

Intelligent Computational Systems for Scientific Problems: Principles, Methodologies, and Intricate Data Classification

Lukas Steiner

Department of Applied Sciences, Institute of Technology and Engineering, Vienna, Austria

Anna Gruber

Department of Applied Sciences, Institute of Technology and Engineering, Vienna, Austria

Received: 21 Aug 2026 | Received Revised Version: 09 Sep 2026 | Accepted: 22 Sep 2026 | Published: 02 Oct 2026

Volume 08 Issue 10 2026 | Crossref DOI: 10.37547/tajas/Volume08Issue10-01

Abstract

The increasing complexity, dimensionality, and dynamical nature of scientific and engineering problems have created a growing demand for computational systems capable of learning from data, representing nonlinear relationships, solving optimization problems, and supporting decision-making under uncertainty. Intelligent computational systems integrate concepts from machine learning, neural computation, optimization, distributed computing, signal processing, control theory, and scientific computing to address problems that are difficult to solve through conventional analytical techniques alone. This review examines the theoretical foundations, methodological development, and application-oriented characteristics of such systems, with particular emphasis on neural architectures, deep learning, recurrent computation, intelligent optimization, adaptive control, distributed computation, and data classification. The analysis synthesizes the supplied literature to establish a coherent framework connecting foundational learning models with contemporary scientific and engineering applications.

The reviewed literature demonstrates that intelligent computation has evolved from relatively compact neural models toward multilayer, recurrent, distributed, and hybrid computational architectures. Early developments in deep learning established mechanisms for learning hierarchical representations, while recurrent and complex-valued neural models expanded computational learning toward temporal, dynamic, and mathematically structured problems (Hochreiter and Schmidhuber, 1997; Hinton et al., 2006; Bengio, 2009; LeCun et al., 2015). Subsequent work has demonstrated that intelligent algorithms can also be integrated with optimization and control mechanisms to solve time-varying systems, robotic motion problems, portfolio-selection problems, engineering prediction tasks, and infrastructure-management problems. In parallel, cloud and volunteer distributed computing have demonstrated how computational resources can be reorganized to address scientific workloads that exceed the practical capabilities of individual systems (Armbrust et al., 2010; Beberg et al., 2009).

A central finding of this synthesis is that intelligent computational systems should not be understood solely as predictive models. Their broader value lies in their capacity to create computational representations of complex relationships and to connect perception, prediction, optimization, control, and decision processes. Neural networks can provide nonlinear approximation and representation learning; recurrent architectures can model temporal dependencies; optimization algorithms can search complex solution spaces; and distributed computational environments can provide the resources required for large-scale processing. Applications in wind-energy estimation, grounding-system design, robotic manipulation, nonlinear landslide prediction, visual tracking, financial optimization, and dynamic equation solving

illustrate the breadth of this paradigm (Deng et al., 2020; Hu et al., 2019a, 2021; Huang et al., 2022; Khan et al., 2020a, 2022a).

The literature also reveals important limitations. Computational intelligence does not automatically guarantee interpretability, stability, generalization, or computational efficiency. Increasing model complexity may introduce substantial resource requirements, while optimization-driven approaches can remain sensitive to initialization, parameter selection, problem formulation, and noise. Deep and language-oriented systems additionally raise concerns regarding computational scale and responsible deployment (Bender et al., 2021). Consequently, future intelligent scientific systems are likely to depend on hybrid architectures that combine learned representations with mathematical constraints, optimization procedures, domain knowledge, distributed infrastructure, and stability-aware control. Such integration can provide a more systematic foundation for reliable computational intelligence across scientific and engineering domains.

Keywords: Computational intelligence; Neural networks; Deep learning; Machine learning; Intelligent optimization; Recurrent neural networks; Scientific computing; Data classification; Nonlinear systems; Engineering applications

© 2026 Lukas Steiner, Anna Gruber, This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). The authors retain copyright and allow others to share, adapt, or redistribute the work with proper attribution.

Cite This Article: Steiner, L., & Gruber, A. (2026). Intelligent Computational Systems for Scientific Problems: Principles, Methodologies, and Intricate Data Classification. *The American Journal of Applied Sciences*, 8(10), 01–17. <https://doi.org/10.37547/tajas/Volume08Issue10-01>

1. Introduction

Scientific and engineering research increasingly involves computational problems characterized by nonlinear relationships, high-dimensional observations, temporal dependencies, uncertainty, and large solution spaces. Conventional analytical methods remain fundamental to scientific investigation, but many practical problems cannot be represented adequately through closed-form mathematical expressions or solved efficiently using deterministic procedures alone. The increasing availability of computational resources and data has therefore encouraged the development of intelligent computational systems capable of learning, adapting, predicting, optimizing, and classifying complex information.

The concept of intelligent computation encompasses a broad set of methodologies rather than a single algorithmic family. Neural networks provide computational mechanisms for approximating nonlinear mappings and extracting representations from data. Deep learning extends these principles through multilayer architectures capable of hierarchical representation learning. Recurrent models introduce temporal dependencies and sequential processing, while optimization algorithms search for solutions in complex and often non-convex spaces. Distributed and cloud computing provide the infrastructure required to execute

computationally intensive workloads. These components increasingly interact in scientific applications, creating systems in which learning, optimization, computation, and decision-making are not isolated processes.

The theoretical foundations of this field can be traced through the development of increasingly sophisticated neural architectures. Hochreiter and Schmidhuber introduced long short-term memory as a mechanism for handling temporal dependencies in recurrent neural computation, addressing limitations associated with learning long-range relationships (Hochreiter and Schmidhuber, 1997). Hinton et al. demonstrated a fast learning algorithm for deep belief networks, contributing to the development of multilayer learning architectures (Hinton et al., 2006). Bengio subsequently examined the learning of deep architectures and the challenges associated with training hierarchical representations, establishing an important theoretical foundation for modern deep learning (Bengio, 2009). The difficulty of training deep feedforward neural networks was further examined by Bengio and Glorot, highlighting the importance of appropriate initialization and optimization mechanisms for deep architectures (Bengio and Glorot, 2010).

The development of deep learning subsequently transformed intelligent computation from manually engineered feature systems toward representation-

learning architectures. LeCun et al. characterized deep learning as a major approach for learning multiple levels of representation, while Schmidhuber provided a broad overview of deep learning within neural networks and its historical development (LeCun et al., 2015; Schmidhuber, 2015). These developments have implications far beyond conventional pattern recognition. A computational model that learns representations can potentially support classification, prediction, anomaly identification, signal interpretation, control, and optimization within a common framework.

At the same time, intelligent computational systems have expanded beyond traditional neural networks. The supplied literature includes zeroing neural networks, complex-valued recurrent neural networks, fuzzy-control strategies, neutrosophic approaches, beetle antennae search, and hybrid metaheuristic systems. These approaches demonstrate that computational intelligence can be formulated around specific mathematical structures and engineering requirements rather than being limited to general-purpose prediction. For example, Ding et al. investigated recurrent neural models for complex-valued equations, while Jia et al. integrated adaptive fuzzy control with zeroing neural networks for time-varying quadratic programming problems (Ding et al., 2018; Jia et al., 2021). Such studies demonstrate a movement toward task-specific intelligent computation.

