPL), it typically displays first-order grey to white interference colors, while grains affected by lattice strain may exhibit undulose extinction (Figure 5) (Deer, Howie, & Zussman, 2013).

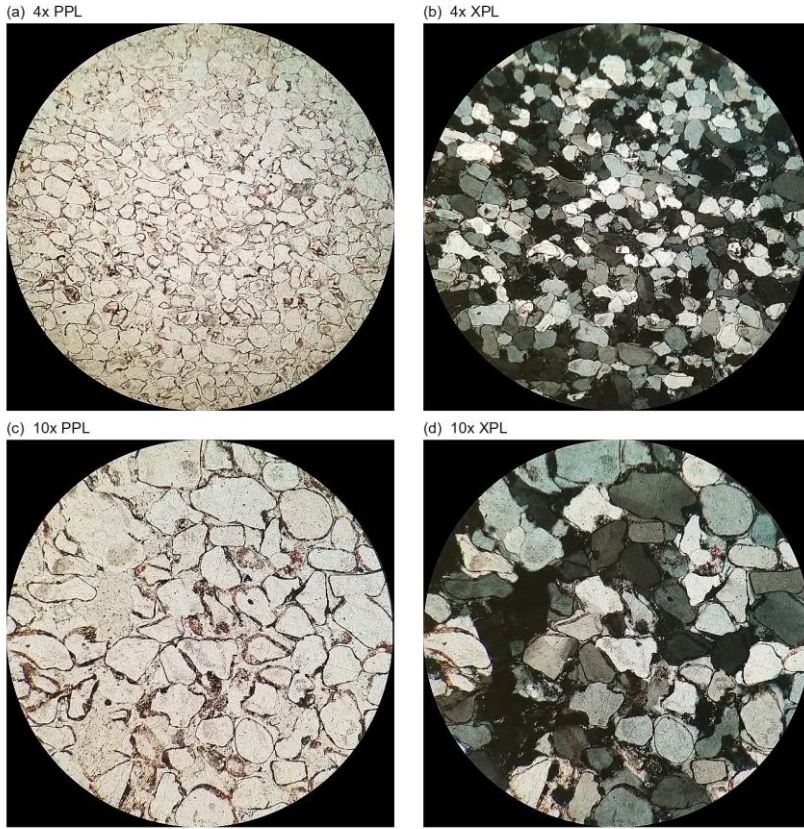


Figure 5 Thin-section images labelled quartz sandstone from point P00064 at nominal angle 0 degrees. (a) 4x PPL; (b) 4x XPL; (c) 10x PPL; (d) 10x XPL. (Daxberger and Kennedy, 2026)

Olivine basalt is a mafic volcanic rock formed by the cooling and solidification of basaltic lava, with olivine occurring alongside plagioclase and clinopyroxene. Its petrographic fabric may be fine-grained or porphyritic, with olivine phenocrysts enclosed in a finer-grained groundmass (Philpotts & Ague, 2009). In plane-polarized light (PPL), fresh olivine is generally colorless and non-pleochroic, with high relief, irregular fractures, and no prominent cleavage. Under crossed polars (XPL), its relatively high birefringence commonly produces bright second-order interference colors, contrasting with the lower-order colors of plagioclase (Figure 6) (Deer, Howie, & Zussman, 2013).

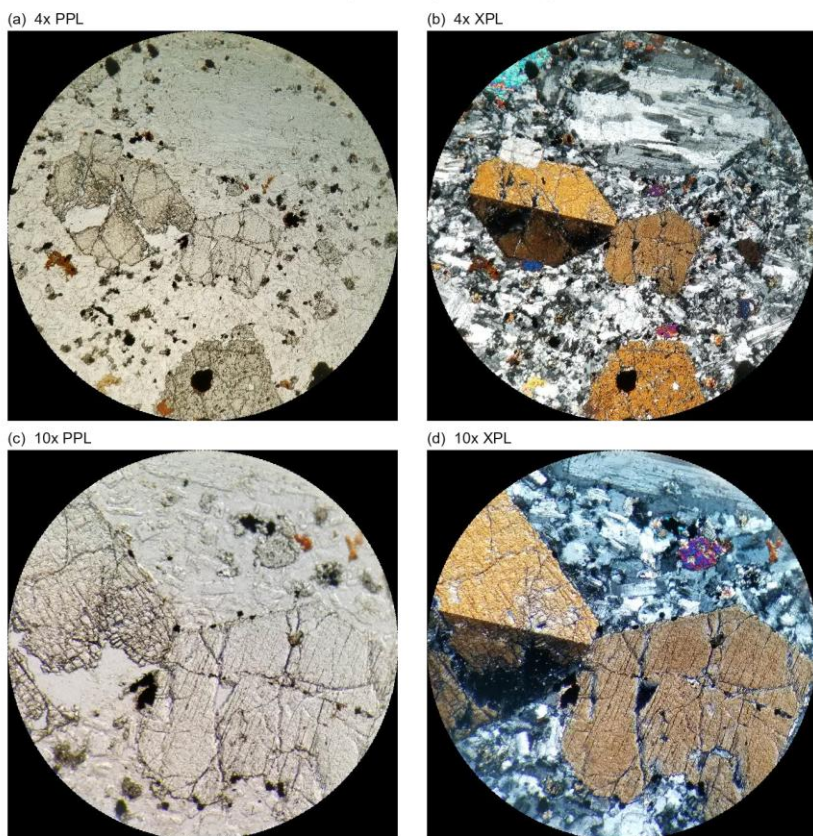


Figure 6 Thin-section images labelled olivine basalt from point P00057 at nominal angle 0 degrees. (a) 4x PPL; (b) 4x XPL; (c) 10x PPL; (d) 10x XPL. (Daxberger and Kennedy, 2026)

Schist is a foliated metamorphic rock commonly formed during regional metamorphism, in which recrystallization and deformation produce a preferred orientation of platy or elongate minerals. Its characteristic schistosity is often defined by aligned mica flakes associated with quartz and feldspar, although the mineral assemblage varies with protolith composition and metamorphic conditions (Philpotts & Ague, 2009). In mica-bearing schists, muscovite is generally colorless in plane-polarized light (PPL), whereas biotite commonly displays brown or green colors and marked pleochroism. Under crossed polars (XPL), muscovite typically exhibits bright second- to third-order interference colors, while biotite's interference colors may be partly masked by its intrinsic color (Figure 7) (Deer, Howie, & Zussman, 2013).

