WCRG Global Research WORLD COMET RESEARCH GROUP

International Journal of Modern Innovations and Emerging Trends
Vol. 3, No. 2, 2020
Doi: 10.67228/30715628/IJMIET-2020PII1X8Q
PP. 01-14

Original Article

Emerging Trends in Bio-Inspired Computing Models

Dr. Rajesh Kumar Sharma

Professor, University of Delhi, India.

Received: 06-06-2020 Revised: 06-27-2020 Accepted: 07-03-2020 Published: 07-05-2020

ABSTRACT

Bio-inspired computing models are computational approaches inspired by biological systems such as evolution, neural processing, swarm behavior, immune systems, and cellular communication. They are widely used for optimization, learning, adaptation, and control in complex environments. Recent advances in AI, edge computing, healthcare, smart cities, and autonomous systems have increased interest in these models due to their adaptability, robustness, and energy efficiency. Current research focuses on evolutionary computing, swarm intelligence, neural and spiking neural networks, immune-inspired computing, and hybrid approaches that combine multiple biological principles. Emerging technologies such as neuromorphic computing, memristive hardware, and analog in-memory computing further support low-power intelligent systems. This work presents a unified framework based on biological mechanisms, computational objectives, deployment contexts, and performance constraints, demonstrating its application in edge-IoT anomaly detection and resource management. Overall, bio-inspired computing is emerging as a practical and effective paradigm for developing adaptive, distributed, and energy-efficient intelligent systems.

KEYWORDS

Bio-Inspired Computing , Evolutionary Algorithms, Genetic Algorithms,Particle Swarm Optimization,Ant Colony Optimization, Artificial Immune Systems, Spiking Neural Networks, Neuromorphic Hardware, Membrane Computing, Hybrid Optimization,Edge Intelligence, Distributed Systems, Robust AI.

ISSN: 3071-5628
PP: 01-14.

1. INTRODUCTION

1.1. Background

Biological systems exhibit strong computational properties that are of great relevance to intelligent systems in the modern world: they are self-adaptive to changing environments, they can be highly efficient even with strict resource limitations, and strong even in presence of noise, partial failure, and uncertainty. The following properties drive a diverse collection of bio-inspired computational models intended to address problems that are frequently challenging in conventional methods, such as non-stationary environments, limited or imbalanced training information, controlling tasks based on cyber-physical devices, and deploying this control to constrained energy resources on edges. Notions In most real-life problems, the use of classical optimization does not require smoothness, stable objective, or reliable gradients, whereas pure data learning can bog down with large labeled data sets and tend to become worse when the distribution changes. The two paradigms may be challenged by geometries of high dimension, discontinuous cost functions, multi-objective cost functions, and dynamically changing constraint during dynamic systems. Bio-inspired models provide an alternative which exploits the idea of population-based search, decentralized coordination and self-organizing adaptation to explore complex landscapes without making rigid mathematical assumptions. Through diversity maintenance, stochastic exploration, and simple interaction rules, they are capable of generating solutions that are both effective and robust, and flexible, which is why any of them is useful in the design of intelligent systems which need to learn and behave with a high level of reliability when faced with realistic, changing conditions.

1.2. Bio-Inspired Computing Families

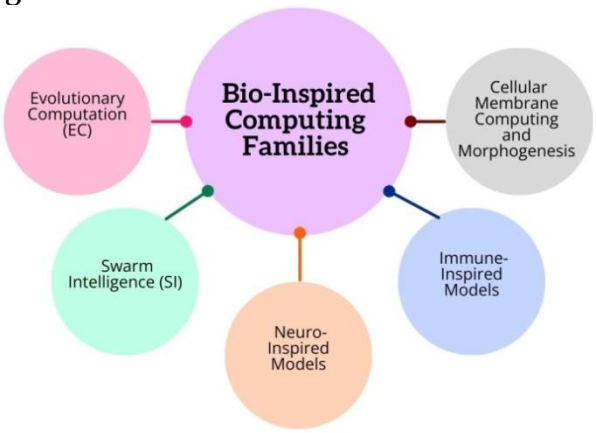

Fig 1 - Bio-Inspired Computing Families

1.2.1. Evolutionary Computation (EC)

Evolutionary computation also forms part of algorithms that simulate natural selection, through the use of an evolving population of candidate solutions across generations. Genetic Algorithms (GA), Differential Evolution (DE), and Evolution Strategies (ES) are methods that are iterative and based on neglecting selection and variation (mutation / recombination) to enhance fitness, whereas Genetic Programming (GP) is a method that evolves program code or symbolic expressions. EC has especially been applied to high-dimensional non-convex and discontinuous optimization where gradients are not available, and we need only go to multi-objective optimization where we seek not a best solution but a representation of Pareto-optimal trade-offs.