The engineering relevance of these developments is particularly evident in applications involving physical systems. Wind-energy capture requires accurate estimation and control of variable operating conditions. Deng et al. proposed an effective wind-speed estimation approach integrated with extended optimal torque control for maximum wind-energy capture (Deng et al., 2020). Similarly, grounding-system design involves complex electrical and physical relationships that may benefit from optimization and computational modeling (Hu et al., 2019a, 2021). Landslide prediction introduces another class of nonlinear engineering problems in which prediction models must account for complex relationships between variables and observed displacement. Huang et al. developed a nonlinear prediction model based on MIC-SVR for antislides pile-top displacement in Jurassic landslides (Huang et al., 2022).

Robotics provides another important domain in which intelligent computational methods combine learning,

optimization, and control. Reconfigurable micro-positioning stages, constant-force mechanisms, polishing end-effectors, redundant manipulators, and biped robots all require computational procedures capable of handling constraints and nonlinear behavior (Ding et al., 2019a, 2021, 2022; Khan et al., 2021, 2022b). These applications illustrate that intelligent computation becomes particularly valuable when a problem requires continuous adaptation rather than a single static prediction.

Financial systems further demonstrate the transferability of computational intelligence. Katsikis and colleagues examined intelligent algorithms and beetle antennae search approaches for portfolio-related optimization problems, while Khan et al. investigated BAS-based methods for optimization, fraud detection, and multi-portfolio selection (Katsikis and Mourtas, 2021; Katsikis et al., 2021, 2023; Khan et al., 2020a, 2022c, 2022d). Although the underlying application differs from engineering control, the computational requirement is comparable: a complex search space must be explored under multiple constraints.

Another dimension of intelligent scientific computation concerns infrastructure. Scientific problems increasingly require computational resources that cannot be supplied efficiently by individual machines. Volunteer distributed computing demonstrated through Folding@home illustrates how large numbers of distributed resources can collectively support scientific computation (Beberg et al., 2009). Cloud computing similarly provides scalable computational infrastructure and changes the relationship between scientific workloads and available resources (Armbrust et al., 2010). Optical computing has also been discussed as a potential response to future supercomputing requirements, reflecting the continuing search for more efficient computational substrates (Caulfield and Dolev, 2010).

The relevance of intelligent computational systems therefore arises from the interaction between algorithmic intelligence and computational infrastructure. A sophisticated learning algorithm is of limited practical value if the computational environment cannot execute it efficiently. Conversely, large computational resources do not automatically generate meaningful scientific insight without suitable models, representations, and validation procedures.

A further issue concerns the interpretation of complex learned representations. Ethayarajh examined the geometry of contextualized word representations and compared representation structures associated with BERT, ELMo, and GPT-2, illustrating the importance of understanding what learned representations encode (Ethayarajh, 2019). Although this work focuses on language representations, the underlying question is broader: intelligent computational systems should be evaluated not merely by output accuracy but also by the properties and reliability of their internal representations.

The problem addressed by this review is therefore the fragmentation of intelligent computational approaches across different scientific and engineering domains. Neural networks, deep learning, optimization, recurrent computation, intelligent control, and distributed computing are frequently studied independently, even though practical intelligent systems increasingly combine these components. A unified perspective is needed to understand their theoretical relationships, application mechanisms, strengths, limitations, and potential integration.

The objective of this paper is to synthesize the supplied literature into a structured framework for intelligent computational systems applied to scientific problems. The review specifically examines the progression from foundational neural computation to deep and recurrent learning, task-oriented computational intelligence, intelligent optimization, scientific prediction, robotics, infrastructure applications, and distributed computation. It also analyzes the methodological trade-offs associated with model complexity, interpretability, computational cost, stability, generalization, and domain integration.

The significance of this review lies in its interdisciplinary perspective. Rather than treating each algorithm as an isolated technique, the paper conceptualizes intelligent computation as a layered system. At the lowest level, computational infrastructure provides processing capability. Above this layer, mathematical and algorithmic mechanisms perform optimization or learning. Representation and prediction mechanisms transform raw observations into computationally useful information. Finally, application-specific control or decision mechanisms translate computational outputs into scientific or engineering actions.

2. Literature Review

2.1 Foundations of neural and deep computation

The development of modern intelligent computation is strongly associated with neural-network architectures capable of learning nonlinear mappings. Bengio's work on deep architectures established that learning multiple levels of representation can provide a mechanism for transforming complex input structures into increasingly abstract representations (Bengio, 2009). However, deep architectures introduce optimization challenges. Bengio and Glorot analyzed the difficulty of training deep feedforward neural networks, demonstrating that successful learning depends substantially on the properties of the training process and parameter initialization (Bengio and Glorot, 2010).

Recurrent computation addressed a different challenge: temporal dependency. Hochreiter and Schmidhuber's long short-term memory architecture was designed to facilitate learning over extended temporal intervals, providing a mechanism for retaining relevant information while limiting the effects of vanishing gradients (Hochreiter and Schmidhuber, 1997). This principle became important for scientific and engineering systems where observations evolve over time rather than existing as independent samples.

Hinton et al. contributed to the development of deep belief networks through an efficient learning algorithm, reinforcing the broader transition toward multilayer representation learning (Hinton et al., 2006). Later, LeCun et al. presented deep learning as a general paradigm in which multiple processing layers learn representations of increasing abstraction (LeCun et al., 2015). Schmidhuber's review similarly placed deep learning within the longer historical development of neural networks and identified its broad methodological foundations (Schmidhuber, 2015).

2.2 Scientific representation and complex data

The supplied literature demonstrates that intelligent systems are increasingly required to operate on complex data rather than simple scalar observations. Ding et al. developed a modified recurrent neural network for complex-valued linear equations, while another study examined complex-valued Sylvester equations using an improved recurrent neural network model (Ding et al., 2018, 2019b). These studies are significant because they demonstrate how neural computation can be adapted to mathematical structures in which real-valued assumptions are insufficient.

Complex-valued computation is particularly relevant to engineering fields involving frequency, phase, and electromagnetic information. The RF studies by Jin and colleagues illustrate computational problems in circuit and communication-system design, including current-mode ring oscillators and current-reuse current mirrors (Jin and Yu, 2013; Jin et al., 2017a). Although these studies are not identical to modern deep-learning systems, they demonstrate the broader role of computational modeling in designing systems characterized by complex physical relationships.

Visual data introduce another form of complexity. Hu et al. examined online visual tracking using weighted multiple-instance learning and neutrosophic similarity-based objectness estimation (Hu et al., 2019b). The approach illustrates the combination of data representation, uncertainty-related reasoning, and object tracking. This provides an example of intelligent computation functioning as a pipeline rather than a single predictive model.