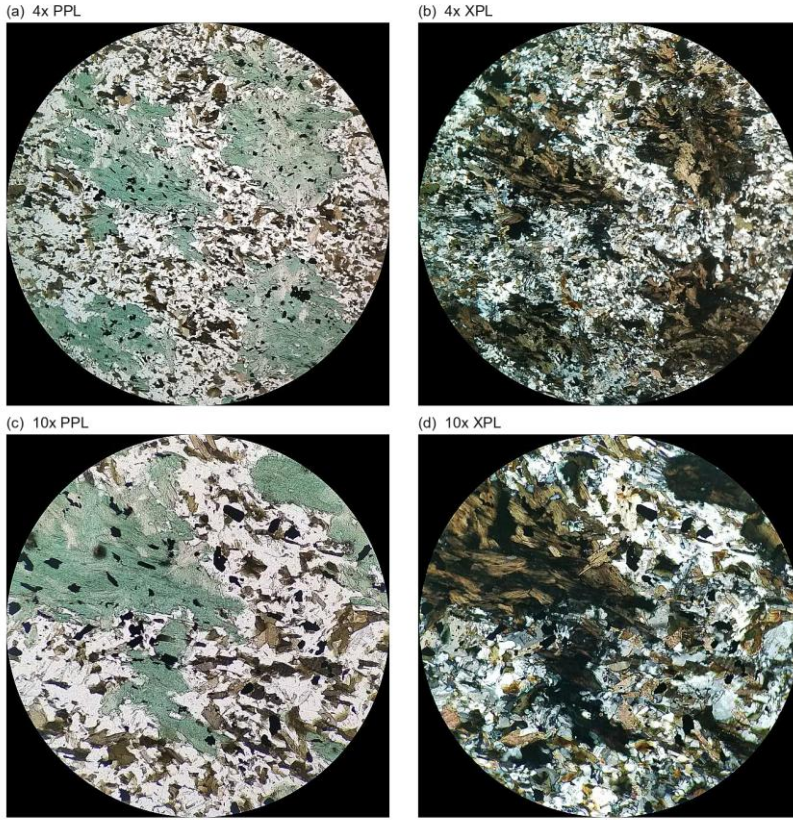


Figure 7 Thin-section images labelled schist from point P00076 at nominal angle 0 degrees. (a) 4x PPL; (b) 4x XPL; (c) 10x PPL; (d) 10x XPL. (Daxberger and Kennedy, 2026)

2 4 Partitioning and units of observation

The saved partition uses the combined class and point identifier as its grouping key. All angles, polarization channels and magnification labels belonging to that key remain in the same training, validation, or test partition. The split seed is 42. Although the configured target fractions were 70%, 15% and 15%, allocation within the small classes yielded the actual counts in Table 1. The corresponding stored-image counts are 3,744 for training, 864 for validation and 864 for testing. Every scenario and both model families use these same point assignments.

The six test points are dolostone P00018 and P00030, limestone P00027, quartz sandstone P00064, olivine basalt P00057 and schist P00076. Keeping each point intact prevents its angular frames from being assigned to different partitions. The grouping key does not include an original source-sample or thin-section identifier, however, so the design does not demonstrate separation of physical specimens. Views from a common source could remain related across

partitions. Consequently, the evaluation concerns held-out imaging points in this subset.

The evaluation distinguishes classification samples from imaging points. A classification sample comprises either an individual image or a matched image tuple processed jointly by a fusion model. Point-level prediction is obtained by averaging output class probabilities across all classification samples associated with a test point and selecting the class with the maximum mean probability. Although the original reports designate sample-level predictions as image-level results, a fusion prediction corresponds to a multi-image tuple. The term sample level is used throughout this chapter to preserve this distinction.

2 5 Input configurations and pairing

The 11 completed scenarios use pooling or fusion to combine imaging conditions. In pooling, images from different conditions are treated as separate samples and processed through a single branch. In fusion, conditions sharing the same class, point identifier, and nominal angle are paired, their learned features are concatenated, and a single prediction is generated for each tuple. In A3, pairing is performed within each magnification, with the 4× and 10× pairs treated as separate samples. In B3, pairing is performed within each polarization, with the PPL and XPL pairs treated as separate samples. Scenario D combines all four conditions into a single tuple (Table 2).

Table 2 Completed input configurations and classification-sample counts.