ISSN: 3071-5628
PP: 01-14.

1.2.2. Swarm Intelligence (SI)

Swarm intelligence dwells on collective behaviour that arises out of simple interactions of large numbers of agents and results in decentralized optimization and coordination. Particle Swarm Optimization (PSO) is popular due to continuous search, consisting of a combination of individual learning and social influence, whereas Ant Colony Optimization (ACO) is effective in routing and combinatorial graph issues, which are based on the pheromone-like reinforcement. Such alternative exploration-exploitation dynamics and communication patterns as Artificial Bee Colony (ABC), Firefly, Bat algorithms are proposed, which is why SI can be considered in a context of distributed system, where robustness and adaptability are important.

1.2.3. Neuro-Inspired Models

Neuro-inspired models are based on the organization and functionality of nervous systems to acquire representations of data and aid intelligent decision-making. Most of the perception and prediction tasks are dominated by classical Artificial Neural Networks (ANN) and deep learning involving hierarchical feature learning. Spiking Neural Networks (SNN) are a generalization of this concept, which applies time-varying spikes and makes event-based computation accessible to streaming signals and edge computing on low-power devices. Another neuro-inspired direction proposed by reservoir computing takes advantage of the dynamic recurrent systems to perform temporal processing using a much simplified training pipeline, and it is usually beneficial in sequence learning in real-time.

1.2.4. Immune-Inspired Models

Such Immune-inspired models are models that attempt to recreate biological immunity, in which normal behavior is considered the self, and impairments in behavior are considered non-self that evolves detectors with time. Artificial Immune Systems (AIS) are typically applied to detect anomalies, monitor faults and develop cyber resilience. The detection algorithms are applicable to clonal selection whereby each high-performing detectors undergoes replication and mutation to enhance its performance, and negative selection whereby detectors that do not match normal patterns are produced to minimize false alarms. Immune network models are models of interaction dynamics between detectors, enabling diversity and adaptive memory to evolve in varying environments.

1.2.5. Cellular/Membrane Computing and Morphogenesis

Cellular and morphogenetic methods investigate the way global complex forms are created out of local interactions as happens in biological development and chemical signaling. P systems (membrane computing) A parallel set of distributionally silent compartments with rule-based transformations describe computation in a parallel and distributed perspective of problem solving. Some examples of pattern formation, self-organization and even self-repair are illustrated by reaction diffusion models and cellular automata using a set of simple local rules. Developmental computation builds upon these concepts by having growth and differentiation processes, which can generate scalable and robust designs, although scaling them consistently to standardised benchmarks is an ongoing research field.

1.3. Emerging Trends in Bio-Inspired Computing Models

Bio-inspired computing is quickly moving beyond single algorithmic implementations of heuristic processes to systems oriented thinking and execution through modern machine learning, edge computing, and cyber-physical solutions. One trend is hybridization, in which evolutionary search, swarm coordination, neuro-inspired learning, and immune-style monitoring are combined in

ISSN: 3071-5628
PP: 01-14.

multiple layers of workflow such that exploration, adaptation, and efficiency are tackled in a unified manner as opposed to individually. In parallel with this, the community is gravitating towards multi-objective and constraint-sensitive optimization, which is indicative of actual operational demands in the form of accuracy latency energy trade-offs, fairness, reliability, and communication constraints; Pareto-fueled evaluation, and quality-diversity methods are increasingly being employed to give rise to solution sets rather than an optimum solution. Resource-efficient intelligence (motivated by edge deployment) is another key direction that is getting attention, since event-driven inference using low power consumption and event responsiveness can be realized through spiking neural networks, neuromorphic hardware, sparse computation, quantization, and pruning. Simultaneously, bio-inspired techniques are being scaled to distributed and federated systems, where coordination through swarm-like coordination can determine which clients to serve, aggregation approaches, and resource utilization with bandwidth limits and data distributions with heterogeneities. In order to overcome the classical objections that bio-inspired algorithms may be computationally infeasible, there is an increasing interest in cost-sensitive acceleration, such as surrogate models, multi-fidelity analysis, early termination and learned predictors of fitness that minimise costly evaluations with no loss in search power. The other theme emerging is robustness and trust in which immune-inspired detectors and self-adaptation mechanisms are applied to deal with concept drift, adversarial behaviour and operational uncertainty, particularly during security critical applications. Lastly, the field is getting rigorous with standardized benchmarking and reporting, and there are increasing demands to publish compute budgets, energy per inference, latency distributions, and adaptation time, to allow fair comparisons and enable progress to be made reproducible and deployment relevant.