Ethayarajh's analysis of contextualized word representations provides an additional perspective on representation geometry. The comparison of BERT, ELMo, and GPT-2 embeddings illustrates that learned representations can contain structured relationships that differ across models and contextual conditions (Ethayarajh, 2019). This is relevant to scientific computation because classification performance alone does not fully describe the characteristics of an intelligent model.

2.3 Intelligent optimization and metaheuristics

Optimization constitutes a major component of computational intelligence. Many scientific problems require identification of an acceptable solution among numerous possible configurations, frequently under nonlinear or conflicting constraints. The literature supplied includes beetle antennae search, BAS-Adam, nonlinear activated BAS, and dual BAS approaches.

Khan et al. developed BAS-Adam by combining beetle antennae search with Adam-based optimization, demonstrating the possibility of integrating metaheuristic exploration with gradient-based adaptive optimization (Khan et al., 2020a). Other work applied beetle behavior to computational models for operational management, illustrating how biological metaphors can be converted into search mechanisms (Khan et al., 2020b).

Portfolio optimization studies further demonstrate the role of intelligent search. Katsikis et al. applied beetle antennae search to time-varying mean-variance portfolio selection under transaction costs and cardinality constraints (Katsikis et al., 2021). Khan et al. developed a nonlinear activated BAS approach for non-convex tax-aware portfolio optimization (Khan et al., 2022a). Such problems demonstrate why intelligent optimization is relevant: conventional optimization assumptions may become restrictive when the objective function is non-convex, dynamic, discontinuous, or subject to multiple practical constraints.

2.4 Intelligent computation in control and robotics

Robotic systems require simultaneous consideration of position, velocity, constraints, dynamics, uncertainty, and task objectives. The supplied literature demonstrates several approaches to computational intelligence in robotics. Ding et al. studied reconfigurable planar micro-positioning stages based on functional modules, while another study examined constant-force compliant mechanisms experimentally (Ding et al., 2019a; Ding et al., 2022). A spatial constant-force end-effector was developed for polishing and deburring operations, showing how computational and mechanical design principles can be integrated for industrial tasks (Ding et al., 2021).

Redundant robot manipulators create an especially important optimization problem because multiple joint configurations can correspond to a single end-effector objective. Khan et al. examined tracking control of redundant manipulators under active remote-center-of-motion constraints using an RNN-based metaheuristic approach (Khan et al., 2021). Jin et al. investigated cooperative motion generation in distributed networks of redundant robot manipulators under noise, emphasizing the importance of decentralized and robust computational mechanisms (Jin et al., 2017b).

Biped robots represent another difficult control problem because locomotion involves dynamic stability and multiple interacting constraints. Khan et al. developed a dual beetle antennae search system for optimal planning and robust control of five-link biped robots (Khan et al., 2022b). Collectively, these studies demonstrate that intelligent computation can function as an optimization and control mechanism rather than merely a classification tool.

2.5 Prediction and infrastructure applications

Scientific prediction is another major application. Deng et al. developed a wind-speed estimation approach connected to optimal torque control for maximum wind-energy capture (Deng et al., 2020). This represents a closed-loop computational perspective in which estimation is valuable because it contributes directly to control.

Electrical infrastructure provides additional examples. Hu et al. proposed a new optimization design for grounding grids, while another study considered substation grounding through electrolytic cathodic protection and transfer corrosion current (Hu et al., 2019a, 2021). These applications demonstrate that computational intelligence can support infrastructure design by considering interacting electrical and physical parameters.

Huang et al. investigated nonlinear prediction of antislide pile-top displacement using MIC-SVR for Jurassic landslides (Huang et al., 2022). The importance of such prediction lies in the relationship between computational output and engineering risk management. A prediction model becomes scientifically useful when it can represent nonlinear relationships sufficiently well to inform monitoring and intervention.

2.6 Stability and mathematical problem solving

Not all intelligent computational systems are designed for empirical prediction. Some are constructed to solve mathematical problems dynamically. Ding et al. studied stability criteria for static recurrent neural networks with time-varying delays, illustrating the importance of mathematical stability in neural computation (Ding et al., 2014). Jia et al. developed an adaptive fuzzy-control strategy integrated with a zeroing neural network for solving time-variant quadratic programming problems (Jia et al., 2021).

Katsikis et al. developed a zeroing neural network based on neutrosophic logic for calculating minimal-norm least-squares solutions to time-varying linear systems (Katsikis et al., 2023). Han et al. examined an accelerated double-integral zeroing neural network with resistance to linear noise for dynamic Sylvester equation solving and its application to chaotic-system control (Han et al., 2023).

These studies establish an important theoretical distinction. Neural computation can be used not only to learn from historical data but also to construct dynamical

computational processes whose evolution approximates solutions to mathematical problems. This expands the meaning of "intelligent system" from statistical prediction toward computational problem solving.

2.7 Distributed scientific computation

The growth of intelligent computational methods has occurred alongside advances in computational infrastructure. Beberg et al. described lessons from eight years of volunteer distributed computing through Folding@home, demonstrating how distributed computational resources can contribute to scientific workloads (Beberg et al., 2009). Armbrust et al. analyzed cloud computing as a model for scalable computational infrastructure, emphasizing the ability to provision computing resources according to workload requirements (Armbrust et al., 2010).

Caulfield and Dolev discussed optical approaches to future supercomputing, indicating that continued growth in computational requirements may require new hardware paradigms (Caulfield and Dolev, 2010). These infrastructure studies are important because model development and computational deployment are inseparable at scale.

2.8 Research gap

The reviewed literature reveals three principal gaps. First, intelligent computation is frequently investigated at the level of individual algorithms, while fewer conceptual frameworks connect learning, optimization, control, classification, and infrastructure. Second, application studies often demonstrate task-specific effectiveness without establishing a general framework for comparing computational requirements across domains. Third, increasing model complexity creates a need for simultaneous consideration of predictive capability, stability, interpretability, computational cost, and deployment requirements.

The present synthesis addresses these gaps by proposing a layered view of intelligent computational systems in which representation, inference, optimization, control, and infrastructure are treated as interconnected components.

3. Methodology

3.1 Conceptual architecture of intelligent computational systems

An intelligent computational system can be represented conceptually as a sequence of interacting layers:

Data acquisition → preprocessing → representation → learning/inference → optimization → decision/control → computational deployment → feedback

This representation should not be interpreted as a rigid pipeline. In practical systems, feedback can return from the control layer to the data-acquisition layer, while optimization may influence representation or model parameters.

The first layer concerns data acquisition. Scientific systems may generate sensor observations, numerical simulations, images, time series, financial records, or communication signals. The data characteristics determine the appropriate computational representation. For example, sequential data may require recurrent architectures, while complex-valued measurements may require specialized mathematical treatment.

The second layer concerns preprocessing and representation. Deep-learning systems attempt to learn useful representations automatically, reducing reliance on manually specified features (Bengio, 2009; LeCun et al., 2015). However, representation quality depends on data distribution, architecture, training conditions, and objective functions. Ethayarajh's analysis of contextualized representations demonstrates that learned embeddings possess geometric properties that differ among architectures and contexts (Ethayarajh, 2019).