ID	Input configuration	Branches	Train	Validation	Test
A1	PPL at 4x and 10x pooled	1	1872	432	432
A2	XPL at 4x and 10x pooled	1	1872	432	432
A3	PPL + XPL fused within each magnification	2	1872	432	432
B1	PPL and XPL at 4x pooled	1	1872	432	432
B2	PPL and XPL at 10x pooled	1	1872	432	432
B3	4x + 10x fused within each polarization	2	1872	432	432
C1	4x PPL	1	936	216	216
C2	4x XPL	1	936	216	216
C3	10x PPL	1	936	216	216
C4	10x XPL	1	936	216	216
D	4x PPL + 4x XPL + 10x PPL + 10x XPL	4	936	216	216

All scenarios cover the same 38 points, but the number of classification samples and images per sample differs. A1, A2, B1 and B2 use 72 single-image samples per point. A3 and B3 use 72 two-image samples per point. C1–C4 use 36 single-image samples per point, and D uses 36 four-image samples per point. Consequently, a common point partition does not make these configurations equal in information volume, training examples, or model capacity. Comparisons are descriptive assessments of the implemented configurations.

2 6 Preprocessing and model implementation

Both implementations convert images to RGB, resize directly to 224×224 pixels, and normalize the channels using means of 0.485, 0.456 and 0.406 and standard deviations of 0.229, 0.224 and 0.225. The implemented transform contains no random augmentation, field-of-view masking, color normalization, or registration operation. Resizing changes the computational input dimensions; it does not equate the physical field of view between source magnification labels.

The ResNet50 implementation removes the original final classifier and uses a 2,048-dimensional feature vector per branch. The Swin-Tiny implementation replaces its classification head with an identity operation and uses a 768-dimensional vector per branch. The branch vectors are concatenated and passed to a linear classifier with five outputs. Multi-branch models use separate backbones because parameter sharing is disabled. The resulting models therefore differ in parameter count as branches are added; the four-branch Swin-Tiny run records 110,092,781 trainable parameters, compared with 27,523,199 in a single-branch run.

2 7 Training and recorded evaluation

Training used a batch size of 16, five epochs and stochastic gradient descent with a learning rate of 0.001, momentum of 0.9 and weight decay of 0.0005. The random seed was 42. Cross-entropy loss was unweighted, and the optimizer received the model parameters without a frozen-encoder stage. Mixed precision was enabled conditionally for CUDA. The available code selects CUDA when available and otherwise uses the CPU.

The selected checkpoint maximizes validation macro-F1 over the five epochs, with lower validation loss breaking a tie. The saved checkpoint is then evaluated on the test partition. The reported outcomes are sample-level accuracy and macro-F1, point-level accuracy and selected timing and checkpoint-size records. Macro-F1 gives equal weight to the five class-specific F1 scores, whereas accuracy counts correct classifications among all samples. All macro-F1 values in this chapter are expressed on a 0–100 scale.

2 8 Operational characteristics of pooling and feature fusion

Pooling and fusion differ in the relationships among observations presented to the classifier. In A2, 4x XPL and 10x XPL images are processed as separate training samples by a single branch. In A3, PPL and XPL images at a common magnification are processed by separate branches within the same forward pass, after which their feature vectors are concatenated. The common point partition therefore controls point membership but does not render the input representations equivalent.

In D, each training sample comprises the four imaging conditions at a common nominal angle, and each prediction is conditional on their joint representation. The comparison with C2 evaluates the performance of the implemented joint representation relative to the implemented single-condition representation. It does not estimate the effect of supplying three additional channels to an otherwise unchanged C2 model, because the number of backbones and the classifier input dimensionality also differ. These structural differences constrain the attribution of performance differences to information content alone.

The implementations specify two-branch and four-branch configurations in which the corresponding feature vectors are concatenated without a learned reduction to a shared 256-dimensional representation. The branch structures and feature dimensionalities are summarized accordingly (Table 3).

Table 3 Structural differences among the implemented configuration families

Family	Samples per point	Images per sample	ResNet50 features	Swin-Tiny features
A1 A2 B1 B2	72	1	2048	768
A3 B3	72	2	4096	1536
C1 C2 C3 C4	36	1	2048	768
D	36	4	8192	3072

2 9 Evaluation metrics and interpretive framework

Accuracy measures the fraction of correct sample classifications. Class precision describes the fraction of predictions assigned to a class that matches its reference label, while recall describes the fraction of reference-class samples recovered. F1 combines precision and recall, and macro-F1 averages the class F1 value equally. ROC-AUC instead evaluates probability ranking over thresholds rather than only the final winning class (Pedregosa et.al, 2011; scikit-learn Developers, 2026). These measures answer different questions and are reported without treating one as a substitute for the others.

For the present test partition, dolostone contributes twice the sample support of every other class because two dolostone points were held out. This affects the relative contribution of dolostone to ordinary sample accuracy. Macro-F1 assigns each of the five class F1 values the same weight, but the underlying class measurements still come from one or two points. The weighting convention changes the summary calculation; it does not add independent class examples or establish the stability of a class estimate.

The evaluation distinguishes between sample-level outcomes and point-level predictions. Point-level accuracy is based on the label selected from the mean probability vector; it does not represent the proportion of correctly classified angular frames or a majority-vote statistic. Probability averaging may yield the reference label despite disagreement among constituent samples. Conversely, an alternative class may attain the highest mean probability, resulting in an incorrect aggregated prediction. Both sample-level outcomes and point-level predictions are therefore retained in the detailed evaluation (Table 4).

Table 4 Evaluation units and interpretation of reported performance measures

Quantity	Unit	Interpretive use
Sample accuracy	Image or matched tuple	Overall frequency of correct classifications
Macro-F1	Five class F1 values	Equal-weight summary of class precision and recall
Class precision and recall	Samples for one label	Distinguishes false assignments from missed reference samples
Point accuracy	Six held-out points	Correctness after averaging sample probabilities
ROC-AUC	Sample probability rankings	Ranking across thresholds rather than winning-label accuracy
Recorded training time	One run	Elapsed time for the implemented training loop

2 10 Experimental workflows

The completed experiments begin with point-based partitioning, followed by the construction of condition-specific sample lists, with identical point assignments maintained across both architectural families. Preprocessing and the specified input structure are applied within each partition. Each model is trained independently, and checkpoints are selected based on validation macro-F1, with validation loss used to resolve ties. The selected checkpoint is then evaluated on the retained test points (Figure 8).