2. *LITERATURE SURVEY*

2.1. Evolutionary Computation: From Classical GA to Neuro evolution

Evolutionary computation (EC) has progressed to a significant amount compared to early genetic algorithms (GAs) which used only fixed crossover/mutation operators and hand-tuned parameters. In modern EC, adaptivity and efficiency is important: in Differential Evolution (DE), Evolution Strategies (ES) self-adaptive components can vary over the course of the run, increasing robustness to problem instances and eliminating costly trial-and-error tuning. EC has developed at the same time into neuro evolution and Neural Architecture Search (NAS), where instead of optimizing weights, evolutionary algorithms can also optimize network architectures, hyperparameters, even training recipes; providing an alternative to gradient-based design, particularly when objectives are discontinuous, multi-modes, or constrained by hardware. Multi-objective and quality-diversity optimized algorithms An alternative significant change is being made to discover a repertoire of high-quality, diverse solutions instead of an optimum, and in this case, the situation of deployment is heterogeneous, or interpretability and resiliency are important. Regardless of such progress, the cost of computation is also a limiting factor especially in NAS where the candidate can be costly to train and evaluate. In response to this, recent EC practice is adopting a variety of surrogate modeling (predicting performance with partial signals), multi-fidelity assessment (cheap approximations prior to full training) and early stopping/bandit-like allocation of computational resources to promising candidates, so EC can achieve higher scalability without compromising its exploratory capabilities.

2.2. Swarm Intelligence: Distributed Optimization and Coordination

Collective behavior or flocking, forging, ant trail formation, swarm intelligence techniques are inspired by natural examples of collective behavior:Swarm intelligence techniques are of particular

ISSN: 3071-5628
PP: 01-14.

interest in distributed and decentralized systems, where global coordination is challenging or expensive. The particle swarm optimization (PSO) is still among the most popular continuous optimization, controller tuning and adaptive parameter search methods: it is straightforward, gradient-free, and inherently parallel: particles exchange partial information between themselves and descend upon candidate solutions by social learning dynamics. ACO remains influential in routing and scheduling and other combinatorial graph optimization problems in which the pheromone-like memory is used to facilitate exploration-exploitation trade-offs and distributed path building. Later forms of swarms seek to do a better job in realistic settings, causing a combination of dynamic neighborhood topologies (communicating with who) and multi-swarm or coevolutionary programs (sub-swarms specializes on different areas or goals), and adaptive communication (change the intensity of interaction with constraint violations). These concepts are applicable to contemporary distributed learning systems, such as federated learning, where swarm principles may be applicable to coordinate client selection, local training schedules, adapt aggregation policies in the event of heterogeneous data, and bandwidth limits. In general, swarm intelligence is a scalable and adaptable optimization and coordination framework, although continuing studies aim at ensuring that stability and convergence remain in case the environment is non-stationary and unreliable/expensive communication is used.

2.3. Immune-Inspired Models: Anomaly Detection and Cyber Resilience

Artificial Immune Systems (AIS) is computed based on biological immunity metaphors, where normal patterns are the self and anomalies are the non-self, as such it is conceptually suitable to intrusion detection, fault monitoring and cyber resilience in distributed infrastructures. Negative selection algorithms are based on the idea that they replicate the process of eliminating self-reactive detectors, generating a set of detectors which is (in theory) only sensitive to abnormal patterns; this is desirable in the security case since it allows decentralized monitoring with no explicit labelling of each type of attack. Clonal selection schemes based on the growth and diversification of high-affinity immune cells have applied to optimization as well as to adaptive classification: improving detectors with new evidence coming. In practice however, on real deployments, naive non-self detection can be very prone to large false positive rates due to the diversity of normal behavior and optimistic behavior changing over time (concept drift) (e.g. new workloads, software updates, new user behavior). In response to this, danger theory inspired models add contextual cues such as indicators of tissue damage or other stress to alert mechanism, such that alerts are not only determined based on the mismatch of a pattern, but also based on demonstration of danger, enhancing sensitivity to noise. The key trade-off between sensitivity (true anomaly detection fast) and specificity (preventing false alarms) has inspired a growing interest in hybrid AIS systems based on streaming adaptation, contextual features and an ensemble decision-making approach to robust long-term operation.