The third layer concerns inference. A model may classify an input, estimate a continuous quantity, detect an anomaly, or predict future system behavior. In scientific engineering, prediction may be only an intermediate step. Wind-speed estimation, for example, becomes operationally meaningful when it supports optimal torque control (Deng et al., 2020).

The fourth layer is optimization. Intelligent optimization methods search for parameters, configurations, or actions that improve an objective function. Metaheuristic algorithms are particularly useful when the optimization landscape is difficult to characterize analytically. BAS-based studies illustrate this principle across management, portfolio selection, fraud detection, and robotic control (Khan et al., 2020a, 2020b, 2022a, 2022c).

The fifth layer concerns decision and control. Here computational outputs are translated into system actions. Robotic systems provide clear examples because predictions and optimization outputs must ultimately generate physically realizable movements (Khan et al., 2021, 2022b).

The final layer concerns deployment infrastructure. Cloud computing and distributed systems allow computational tasks to scale beyond individual machines (Armbrust et al., 2010). Scientific distributed computing demonstrates that computation itself can become a collaborative resource (Beberg et al., 2009).

3.2 Neural representation and nonlinear approximation

Neural computation is particularly valuable for nonlinear scientific relationships. A conventional linear model assumes that output can be represented through a linear combination of inputs. Scientific systems frequently violate this assumption. Physical processes may involve interactions, thresholds, feedback, saturation, delays, and nonlinear coupling.

Deep architectures provide multiple layers through which nonlinear transformations can be composed. Conceptually, a deep model may be expressed as:

$$\begin{aligned} &[\\ &h_1 = f_1(W_1x + b_1) \\ &] \\ &[\\ &h_2 = f_2(W_2h_1 + b_2) \\ &] \\ &[\\ &\hat{y} = f_3(W_3h_2 + b_3) \\ &] \end{aligned}$$

where (x) represents input data, (h_i) are intermediate representations, (W_i) are learned parameters, (b_i) are biases, and (f_i) are nonlinear transformations.

The importance of depth lies not simply in adding layers but in enabling hierarchical transformations. Bengio's work emphasizes representation learning as a central component of deep architectures (Bengio, 2009). However, the training difficulty identified by Bengio and Glorot demonstrates that increasing depth introduces optimization challenges (Bengio and Glorot, 2010).

The practical implication is that model capacity should correspond to problem complexity. A model that is too simple may fail to capture nonlinear relationships, while excessive complexity can increase computational requirements and potentially impair generalization. Thus, intelligent computation involves a balance between representational capacity and computational robustness.

3.3 Temporal intelligence and recurrent systems

Many scientific processes evolve over time. Weather-related measurements, network traffic, robotic trajectories, electrical signals, and dynamic physical systems cannot always be modeled adequately using independent observations.

Recurrent architectures introduce state:

$$\begin{bmatrix} h_t = f(W_{xx}x_t + W_{hh}h_{t-1} + b) \end{bmatrix}$$

where (x_t) is the current input and (h_{t-1}) represents information retained from previous time steps.

LSTM architecture was specifically developed to facilitate long-range temporal learning (Hochreiter and Schmidhuber, 1997). The conceptual significance of recurrent computation extends beyond sequence prediction. Ding et al. demonstrated recurrent neural computation for complex-valued equations and time-varying Sylvester equations (Ding et al., 2018, 2019b).

The value of recurrence in scientific systems therefore lies in its ability to represent evolving state. Nevertheless, recurrence also introduces additional complexity. Stability, initialization, sequence length, noise, and computational cost become important considerations. Ding et al.'s investigation of stability criteria for recurrent networks with time-varying delays reinforces the importance of analyzing mathematical behavior rather than relying exclusively on empirical performance (Ding et al., 2014).

3.4 Complex-valued computational intelligence

Scientific and engineering data frequently contain amplitude and phase information. In such cases, forcing data into purely real-valued representations may discard structural information. Complex-valued neural computation provides a framework in which real and imaginary components can be processed jointly.

Ding et al. proposed a recurrent neural network model with a modified nonlinear activation function for complex-valued linear equations and later examined an improved complex-valued recurrent neural network for time-varying complex-valued Sylvester equations (Ding et al., 2018, 2019b).

These approaches suggest a broader principle: intelligent computational architectures should respect the mathematical structure of the problem. Rather than transforming every scientific problem into a generic data-classification task, computational models can incorporate domain-specific structures directly.

3.5 Intelligent classification of intricate data

Classification can be represented as the mapping:

$$\begin{bmatrix} \rightarrow C \end{bmatrix}$$

where (X) represents an input space and (C) represents a set of classes.

For simple datasets, classification may be straightforward. Scientific datasets often contain overlapping distributions, noise, missing values, nonlinear boundaries, temporal dependencies, and high-dimensional representations. Consequently, classification should be understood as a representation problem as much as a decision problem.

Deep architectures attempt to construct progressively useful representations. Contextualized language representations demonstrate how representation geometry can vary across models (Ethayarajh, 2019). Visual tracking research similarly illustrates the need to integrate object representation, similarity estimation, and uncertainty-aware mechanisms (Hu et al., 2019b).

The term "intricate data classification" therefore encompasses more than assigning labels. It includes extracting meaningful structure from observations where class boundaries may depend on context, temporal information, nonlinear relationships, or uncertainty.

3.6 Optimization as an intelligence mechanism

Prediction and classification answer questions about system behavior, whereas optimization identifies preferred configurations or actions. These functions can be coupled.

For an optimization problem,

$$\begin{bmatrix} \min_x F(x) \end{bmatrix}$$

subject to

$$\begin{bmatrix} g_i(x) \leq 0, \quad i=1, \dots, m \end{bmatrix}$$

the computational objective is to identify a feasible (x) that minimizes the objective function.

Real scientific problems may involve multiple objectives, nonlinear constraints, dynamic parameters, or incomplete information. This explains the interest in metaheuristic techniques such as BAS.

BAS-inspired approaches provide search mechanisms based on directional exploration. Khan et al. extended this principle through BAS-Adam, combining metaheuristic exploration with an Adam-based optimization mechanism (Khan et al., 2020a). This hybridization reflects an important direction in computational intelligence: different algorithms may possess complementary strengths.

Portfolio studies provide a useful example. Transaction costs and cardinality constraints complicate the optimization landscape, while tax-aware portfolio selection introduces additional nonlinear considerations (Katsikis et al., 2021; Khan et al., 2022a). The resulting problem cannot be reduced simply to prediction. It requires systematic exploration of feasible alternatives.

3.7 Hybrid intelligence for robotic control

Robotic systems reveal why intelligent computation often needs multiple components. Consider a redundant manipulator. The system may need to:

1. estimate the current state;
2. identify a desired trajectory;
3. satisfy kinematic constraints;
4. avoid undesirable configurations;
5. optimize joint motion;
6. execute the resulting control action.