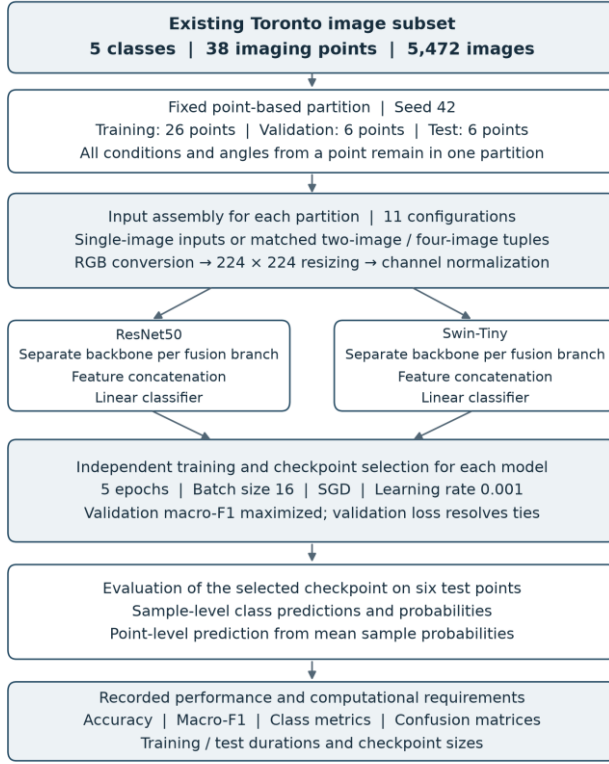


Figure 8 Workflow of the completed comparative experiments.

3 Results

3 1 Comparative performance across experimental configurations

Performance across the 11 scenarios varied with both the input condition and the model backbone. Among the single-condition scenarios, 4× XPL (C2) yielded the highest recorded sample-level macro-F1 for both models, reaching 97.84% for ResNet50 and 100.00% for Swin-Tiny. The relative performance of the models differed across the remaining conditions: ResNet50 achieved a higher score under 4× PPL (C1), whereas Swin-Tiny achieved a higher score under 10× PPL (C3). These results do not establish an architectural advantage independent of the input configuration (Table 5).

Table 5 Recorded test performance for both model families

ID	ResNet50 accuracy %	ResNet50 macro- F1 %	Swin-Tiny accuracy %	Swin-Tiny macro-F1 %
A1	84.72	83.65	83.10	83.42
A2	92.36	92.75	90.51	89.71
A3	90.74	90.71	98.15	97.99
B1	87.27	87.77	85.88	85.64
B2	91.90	90.48	89.81	87.45
B3	90.74	90.63	89.35	88.32
C1	81.48	74.48	67.59	55.56
C2	97.69	97.84	100.00	100.00
C3	69.91	61.32	87.96	87.17
C4	88.89	87.02	82.87	81.64
D	91.20	90.72	100.00	100.00

The 100.00% values in C2 and D for Swin-Tiny refer to 216 correct classification-sample predictions in each scenario, obtained from the same six test points. C2 evaluates single images, while D evaluates four-image tuples. The equality of their scores therefore does not imply identical representations or equivalent computational cost. The additional conditions were not associated with a higher measured score than the single-condition configuration on this partition.

Sample-level accuracy and macro-F1 are presented for all configurations on common 0–100 scales. The horizontal axis denotes categorical experimental configurations rather than a continuous magnification or fusion variable. Connecting lines aid comparison without implying that intermediate configurations were evaluated. The largest differences between models occur under the individual PPL conditions, with their relative ordering reversed between C1 and C3. Smaller differences are observed in the pooled configurations, indicating that model comparisons depend on the input definition (Figure 9).

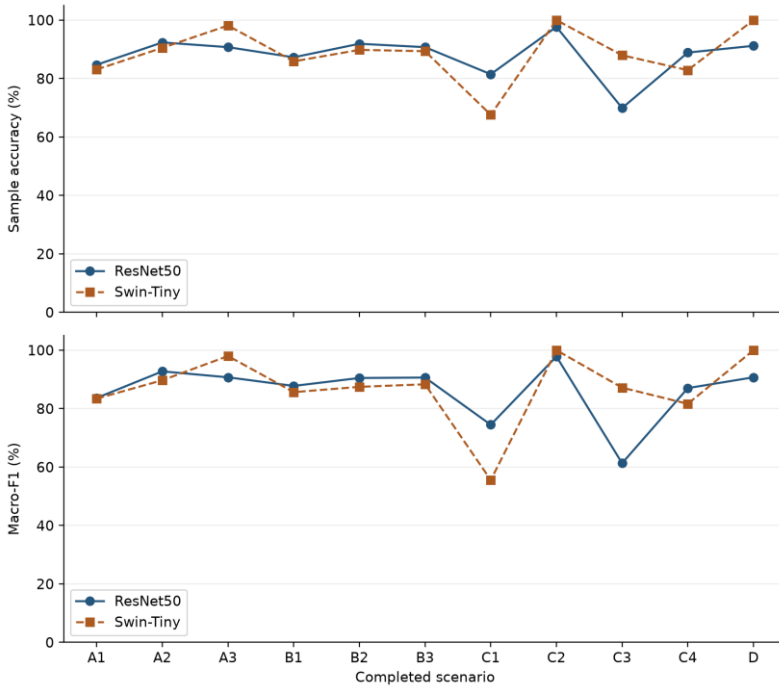


Figure 9 Sample-level accuracy and macro-F1 across the 11 experimental configurations.