2.4. Spiking and Neuromorphic Models: Efficient Event-Driven Intelligence

Spiking Neural Networks (SNNs) encode information with discrete spikes, as a function of time instead of continuous activations, and are more biological, as well as allow event-driven processing. This temporal property would render SNN to be of core importance in low-power, real-time intelligence especially when combined with neuromorphic hardware which takes advantage of the sparse activity: when neurons firing are infrequent, the energy consumption of computation and memory access can be very low, better adapted to the needs of edge devices. Increment in temporal coding scheme (e.g., latency-based or rate-based codecs) and in sparse dynamics have enhanced representational capacity whilst maintaining efficiency. Significant historical obstacles have been training, since generating spikes cannot be differentiable; recent solutions to this problem include

ISSN: 3071-5628
PP: 01-14.

surrogate gradient methods, which use spike generation trained using gradients, which are then solved to generate a spiking spike at inference. With the current progress towards practical implementation of SNNs, more attention in research has been on hardware-conscious learning: quantization, low-precision synapses, learning efficiently constrained by hardware on chips, and trade-offs between latency and throughput determine which models can be effectively implemented using a particular neuromorphic platform. The consequence is an ever-expanding ecosystem in which algorithm design, training methodology, and hardware characteristics should be co-evolved with the goal of producing systems that are not merely able to be accurate but also able to be efficient, robust to timing noise, and always-on sensing.

2.5. Cellular, Membrane, and Morphogenetic Computation

The cellular, membrane, and morphogenetic computation research the possibility of complex global behaviour being generated by simple local rules, inspired by biological development, chemical signalling, and self-organising tissues. Cellular automata (CA) and reaction diffusion systems exemplify how simple update rules on local patches of cells and spatial interactions can produce rich pattern formation stripes, spots, waves etc, and these concepts have been applied to distributed control and collective construction and self-repairing systems. Computation Membrane computing (P systems) conceptualizes computation in terms of compartments nested inside other compartments and rules that rewrite, inspired by biochemical mechanisms and making innovative massively parallel transformations conceptually consistent with distributed computation. The morphogenetic or developmental approaches go even further and lay more stress on the growth, differentiation and regeneration where solutions are not necessarily encoded as final structures but rather as developmental programs that may be replicated over a period of time often with robustness and scalability. These paradigms are attractive to engineering systems that need to be adapted locally (e.g., swarms, modular robots, sensor networks) due to their lack of centralized control and capability to re-organize locally after some failure. Nevertheless, one of the outstanding issues is methodological in nature: how to project these biologically-based models onto standard benchmarks and evaluation procedures is an emerging field, and it is becoming increasingly difficult to pit the methods against other rigorous approaches of optimisation or learning. Further advances are increasingly being made with respect to conceptualizing benchmark suites that not only best reflects what these models do, but also relates them to concrete performance measures that are pertinent to actual deployments.

3. METHODOLOGY

3.1. Hybrid Bio-Inspired Computing Workflow (HBCW)

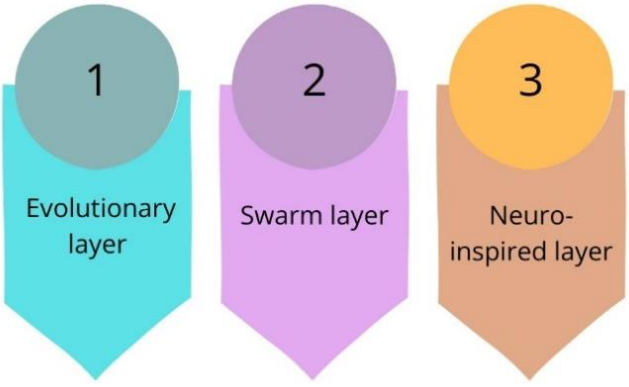

Fig 2 - Hybrid Bio-Inspired Computing Workflow (HBCW)

ISSN: 3071-5628
PP: 01-14.