Khan et al. combined recurrent neural computation with metaheuristic optimization for redundant-manipulator tracking under active remote-center-of-motion constraints (Khan et al., 2021). This demonstrates how learning and optimization can complement one another.

Similarly, cooperative motion generation among distributed redundant manipulators introduces communication and noise considerations (Jin et al., 2017b). The problem therefore becomes simultaneously computational, control-oriented, and distributed.

Mechanical design studies provide a complementary perspective. Reconfigurable micro-positioning stages and constant-force mechanisms demonstrate how intelligent computational principles can support the design and verification of physical mechanisms (Ding et al., 2019a, 2022). A constant-force end-effector designed for polishing and deburring similarly shows the importance of integrating computational requirements with mechanical functionality (Ding et al., 2021).

3.8 Intelligent systems for energy and electrical infrastructure

Energy systems require continuous estimation and control under changing operating conditions. Deng et al. investigated effective wind-speed estimation within an extended optimal torque-control strategy for maximum wind-energy capture (Deng et al., 2020).

The computational architecture can be understood as:

$$\begin{bmatrix} \text{Measurements} \rightarrow \text{Wind-speed estimation} \rightarrow \text{Optimal torque calculation} \rightarrow \text{Generator control} \end{bmatrix}$$

The important principle is that prediction is connected to action. An estimated variable has practical value because it improves the subsequent control process.

Electrical grounding presents another computational challenge. Hu et al. developed an optimization design for grounding grids, while later work considered substation grounding with electrolytic cathodic protection and transfer corrosion current (Hu et al., 2019a, 2021). These applications illustrate the need to integrate computational optimization with physical engineering constraints.

3.9 Intelligent prediction in geotechnical systems

Landslide monitoring demonstrates the role of nonlinear prediction in infrastructure safety. Huang et al. proposed a nonlinear prediction model of antislide pile-top displacement using MIC-SVR for Jurassic landslides (Huang et al., 2022).

The general computational problem can be conceptualized as:

$$\begin{bmatrix} \hat{y}_{t+\Delta} = f(x_t, x_{t-1}, \dots, x_{t-k}) \end{bmatrix}$$

where (x_t) represents observed system variables and $(\hat{y}_{t+\Delta})$ represents predicted future displacement.

The practical challenge is that geological systems may contain nonlinear interactions and changing conditions. A prediction system must therefore be evaluated not only according to average error but also according to robustness under changing environmental conditions.

3.10 Zeroing neural networks and dynamic mathematical computation

A distinctive branch of intelligent computation treats neural networks as dynamic solvers. Instead of learning a mapping from a large dataset, the network evolves toward a mathematical solution.

Suppose the desired solution satisfies:

$$\begin{bmatrix} F(x(t), t) = 0 \end{bmatrix}$$

A zeroing neural network can construct an error function and evolve the computational state toward zero error. Jia et al. integrated adaptive fuzzy control with a zeroing neural network for time-varying quadratic programming (Jia et al., 2021). Katsikis et al. applied a zeroing neural network based on neutrosophic logic to minimal-norm least-squares problems involving time-varying linear systems (Katsikis et al., 2023).

Han et al. further investigated accelerated double-integral zeroing neural networks with resistance to linear noise for dynamic Sylvester equation solving and chaotic-system control (Han et al., 2023).

These approaches expand computational intelligence beyond statistical learning. The computational system

itself becomes a dynamical mechanism for obtaining a solution.

3.11 Uncertainty-aware and neutrosophic computation

Real scientific observations are frequently incomplete or uncertain. Classical binary logic assumes that propositions are either true or false, whereas scientific data may involve ambiguity, inconsistency, or incomplete information.

Neutrosophic approaches attempt to represent such conditions explicitly. Hu et al. used neutrosophic similarity-based objectness estimation in online visual tracking (Hu et al., 2019b), while Katsikis et al. incorporated neutrosophic logic into a zeroing neural network for time-varying linear systems (Katsikis et al., 2023).

The implication is that intelligent systems may benefit from representing uncertainty within the computational architecture rather than treating uncertainty solely as noise. However, the increased representational flexibility can also increase methodological complexity and complicate interpretation.

3.12 Intelligent algorithms for financial decision systems

Financial optimization demonstrates how computational intelligence operates under conflicting objectives and constraints. Portfolio selection requires balancing expected return, risk, transaction costs, cardinality restrictions, and potentially tax considerations.

Katsikis and Mourtas examined portfolio insurance and intelligent algorithms, while Katsikis et al. applied BAS to time-varying mean-variance portfolio selection with transaction costs and cardinality constraints (Katsikis and Mourtas, 2021; Katsikis et al., 2021). Khan et al. extended BAS toward tax-aware portfolio optimization, demonstrating its application to non-convex decision spaces (Khan et al., 2022a).

The same computational principles can be applied to fraud detection. Khan et al. developed a BAS-based machine-learning approach for detecting fraud in publicly traded US firms (Khan et al., 2022c). This demonstrates a transition from optimization toward classification, showing that a single intelligent

computational family can support different functional objectives.

Khan et al. also investigated distributed confidential multi-portfolio selection using bio-inspired machine learning (Khan et al., 2022d). The study illustrates the convergence of optimization, machine learning, privacy considerations, and distributed computation.

3.13 Distributed and cloud-enabled intelligence

The computational demand of intelligent systems increases with dataset size, model complexity, and optimization requirements. Cloud computing provides scalable resources and changes the relationship between computation and physical infrastructure (Armbrust et al., 2010).

Distributed computing can divide computational tasks among multiple resources. Folding@home demonstrates how geographically distributed computational resources can contribute to scientific calculations (Beberg et al., 2009). This approach is conceptually important for intelligent computation because large models and simulations may require more computational capacity than conventional laboratory systems can provide.

The infrastructure challenge is not purely about speed. Distributed systems introduce communication overhead, synchronization requirements, fault tolerance, data movement, security, and resource allocation issues. Therefore, algorithmic scalability must be considered together with architectural scalability.

3.14 Computational efficiency and future hardware

As intelligent systems become more complex, hardware efficiency becomes increasingly important. Caulfield and Dolev argued that future supercomputing requirements could motivate increased use of optical computation (Caulfield and Dolev, 2010).

This observation suggests a three-level scalability problem:

$$[\begin{array}{l} \text{Algorithmic scalability} \\ + \\ \text{System scalability} \\ + \\ \text{Hardware efficiency} \end{array}]$$

An algorithm may be mathematically efficient but computationally expensive when implemented at large scale. Conversely, powerful hardware cannot compensate indefinitely for inefficient algorithmic structures.

Future intelligent scientific systems will therefore need co-design between algorithms and computational architectures.

3.15 Responsible design and model complexity

Increasing computational capability also creates methodological and societal concerns. Bender et al. examined risks associated with very large language models, emphasizing that increasing model size does not automatically resolve issues related to data, environmental cost, documentation, or social impact (Bender et al., 2021).

Although the present review covers a broader range of scientific systems, the principle is transferable. Model complexity should be justified by computational requirements and scientific value. A larger model is not inherently a better model. The appropriate architecture depends on the problem, available data, validation objectives, computational resources, and deployment environment.