3 2 Polarization-specific classification and feature fusion

When magnification labels were pooled, XPL alone (A2) had a higher macro-F1 than PPL alone (A1) for both backbones. The recorded values were 92.75% versus 83.65% for ResNet50 and 89.71% versus 83.42% for Swin-Tiny. The existing comparison summary gives differences of 9.10 and 6.29 percentage points, respectively. This is an observed association within the tested configurations, rather than a causal estimate of polarization or a universal preference for XPL.

PPL–XPL feature fusion (A3) showed a different pattern between models. Swin-Tiny recorded 97.99% macro-F1, exceeding its A2 score by 8.28 percentage points in the existing summary. ResNet50 recorded 90.71%, which was 2.04 percentage points below its A2 result. The backbones therefore did not exhibit a consistent improvement under polarization fusion. Because A3 also uses two separately trainable backbones and two images per sample, the comparison cannot attribute the differences solely to complementary optical information.

3 3 Magnification-specific classification performance

With polarization channels pooled, the 10x configuration (B2) recorded higher macro-F1 than the 4x configuration (B1) for both models: 90.48% versus 87.77% for ResNet50 and 87.45% versus 85.64% for Swin-Tiny. The

corresponding differences in the existing summary were 2.71 and 1.81 percentage points. Magnification fusion (B3) yielded 90.63% and 88.32%, respectively, 0.15 and 0.87 percentage points above B2. With a single split and no uncertainty estimates, these small differences do not establish a stable advantage.

The single-condition comparisons qualify the interpretation of the pooled magnification results. Under XPL, the 4x configuration yielded higher scores than the 10x configuration for both models. Under PPL, the higher-scoring magnification was 4x for ResNet50 and 10x for Swin-Tiny. These outcomes concern source-labelled images subjected to the same computational resizing procedure. They do not separate physical objective magnification from field-of-view differences, image content, or optimization characteristics.

3 4 Classification errors under four-condition feature fusion

In scenario D, ResNet50 recorded 91.20% sample-level accuracy and 90.72% macro-F1, whereas Swin-Tiny recorded 100.00% for both measures. Four-condition fusion did not exceed the highest single-condition macro-F1 for either model: ResNet50 remained below its C2 value, while Swin-Tiny matched its C2 result. Figure 10 and Figure 11 show the corresponding sample-level confusion matrices.

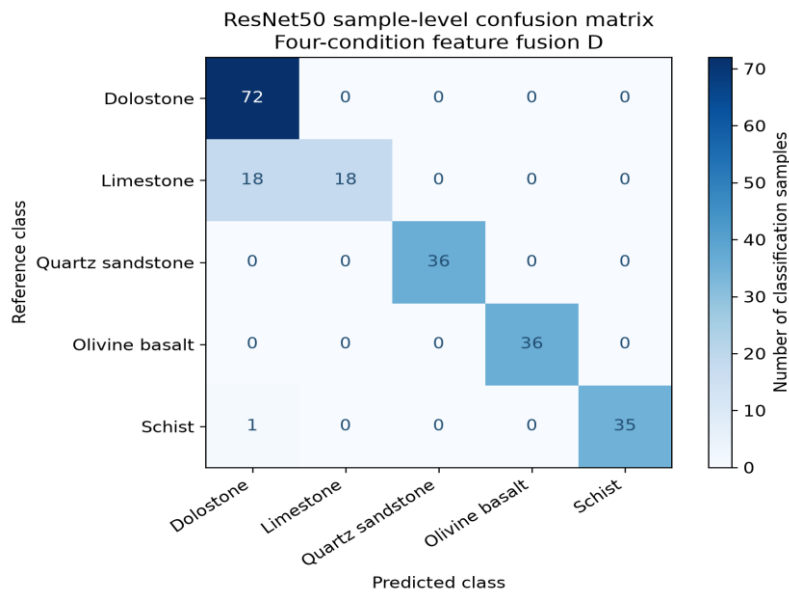


Figure 10 ResNet50 sample-level confusion matrix for four-condition fusion D.

The ResNet50 D report records recall of 1.0000 for dolostone, quartz sandstone, and olivine basalt, 0.5000 for limestone and 0.9722 for schist. Following point-level probability averaging, limestone P00027 was assigned to dolostone. A substantial component of the observed error is therefore associated

with this limestone view. The classification outputs do not establish carbonate composition or identify a mineralogical mechanism underlying the confusion, because no mineral-level attribution result is available.

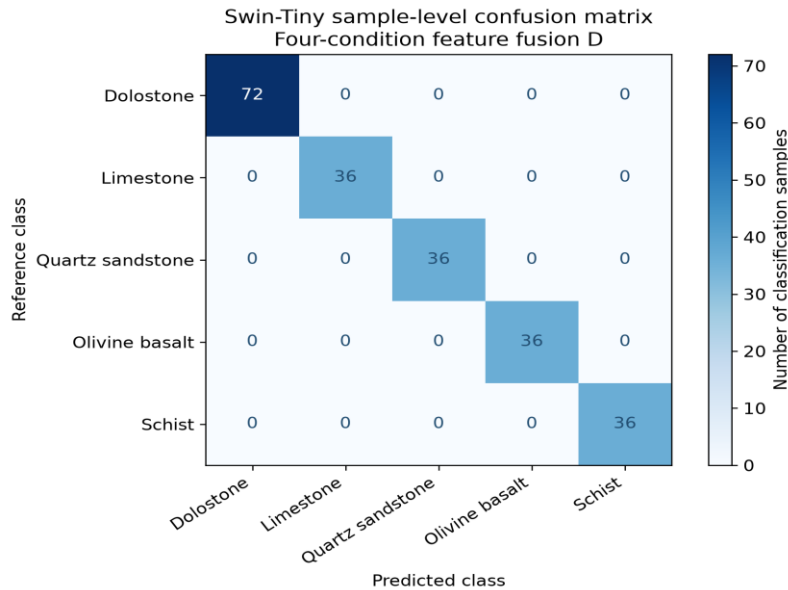


Figure 11 Swin-Tiny sample-level confusion matrix for four-condition fusion D.