3.1.1. Evolutionary layer

The evolutionary layer carries out exploration in the world, taking refined candidate solutions and enables evolution through selection, variation (mutation/crossover) and the pressure of survival of the fittest. This layer in an HBCW pipeline has the usual role of finding good initial design or configuration (e.g. feature subsets, model hyperparameters, control policies, architectures), and maintaining a mixed population to ensure the search does not early terminate. It is best-adapted to the rugged, multi-modal search spaces in which gradients are either non-existent or unreliable, and that can be generalized to problems multi-objective optimization (e.g. accuracy vs. latency vs. energy). To regulate the computational cost, evolutionary layer frequently incorporates early stopping, surrogate fitness prediction or multi-fidelity evaluation to efficiently eliminate weak candidates, and give full evaluation to promising ones.

3.1.2. Swarm layer

The swarm layer focuses on distributed coordination and quick refinement by means of collective information sharing in the agents (particles/ants/agents). In HBCW, it normally serves as an effective to intermediary stage optimizer which accepts the promising regions found by the evolution and executes local-to-semi-global search quickly and cooperatively. In continuous parameter gain the PSO-style has the ability to modify solutions by integrating personal experience (best-so-far) and social learning (neighborhood/global best), and the graph/routing subproblems can be solved by ACO-like constructive search with adaptive pheromone updates. Swarm methods inherently embrace decentralization, hence, can also be found applicable in cases when the workflow needs to be executed on many nodes/clients with minimal communication - agents can be highly efficient on a local level and will exchange small summaries at periodic intervals to synchronize their overall search path.

3.1.3. Neuro-inspired layer

Neuro-inspired layer offers learning-based adaptation and fine-tuning, which usually employs neural models to represent intricate patterns, temporal dynamics, or feedback of the environment. In HBCW, this layer frequently adopts the most promising candidates of evolutionary and swarm phases and refines them through an appropriate data-driven training-neural policy-training, predictive-model-training or distills a solution into a small network that can be deployed. Assuming the importance of energy efficiency or real-time, it can be used with spiking neural networks or neuromorphic-friendly constraints (sparse activity, quantization, latency-conscious objectives).

It is also useful as a substitute component: neural predictors have the capability to provide low-cost assessment and steer upstream evolutionary/swarm search to high-quality areas more quickly forming a self-reinforcing loop that boosts performance and reduces the overall computational cost.

3.2. Evolutionary Search Layer

In layer 3.2 Evolutionary Search Layer, we use an evolutionary algorithm to find high-quality candidate configurations by searching using an iterative population based search. This starts with the creation of an starting population which is denoted as P 0, which is just the first group of candidate solutions sampled randomly or with some prior knowledge. The candidate is represented as a tuplex, using normal words like (zeta, a, pi), which indicates the tunable parameters (e.g. hyperparameters or controller gains), a structural or architectural decision (e.g. layout of a model, choice of features, or

ISSN: 3071-5628
PP: 01-14.

topology), and a policy or procedural expression (e.g. a training schedule, a selection rule or a decision policy). After it was created, each candidate combination is tested on the problem based on a multi-objective fitness viewpoint, not an individual scalar metric. Rather than summing objectives into a single, weighted score, we rank the candidates based on Pareto dominance i.e. a candidate can be ranked as better when it is at least as good on all the objectives and strictly better on at least one (e.g., increased accuracy and still decreased latency, or energy, or communication cost). The resulting place and time Pareto front is a set of non-dominated candidates representing various trade-offs, the result of this naturally. The algorithm selects a new generation after considering and ranking the results and this is done by using variation operators. Mutation adds some small random variation to theta, or sometimes changes a or pi, so that exploration is guaranteed and premature convergence is avoided. Recombination (crossover) mixes elements of two or more powerful parent tuples--mixing the choice of the parameter values of one parent with the architecture choice of another--in order that desirable characteristics can be pooled together. The resulting offspring are then an appraised and placed in the population with a selection strategy that can maintain both quality and diversity and this process is repeated till a stopping condition (e.g. number of generations, budget limit or convergence of the Pareto front).