Consequently, scientific intelligent systems should be evaluated using multiple dimensions:

$$[\begin{array}{l} \text{Utility} \\ f(\text{accuracy}, \text{robustness}, \text{stability}, \\ \text{interpretability}, \text{cost}, \text{scalability}) \end{array}] =$$

This multidimensional perspective is more appropriate than relying on a single performance metric.

4. Results / Findings

The synthesis of the supplied literature identifies a coherent progression from foundational neural computation toward integrated intelligent computational systems. The first major finding is that neural computation has expanded from relatively general nonlinear approximation toward specialized architectures designed for deep representation, temporal learning, complex-valued computation, mathematical equation solving, and uncertainty-aware reasoning. Foundational work on deep architectures, deep belief

networks, and long short-term memory established important mechanisms for hierarchical and temporal representation (Hinton et al., 2006; Bengio, 2009; Hochreiter and Schmidhuber, 1997). Subsequent research demonstrates that these principles can be adapted to specialized scientific problems.

The second finding is that intelligent computation increasingly operates as a hybrid methodology. The reviewed applications rarely depend exclusively on one computational mechanism. Instead, neural networks, optimization algorithms, fuzzy logic, neutrosophic reasoning, and control mechanisms are combined according to task requirements. The integration of adaptive fuzzy control with zeroing neural networks, for example, demonstrates how different computational paradigms can address time-varying optimization problems (Jia et al., 2021). BAS-Adam similarly illustrates the combination of metaheuristic search with adaptive gradient optimization (Khan et al., 2020a).

The third finding concerns the transition from prediction to action. In engineering applications, the usefulness of a computational model frequently depends on whether its output can influence a physical or operational process. Wind-speed estimation is connected to optimal torque control, while robotic prediction and optimization are connected to motion generation and trajectory tracking (Deng et al., 2020; Khan et al., 2021). Consequently, intelligent systems should be assessed within their operational context rather than as isolated prediction engines.

The fourth finding is the importance of problem-specific mathematical structure. Complex-valued recurrent neural networks, zeroing neural networks, and neutrosophic systems demonstrate that scientific computation benefits when algorithms reflect the structure of the underlying problem (Ding et al., 2018, 2019b; Katsikis et al., 2023). This suggests that generic learning capacity is not always sufficient. Domain-specific structure can improve the relevance of computational representations.

The fifth finding is the broad transferability of intelligent optimization. BAS-related research extends from operational management and portfolio selection to fraud detection and robotic control (Khan et al., 2020b, 2022a, 2022b, 2022c). This cross-domain application suggests that the value of a metaheuristic lies in its underlying

search principle rather than its association with a particular application.

The sixth finding is that computational infrastructure forms an essential part of intelligent scientific systems. Cloud computing, distributed scientific computation, and prospective optical computing address the resource requirements generated by increasingly complex computational workloads (Armbrust et al., 2010; Beberg et al., 2009; Caulfield and Dolev, 2010). Algorithmic innovation and infrastructure development are therefore mutually dependent.

The seventh finding is that stability and reliability remain fundamental limitations. Research on recurrent-network stability, time-varying equation solving, and noise-resistant zeroing neural networks demonstrates that computational performance cannot be separated from mathematical behavior (Ding et al., 2014; Han et al., 2023). In safety-critical engineering applications, a model must remain reliable when inputs, system conditions, or environmental variables change.

The eighth finding is that model complexity introduces trade-offs. Deep and distributed computational systems can increase representational capacity and scalability, but they can also increase computational requirements and introduce additional sources of uncertainty. Bender et al.'s discussion of large language models illustrates broader concerns associated with scaling computational models without proportional consideration of data and system-level consequences (Bender et al., 2021).

The synthesized framework therefore supports a layered interpretation of intelligent computation: representation provides information structure; learning provides inference; optimization identifies useful solutions; control translates computational outputs into actions; and computational infrastructure enables practical execution. These layers are not independent. Their interactions determine overall system performance.

5. Discussion

The findings indicate that intelligent computational systems should be understood as integrated computational ecosystems rather than collections of independent algorithms. This interpretation is supported by the progression from deep representation learning to recurrent computation, optimization, control, and distributed infrastructure. The literature shows that

scientific applications increasingly require several of these capabilities simultaneously.

Theoretically, the most important implication is the convergence of learning-based and model-based computation. Traditional machine learning emphasizes learning relationships from data, whereas mathematical computational approaches can explicitly encode system constraints. Zeroing neural networks illustrate a hybrid perspective in which neural dynamics are constructed around mathematical solution requirements (Jia et al., 2021; Katsikis et al., 2023). Such approaches may be particularly valuable when scientific problems have strong mathematical structure.

A second implication concerns the role of optimization. Intelligent learning does not eliminate the need for optimization; rather, learning and optimization can be complementary. BAS-Adam demonstrates this explicitly by integrating metaheuristic search with an adaptive optimization mechanism (Khan et al., 2020a). Similar principles appear in robotics and portfolio selection, where learning or estimation must ultimately be combined with constrained search.

A third implication concerns scientific interpretability. As computational systems become more complex, output accuracy alone becomes insufficient for many scientific applications. Researchers need to understand whether a computational model captures physically meaningful relationships, remains stable under perturbations, and behaves consistently outside the training distribution. Ethayarajh's examination of representation geometry illustrates why the internal structure of learned representations can itself be scientifically relevant (Ethayarajh, 2019).

A fourth issue is generalization. An intelligent system trained under one operating condition may not behave similarly under another. This challenge is particularly important for engineering systems exposed to changing environmental conditions. Wind-energy systems, landslide monitoring, electrical infrastructure, and robotic control all operate in dynamic environments (Deng et al., 2020; Hu et al., 2019a, 2021; Huang et al., 2022). Consequently, robustness should be evaluated as a core scientific property.

The literature also reveals a trade-off between specialization and generalization. Specialized computational models can exploit domain-specific structure, as demonstrated by complex-valued recurrent

networks and zeroing neural networks. However, such models may be less transferable than general-purpose architectures. Conversely, highly general architectures may require larger datasets and computational resources and may not automatically encode relevant physical constraints.

Computational infrastructure introduces another trade-off. Cloud and distributed computing improve scalability, but distribution creates communication, synchronization, and resource-management challenges (Armbrust et al., 2010; Beberg et al., 2009). The future development of intelligent systems therefore requires attention to both algorithmic and infrastructure-level efficiency.

The reviewed literature also suggests that hybridization should be selective rather than indiscriminate. Combining several algorithms may increase capability, but each additional component introduces parameters, dependencies, and potential failure modes. A hybrid architecture should therefore be justified by a clearly identified computational requirement.

Robotics provides a particularly useful illustration. A redundant manipulator may benefit from recurrent prediction, metaheuristic optimization, and feedback control, but the integration must preserve physical feasibility and stability. Similarly, portfolio optimization may benefit from intelligent search, but the model must remain consistent with financial constraints (Katsikis et al., 2021; Khan et al., 2022a).