3 5 Point-level probability aggregation and resource requirements

Point-level accuracy obtained through probability averaging is reported alongside the sample-level results discussed above. Several configurations achieved 100.00% point-level accuracy despite lower sample-level accuracy, indicating that disagreement among samples does not necessarily alter the aggregated class prediction. For ResNet50 in scenario D, a sample-level accuracy of 91.20% corresponded to five correctly classified test points out of six. With six test points, each point accounts for approximately 16.67 percentage points in accuracy, limiting the resolution of point-level comparisons (Table 6).

Table 6 Recorded probability-averaged accuracy across the six test points

ID	ResNet50 point accuracy %	Swin-Tiny point accuracy %
A1	83.33	100.00
A2	100.00	100.00
A3	100.00	100.00
B1	100.00	83.33
B2	100.00	100.00
B3	100.00	100.00
C1	83.33	66.67
C2	100.00	100.00
C3	66.67	100.00
C4	100.00	100.00
D	83.33	100.00

3 6 Class-specific performance across experimental configurations

Aggregate scores may obscure heterogeneity in class-specific performance. Table 7 and Table 8 report the class F1 values for ResNet50 and Swin-Tiny, respectively, on the 0–1 scale used in the original reports.

Table 7 Recorded class F1 values for ResNet50 on a 0–1 scale

ID	Dolostone	Limestone	Quartz sandstone	Olivine basalt	Schist
A1	0.8152	0.4902	1.0000	0.9333	0.9437
A2	0.9085	0.8477	1.0000	0.9496	0.9315
A3	0.8882	0.7627	1.0000	0.9650	0.9197
B1	0.8251	0.5846	1.0000	0.9931	0.9859
B2	0.9565	0.9790	1.0000	0.7241	0.8642
B3	0.9030	0.7820	1.0000	0.9173	0.9290
C1	0.7826	0.0000	1.0000	1.0000	0.9412
C2	0.9664	0.9254	1.0000	1.0000	1.0000
C3	0.6829	0.3830	1.0000	0.0000	1.0000
C4	0.9290	0.9863	1.0000	0.6667	0.7692
D	0.8834	0.6667	1.0000	1.0000	0.9859

Table 8 Recorded class F1 values for Swin-Tiny on a 0–1 scale

ID	Dolostone	Limestone	Quartz sandstone	Olivine basalt	Schist
A1	0.7978	0.7899	1.0000	0.8923	0.6909
A2	0.9231	0.8099	0.9931	0.8661	0.8931
A3	0.9895	0.9793	1.0000	0.9640	0.9664
B1	0.8252	0.6667	1.0000	1.0000	0.7899
B2	0.9730	0.9565	0.9863	0.6542	0.8023
B3	0.9114	0.7586	1.0000	0.8571	0.8889
C1	0.6729	0.0000	1.0000	1.0000	0.1053
C2	1.0000	1.0000	1.0000	1.0000	1.0000
C3	0.8971	0.8675	1.0000	0.7368	0.8571
C4	0.8675	0.8750	1.0000	0.7368	0.6027
D	1.0000	1.0000	1.0000	1.0000	1.0000

For ResNet50, quartz sandstone has an F1 of 1.0000 in every scenario. This consistency is restricted to the retained quartz sandstone point P00064 and its samples. The same model shows a markedly different response for limestone: F1 is 0.0000 in C1, 0.3830 in C3, 0.9254 in C2 and 0.9863 in C4. These entries show that the single-view class result changed with the supplied imaging condition.

For ResNet50 C1, limestone F1 and recall are both zero. All C1 samples from P00027 were assigned to dolostone, while dolostone recall was 1.0000 and precision was 0.6429. This precision–recall asymmetry indicates that recovery of all reference dolostone samples coincided with false dolostone assignments from other classes. The sample-level accuracy of 81.48% consequently coexists with zero recall for the retained limestone view, a limitation not apparent from overall accuracy alone.

ResNet50 performance for olivine basalt varied substantially among configurations. F1 was 1.0000 in C1, C2 and D, compared with 0.0000 in C3 and 0.6667 in C4. In B2, which pools the PPL and XPL images at 10x, olivine basalt F1 was 0.7241 while limestone F1 was 0.9790. The aggregate B2 score therefore summarizes contrasting class-specific outcomes and does not support an inference of improvement across all retained classes under 10x input conditions.

For Swin-Tiny C1, class F1 was 1.0000 for quartz sandstone and olivine basalt, 0.0000 for limestone and 0.1053 for schist. The macro-F1 of 55.56% summarizes this heterogeneous distribution. Schist recall was 0.0556 despite

precision of 1.0000; the latter value concerns the limited subset of samples predicted as schist and does not imply adequate recovery of the reference class. Joint interpretation of precision and recall is therefore required to characterize the observed classification behavior.

Swin-Tiny C3 exhibited a different class-specific distribution. Limestone recall was 1.0000, precision was 0.7660 and F1 was 0.8675; dolostone recall and precision were 0.8472 and 0.9531, respectively. The recorded prediction counts include limestone assignments for samples from dolostone P00030. Full recovery of the retained limestone samples therefore coincided with false-positive limestone predictions originating from another reference class.

Under A3, Swin-Tiny class F1 ranged from 0.9640 to 1.0000, with residual variation concentrated in olivine basalt and schist relative to the remaining classes. For ResNet50 A3, limestone F1 was 0.7627 and olivine basalt F1 was 0.9650. Polarization fusion was consequently associated with different distributions of class-specific errors across the two models rather than a uniform change in performance. Swin-Tiny D exhibited no recorded sample misclassifications on this partition, whereas ResNet50 D retained the limestone errors presented in Figure 10.

A zero value denotes a recorded classification outcome for the specified configuration and does not establish an absence of diagnostic petrographic characteristics. Correspondingly, a value of one denotes complete agreement with the relevant test samples within this subset and does not establish classification reliability across independent specimens of the same lithology (Table 9)

Table 9 Selected class-specific precision, recall and F1 values.

Model and ID	Class	Precision	Recall	F1
ResNet50 C1	Dolostone	0.6429	1.0000	0.7826
ResNet50 C1	Limestone	0.0000	0.0000	0.0000
ResNet50 C2	Limestone	1.0000	0.8611	0.9254
ResNet50 C3	Olivine basalt	0.0000	0.0000	0.0000
ResNet50 D	Limestone	1.0000	0.5000	0.6667
Swin-Tiny C1	Schist	1.0000	0.0556	0.1053
Swin-Tiny C3	Dolostone	0.9531	0.8472	0.8971
Swin-Tiny C3	Limestone	0.7660	1.0000	0.8675
Swin-Tiny C3	Olivine basalt	1.0000	0.5833	0.7368
Swin-Tiny A3	Schist	0.9351	1.0000	0.9664

.3 7 Classification performance at individual test points

Point-specific records permit examination of heterogeneity that may be obscured by class aggregation. Table 10 presents the number of correctly classified Swin-Tiny samples and the total sample count for each test point, retaining the two dolostone views separately. Each point contributes 72 samples in the A and B configurations and 36 in the C and D configurations. Interpretation of these counts therefore requires explicit consideration of their denominators.

Table 10 Correctly classified Swin-Tiny samples and sample totals by test point

ID	P00018	P00030	P00027	P00064	P00057	P00076
A1	72/72	72/72	47/72	72/72	58/72	38/72
A2	72/72	72/72	49/72	72/72	55/72	71/72
A3	72/72	70/72	71/72	72/72	67/72	72/72
B1	72/72	72/72	36/72	72/72	72/72	47/72
B2	72/72	72/72	66/72	72/72	35/72	71/72
B3	72/72	72/72	44/72	72/72	54/72	72/72
C1	36/36	36/36	0/36	36/36	36/36	2/36
C2	36/36	36/36	36/36	36/36	36/36	36/36
C3	36/36	25/36	36/36	36/36	21/36	36/36
C4	36/36	36/36	28/36	36/36	21/36	22/36
D	36/36	36/36	36/36	36/36	36/36	36/36

In Swin-Tiny C1, all 36 limestone samples were misclassified, and two of the 36 schist samples were correctly classified. Probability averaging assigned both points to dolostone. The two dolostone points and the retained quartz sandstone and olivine basalt points were correctly classified after aggregation. The point accuracy of 66.67% therefore represents four correct and two incorrect view-level assignments rather than comparable intermediate performance across all five classes.

Swin-Tiny C3 demonstrates sample disagreement within correctly classified points. For olivine basalt P00057, 21 of 36 samples were correctly classified, while three were assigned to dolostone and 12 to schist. The aggregated class remained olivine basalt, with a recorded mean predicted-class probability of 0.4362. For dolostone P00030, the corresponding probability was 0.6456 despite 11 of 36 sample predictions being assigned to limestone. The 100.00% point accuracy consequently does not imply error-free classification of the constituent angular sequences.

These are model outputs after aggregation; calibration was not evaluated. They therefore do not constitute empirical probabilities of petrographic label correctness. In C1, the limestone point was incorrectly assigned to dolostone with a recorded mean predicted-class probability of 0.9839. This observation demonstrates that a high model-assigned probability may accompany an incorrect reference-label prediction, without independently quantifying calibration or providing geological verification (Table 11).

Table 11 Selected Swin-Tiny point predictions and mean predicted-class probabilities.

ID and point	Reference class	Predicted class	Correct total /	Mean predicted-class probability
C1 P00027	Limestone	Dolostone	0/36	0.9839
C1 P00076	Schist	Dolostone	2/36	0.8590
C3 P00030	Dolostone	Dolostone	25/36	0.6456
C3 P00057	Olivine basalt	Olivine basalt	21/36	0.4362
D P00027	Limestone	Limestone	36/36	0.9643
D P00057	Olivine basalt	Olivine basalt	36/36	0.9282

For ResNet50 D, half of the limestone samples were correctly classified, while probability averaging yielded a dolostone prediction. In C2, the same retained limestone point had sample accuracy of 86.11% and the correct aggregated label. The four-condition configuration therefore did not improve the limestone point decision relative to C2 for this backbone. This comparison concerns distinct fitted models and their output probabilities; it does not isolate a causal contribution from an individual additional input condition.

4 Discussion

4 1 Implications of feature fusion for classification performance

The Swin-Tiny A3 outcome is consistent with a potential classification benefit from combined polarization features, as investigated by Ma et al. (2025). However, the corresponding ResNet50 comparison did not exhibit a positive difference, limiting the generality of this interpretation across backbones. The available records do not distinguish the contributions of representation learning, branch capacity, optimization duration, or individual test views. Attribution to any single mechanism would therefore require evidence beyond the completed comparison.