Another major consideration is computational cost. Deep learning and distributed intelligence can require substantial resources. Caulfield and Dolev's discussion of optical supercomputing emphasizes that future computational demands may require alternative hardware architectures (Caulfield and Dolev, 2010). Therefore, the future of intelligent computation cannot be separated from advances in computing infrastructure.

The literature also supports a distinction between predictive intelligence and decision intelligence. Predictive intelligence estimates what may happen, whereas decision intelligence determines what should be done under defined objectives and constraints. Deng et al.'s wind-energy work exemplifies this transition from estimation to control (Deng et al., 2020). Robotic and financial applications demonstrate similar relationships.

The main limitations of the reviewed literature are methodological heterogeneity and application

specificity. The studies address different domains, objectives, datasets, computational models, and evaluation criteria. Direct numerical comparison between their reported outcomes would therefore be inappropriate. Instead, the literature is most useful for identifying common computational principles.

Another limitation is that the supplied reference set contains duplicate bibliographic information for the Deng et al. wind-energy study. This does not alter the conceptual findings, but it indicates that bibliographic normalization is important when constructing systematic evidence bases.

Future research should investigate integrated architectures combining domain constraints, representation learning, optimization, and feedback control. Such architectures could potentially reduce the separation between data-driven and model-driven computation. Another direction involves adaptive systems that modify computational strategies according to changing operating conditions. Distributed intelligence could additionally enable collaborative computational systems in which multiple agents share information and computational resources.

Finally, responsible computational design should become a standard component of scientific methodology. The concerns identified by Bender et al. regarding large computational models reinforce the importance of documenting computational costs, data limitations, model assumptions, and potential failure modes (Bender et al., 2021). Scientific usefulness should therefore be evaluated through a combination of performance, reliability, efficiency, transparency, and domain validity.

6. Conclusion

Intelligent computational systems have evolved from conventional neural-network models into broad computational frameworks capable of representation learning, temporal modeling, optimization, mathematical problem solving, prediction, classification, control, and distributed scientific computation. The literature reviewed in this paper demonstrates that these capabilities are increasingly interconnected.

Foundational neural-network research established the principles of nonlinear representation and temporal learning, while deep learning expanded these principles toward hierarchical feature extraction (Hochreiter and Schmidhuber, 1997; Bengio, 2009; LeCun et al., 2015).

Subsequent work demonstrated that recurrent, complex-valued, zeroing, fuzzy, neutrosophic, and metaheuristic methods can be adapted to specialized scientific problems.

Applications across wind-energy systems, electrical infrastructure, landslide prediction, robotic manipulation, financial optimization, fraud detection, and dynamic equation solving demonstrate that intelligent computation has become a multidisciplinary methodology rather than a narrowly defined machine-learning technique. In many of these applications, the computational model does not operate independently. Prediction interacts with optimization, optimization interacts with control, and all computational functions depend on suitable infrastructure.

The principal contribution of this synthesis is the conceptualization of intelligent computation as a layered system consisting of data representation, learning or inference, optimization, decision/control, and computational infrastructure. This framework provides a basis for understanding why different computational approaches are complementary rather than mutually exclusive.

The reviewed literature also demonstrates that increasing computational sophistication introduces important limitations. Model complexity may increase resource requirements; optimization may be sensitive to problem formulation; neural computation may require stability analysis; and distributed computation may introduce communication and synchronization costs. Consequently, computational intelligence should not be evaluated solely by predictive accuracy or optimization performance.

Future research should focus on hybrid intelligent architectures that combine learned representations with mathematical constraints, optimization, uncertainty modeling, and feedback control. Such systems may provide improved adaptability while retaining domain-specific validity. Further integration with scalable cloud, distributed, and potentially optical computing infrastructures may support increasingly demanding scientific workloads.

Ultimately, the development of intelligent computational systems should be guided by the characteristics of the scientific problem rather than by the popularity of an individual algorithm. The most appropriate computational architecture is one that balances

representational capacity, mathematical validity, computational efficiency, stability, interpretability, scalability, and application-specific requirements. This perspective provides a foundation for developing intelligent systems capable of addressing increasingly intricate scientific and engineering problems.

7. References

1. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, et al. A view of cloud computing. *Commun ACM*. 2010;**53**(4):50–58.
2. Beberg AL, Ensign DL, Jayachandran G, Khaliq S, Pande VS. Folding@home: Lessons from eight years of volunteer distributed computing. Paper presented at: IEEE International Symposium on Parallel & Distributed Processing; 2009 May 3–29; Rome, Italy. p. 1–8.
3. Bender EM, Gebru T, McMillan-Major A, Shmitchell S. On the dangers of stochastic parrots: Can language models be too big? Paper presented at: Proceedings of the ACM Conference on Fairness, Accountability, and Transparency, 2021; Virtual Event, Canada; p. 610–623.
4. Bengio Y. Learning deep architectures for AI. *Mach Learn*. 2009;**2**(1):1–127.
5. Bengio Y, Glorot X, Understanding the difficulty of training deep feedforward neural networks. Paper presented at: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics; 2010 May 13–15; Chia Laguna Resort, Sardinia, Italy. p. 249–256.
6. Caulfield HJ, Dolev S. Why future supercomputing requires optics. *Nat Photonics*. 2010;**4**(5):261–263.
7. Deng X, Yang J, Sun Y, Song D, Yang Y, Joo Y. H. (2020). An effective wind speed estimation based extended optimal torque control for maximum wind energy capture. *IEEE Access* 8, 65959–65969. doi: 10.1109/ACCESS.2020.2984654.
8. Deng X, Yang J, Sun Y, Song D, Yang Y, Joo Y. H. (2020). An effective wind speed estimation based extended optimal torque control for maximum wind energy capture. *IEEE Access* 8, 65959–65969. doi: 10.1109/ACCESS.2020.2984654.
9. Ding B, Li X, Li Y. (2022). Configuration design and experimental verification of a variable constant-force compliant mechanism. *Robotica* 40, 3463–3475. doi: 10.1017/S0263574722000340.
10. Ding B, Yang Z, Xiao X, Zhang G. (2019a). Design of reconfigurable planar micro-positioning stages based on function modules. *IEEE Access* 7, 15102–15112. doi: 10.1109/ACCESS.2019.2894619.
11. Ding B, Zhao J, Li Y. (2021). Design of a spatial constant-force end-effector for polishing/deburring operations. *Int. J. Adv. Manufact. Technol.* 116, 3507–3515. doi: 10.1007/s00170-021-07579-1.
12. Ding L, She J, Peng S. (2013). An integrated prediction model for network traffic based on wavelet transformation. *Elektronika ir elektrotechnika* 19, 73–76. doi: 10.5755/j01.eee.19.3.3700.
13. Ding L, Xiao L, Zhou K, Lan Y, Zhang Y. (2018). A new RNN model with a modified nonlinear activation function applied to complex-valued linear equations. *IEEE Access* 6, 62954–62962. doi: 10.1109/ACCESS.2018.2876665.
14. Ding L, Xiao L, Zhou K, Lan Y, Zhang Y, Li J. (2019b). An improved complex-valued recurrent neural network model for time-varying complex-valued Sylvester equation. *IEEE Access* 7, 19291–19302. doi: 10.1109/ACCESS.2019.2896983.
15. Ding L, Zeng H, Wang W, Yu F. (2014). Improved stability criteria of static recurrent neural networks with a time-varying delay. *Sci. World J.* 2014, 391282. doi: 10.1155/2014/391282.
16. Ethayarajh K. *How contextual are contextualized word representations? Comparing the geometry of bert, elmo, and gpt-2 embeddings*. Paper presented at: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics; 2019. Hong Kong, China. p. 55–65.
17. Han L, He Y, Liao B, Hua C. (2023). An accelerated double-integral ZNN with resisting linear noise for dynamic Sylvester equation solving and its

- application to the control of the SFM chaotic system. *Axioms* 12, 287. doi: 10.3390/axioms12030287.
18. Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. *Neural Comput.* 2006;18(7):1527–1554.
 19. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput.* 1997;9(8):1735–1780.
 20. Hu H, Fang M, Hu F, Zeng S, Deng X. (2021). A new design of substation grounding based on electrolytic cathodic protection and on transfer corrosion current. *Electric Power Syst. Res.* 195, 107174. doi: 10.1016/j.epsr.2021.107174.
 21. Hu H, Luo R, Fang M, Zeng S, Hu F. (2019a). A new optimization design for grounding grid. *Int. J. Electrical Power Energy Syst.* 108, 61–71. doi: 10.1016/j.ijepes.2018.12.041.
 22. Hu K, He W, Ye J, Zhao L, Peng H, Pi J. (2019b). Online visual tracking of weighted multiple instance learning via neutrosophic similarity-based objectness estimation. *Symmetry* 11, 832. doi: 10.3390/sym11060832.
 23. Huang Z, Dong M, Du X, Guan Y. (2022). A nonlinear prediction model of antislide pile top displacement based on MIC-SVR for jurassic landslides. *Adv. Civil Eng.* 2022, 9101234. doi: 10.1155/2022/9101234.
 24. Jia L, Xiao L, Dai J, Qi Z, Zhang Z, Zhang Y. (2021). Design and application of an adaptive fuzzy control strategy to zeroing neural network for solving time-variant QP problem. *IEEE Trans. Fuzzy Syst.* 29, 1544–1555. doi: 10.1109/TFUZZ.2020.2981001.
 25. Jin J, Yu F. (2013). Novel current-reuse current-mirror and its application on 2.4-GHz down-conversion mixer. *Microwave Opt. Technol. Lett.* 55, 2520–2524. doi: 10.1002/mop.27878.
 26. Jin J, Zhou K, Zhao L. (2017a). Designing RF ring oscillator using current-mode technology. *IEEE Access* 5, 5306–5312. doi: 10.1109/ACCESS.2017.2692771.
 27. Jin L, Li S, Xiao L, Lu R, Liao B. (2017b). Cooperative motion generation in a distributed network of redundant robot manipulators with noises. *IEEE Trans. Syst. Man Cybernet. Syst.* 48, 1715–1724. doi: 10.1109/TSMC.2017.2693400.
 28. Katsikis V. N, Mourtas S. D. (2021). “Portfolio insurance and intelligent algorithms,” in *Computational Management: Applications of Computational Intelligence in Business Management* (Springer), 305–323. doi: 10.1007/978-3-030-72929-5_14.
 29. Katsikis V. N, Mourtas S. D, Stanimirović P. S, Li S, Cao X. (2021). “Time-varying mean-variance portfolio selection under transaction costs and cardinality constraint problem via beetle antennae search algorithm (BAS),” in *Operations Research Forum*, Vol. 2 (Springer), 1–26. doi: 10.1007/s43069-021-00060-5.
 30. Katsikis V. N, Stanimirović P. S, Mourtas S. D, Xiao L, Stanujkić D, Karabašević D. (2023). Zeroing neural network based on neutrosophic logic for calculating minimal-norm least-squares solutions to time-varying linear systems. *Neural Process. Lett.* doi: 10.1109/TNNLS.2022.3171715-7.
 31. Khan A. H, Cao X, Li S, Katsikis V. N, Liao L. (2020a). BAS-Adam: an Adam based approach to improve the performance of beetle antennae search optimizer. *IEEE/CAA J. Automat. Sin.* 7, 461–471. doi: 10.1109/JAS.2020.1003048.
 32. Khan A. H, Cao X, Li S, Luo C. (2020b). Using social behavior of beetles to establish a computational model for operational management. *IEEE Trans. Comput. Soc. Syst.* 7, 492–502. doi: 10.1109/TCSS.2019.2958522.
 33. Khan A. H, Li S, Cao X. (2021). Tracking control of redundant manipulator under active remote center-of-motion constraints: an RNN-based metaheuristic approach. *Sci. China Inform. Sci.* 64, 1–18. doi: 10.1007/s11432-019-2735-6.
 34. Khan A. T, Cao X, Brajevic I, Stanimirovic P. S, Katsikis V. N, Li S. (2022a). Non-linear activated beetle antennae search: a novel technique for non-convex tax-aware portfolio optimization problem. *Expert Syst. Appl.* 197, 116631. doi: 10.1016/j.eswa.2022.116631.
 35. Khan A. T, Cao X, Li S. (2022b). Dual beetle antennae search system for optimal planning and

- robust control of 5-link biped robots. *J. Comput. Sci.* 60, 101556. doi: 10.1016/j.jocs.2022.101556.
36. Khan A. T, Cao X, Li S, Katsikis V. N, Brajevic I, Stanimirovic P. S. (2022c). Fraud detection in publicly traded US firms using beetle antennae search: a machine learning approach. *Expert Syst. Appl.* 191, 116148. doi: 10.1016/j.eswa.2021.116148.
37. Khan A. T, Cao X, Liao B, Francis A. (2022d). Bio-inspired machine learning for distributed confidential multi-portfolio selection problem. *Biomimetics* 7, 124. doi: 10.3390/biomimetics7030124.
38. Khan A. T, Li S, Cao X. (2022e). Human guided cooperative robotic agents in smart home using beetle antennae search. *Sci. China Inform. Sci.* 65, 122204. doi: 10.1007/s11432-020-3073-5.
39. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015;**521**(7553):436–444.
40. Lepikhin D, Lee HJ, Xu Y, Chen D, Firat O, Huang Y, Krikun M, Shazeer N, Chen Z. Gshard: Scaling giant models with conditional computation and automatic sharding. arXiv. 2021. <https://doi.org/10.48550/arXiv.2006.16668>
41. Poole D, Mackworth A, Goebel R. *Computational intelligence*. Oxford: Oxford University Press; 1998.
42. Schmidhuber J. Deep learning in neural networks: An overview. *Neural Netw.* 2015;**61**:85–117.