Within this subset, the single-condition Swin-Tiny classifier using 4x XPL attained the same recorded endpoint as the four-branch model. C2 therefore matched D on the retained test points without additional imaging conditions.

4 2 Petrographic interpretation and reference-label limitations

PPL and XPL provide different optical observations, and published petrographic fusion work uses that distinction to motivate combined image representations (Ma et al., 2025). Nevertheless, the present classifiers learn associations with the five retained labels. Their outputs do not independently confirm the lithological naming of a source sample. In particular, the limestone-to-dolostone error in ResNet50 D is an error relative to those labels; it is not evidence of a measured compositional transition or a revised petrographic identification.

Similarly, no feature-attribution experiment in the completed record demonstrates which minerals, textures or image regions controlled a prediction. High class accuracy therefore cannot be used to claim that the networks recognized twinning, measured grain size, or quantified mineral abundance. The appropriate interpretation remains at the level evaluated: correspondence between retained labels and predictions for these image configurations.

4 3 Methodological limitations and scope of inference

Class representation is unequal: dolostone contributes 14 points, quartz sandstone, and olivine basalt four each, and the remaining classes eight each. The test partition contains two dolostone points and one point per remaining class. Macro-F1 assigns equal weight to the class scores but does not resolve uncertainty arising from limited view coverage. Point accuracy also counts the two dolostone points separately rather than weighting each class equally.

The fixed five-epoch schedule defines the implemented optimization budget. Common optimizer settings provide procedural consistency without establishing equivalent optimization of the architectural families. Fusion changes parameter count and input volume, while pooling changes classification-sample count. The experiments therefore compare complete configurations rather than isolating causal effects of architecture, polarization, or magnification.

4 4 Configuration-dependent interpretation of experimental comparisons

The A, B and C configuration families address related but distinct comparisons. A1 and A2 evaluate polarization-specific image pools across two source magnification labels; B1 and B2 evaluate magnification-specific pools across two polarizations. C1–C4 restrict both variables simultaneously. Differences in sample composition prevent direct transfer of a ranking from the pooled comparisons to every individual condition. The opposing PPL rankings at 4x and 10x provide an observed example of this configuration dependence.

The interpretation of magnification-associated differences depends on the comparison structure. B2 exceeded B1 in macro-F1 for both backbones, whereas the highest-scoring individual condition was C2 at 4x XPL. The former

comparison pools polarization channels within magnification, while the latter concerns a specified polarization–magnification combination. Consequently, these results do not support a general physical benefit from increased magnification, which was not isolated experimentally.

Fusion outcomes similarly require specification of the comparator. A3 is evaluated relative to the PPL and XPL pools, whereas D is evaluated relative to the four individual imaging conditions. A positive A3 difference for Swin-Tiny does not establish that D should exceed C2, given differences in fusion structure, sample composition, and classifier inputs. Retaining the single-condition comparator is therefore necessary for interpreting the absence of a D score above C2.

5 Conclusions

Within the completed five-class comparison, performance varied with the combination of backbone and imaging condition. Pooled XPL scored above pooled PPL for both models, but PPL–XPL fusion improved on the XPL score only for Swin-Tiny. Combining magnification labels produced smaller recorded differences. Four-condition fusion did not exceed the highest single-condition macro-F1 for either backbone; the Swin-Tiny 4x XPL and full-fusion runs both recorded 100.00% on the retained test set.

These observations support a conditional interpretation of fusion in this subset. Additional inputs were associated with a gain in some configurations and no gain or a decrease in others. The six-point test set, point-based rather than specimen-based separation and retained source constrain the conclusion to the completed experiments.

References

- Boggs, S., Jr. (2009). Petrology of sedimentary rocks (2nd ed.). Cambridge, England: Cambridge University Press.
- Daxberger, H., and Kennedy, K. (2026). Virtual Petrography [Online educational resource]. University of Toronto Scarborough. <https://planetearth.utsc.utoronto.ca/VirtualMic/> (accessed 10 October 2026). License: Creative Commons Attribution 4.0 International (CC BY 4.0), <https://creativecommons.org/licenses/by/4.0/>.
- Deer, W. A., Howie, R. A., & Zussman, J. (2013). An introduction to the rock-forming minerals (3rd ed.). London, England: Mineralogical Society of Great Britain and Ireland.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778. <https://doi.org/10.1109/CVPR.2016.90>.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using shifted windows. Proceedings of the IEEE/CVF International Conference on Computer Vision, 10012–10022. <https://doi.org/10.1109/ICCV48922.2021.00986>.
- Ma, W., Han, T., Xu, Z., and Lin, P. (2025). Feature fusion of single and orthogonal polarized rock images for intelligent lithology identification. AI in Civil Engineering, 4, 5. <https://doi.org/10.1007/s43503-025-00049-7>.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Philpotts, A. R., & Ague, J. J. (2009). Principles of igneous and metamorphic petrology (2nd ed.). Cambridge, England: Cambridge University Press.
- scikit-learn Developers. (2026). Metrics and scoring: quantifying the quality of predictions. scikit-learn documentation. https://scikit-learn.org/stable/modules/model_evaluation.html (accessed 10 October 2026).
- Torchvision Contributors. (2026-a). resnet50. Torchvision documentation. <https://docs.pytorch.org/vision/stable/models/generated/torchvision.models.resnet50.html> (accessed October 2026).
- Torchvision Contributors. (2026-b). swin_t. Torchvision documentation. https://docs.pytorch.org/vision/stable/models/generated/torchvision.models.swin_t.html (accessed 10 October 2026).
- Tucker, M. E., & Wright, V. P. (1990). Carbonate sedimentology. Oxford, England: Blackwell Scientific Publications.