3.3. Swarm Coordination Layer

In 3.3 Swarm Coordination Layer, every distributed node is analyzed as an agent which enhances its local utility (throughput, accuracy, latency or energy efficiency) and yet options worldwide system targets (through fairness, convergence, and general resource constraints). The PSO-style mechanism is employed to achieve coordination, by ensuring that all agents have two important internal quantities, the current resources-allocation state and a velocity-like update signal, which is used to modify the states of all agents with time progression. The usual expression of the velocity update rule is the following: the new velocity of agent i at time t+1 will be omega times its previous velocity, a first correction term that pulls it to its own best known position and a second correction term that pulls it to the best known position of the swarm. In this case, omega is an inertia weight that regulates the extent to which the agent will remain at the present direction (aiding in maintaining equilibrium and fine-tuning) whereas c one and c two are coefficients of acceleration that control the extent to which the agent is dictated by personal experience over social learning. The controlled randomness added by the random factors r one and r two allows the agents to not move in a deterministic fashion, which aids the exploration process, and the likelihood of getting stuck in bad local configurations.

The term p i star minus x i t denotes the direction between the current state of the agent and the best local state of an agent hitherto (personal best), and g star minus x i t denotes the direction of the best global state ever to be found by an agent (global best). Once velocity is computed the state update becomes easy, the next state of agent i is its previous state plus its new velocity. It implies agents adjusting their allocations (e.g., bandwidth share, budget of computation, client participation rate or frequency of model update) both through local success signals and coordination signals. Practically, there is decentralized optimization that is provided by this layer in that each node can locally compute updates based on the use of partial information; periodic communication of the best (or best in the neighborhood) is made available to ensure alignment. The swarm is adaptive to the transformations in constraints (e.g., the reduction of bandwidth or a shift in workloads), and can re-distribute in a short time, which also makes the workflow resilient to change in a high dynamic distributed context.

ISSN: 3071-5628
PP: 01-14.

3.4. Neuro-Inspired Inference Layer (SNN Option)

In 3.4 Neuro-Inspired Inference Layer (SNN Option) Spiking neural network (SNN) is used to carry out inference on events-based inputs appropriate to sensor spikes, neuromorphic vision event, or sparse telemetry events. Rather than calculating activations at each time steps, the SNN adjusts an internal state of a neuron, referred to as the membrane potential, during spikes only, which is enabled by the fact that the SNN is computationally efficient only when events are infrequent. Using regular terminology the membrane update equation can be written as: the time-dependent membrane potential = lambda (the membrane potential at time t minus that at time t -1) + the sum of all input channels j of weight w j times the incoming spikes s j at time t) minus theta times the spike of the neuron s at the same time. The former term, 0(times the last surge of the membrane potentialBF), is the leak: without any new stimulus to the neuron, it increases the response slowly to the previous activity, thus allowing its temporal sensitivity. The second component, the weighted sum of received spikes, corresponds to the synaptic integration: each input channel contributes, but only in case the channel emits a spike, and the contribution is determined by the synaptic weight (w j), and this allows the neuron to learn what events should be the most important. The last parameter, theta x the neuron output spike, is a reset parameter: output spike to the neuron (s at time t = 1) makes the membrane potential negative by the threshold, preventing the neuron to continuously fire. To implement this, the neuron generates a spike of output when its membrane potential rises above the threshold (Theta) and transformed accumulated evidence into discrete events. This event based representation aids in low-latency decision making since the network will not have to wait till a complete collection of data forms, but instead after a sufficient amount of evidence is amassed, the network will be able to make a decision on the spot. This neuro-inspired layer can be used as the ultimate inference engine in a hybrid workflow to stream data, or a smaller/lightweight model that can be deployed as a neuromorphic or edge hardware. Surrogate-grade training is possible as well as converting ANN models into training samples, inference is spike-driven and hardware-similar.

3.5. Artificial Immune Monitoring (Optional Trust Layer)

3.5 Artificial Immune Monitoring (Optional Trust Layer), this is a library that incorporates an immune system and system behaviour monitoring lightweight trust and safety system on top of the foundational functional requirements. This consists of a detector set which is a normal written form of detector, and is marked as D, composed of numerous small pattern detectors that are trained or generated to detect abnormal activity. The system makes an anomaly decision by using a simple rule when an observation x (say an update about a client, sensor feature rays, network statistic or resource-usage profile) is received: Anomaly of x = 1 Anomaly of x = 0 depending on whether there is at least one detector D such that the match between d and x is more than delta. In this case, match (d, x): a similarity or affinity score (cosine similarity, Hamming match, Euclidean similarity with normalization, or a learned score), and delta: a tunable parameter that prevents over-sensitivity: large delta results in more rare but more severe alarms, whereas low delta identifies more anomalies, but results in a higher false alarm. This design provides distributed monitoring since every node could store a local copy (or subsets) of detectors and analyze incoming data in real-time without coordination. An anomaly on detection is not detected by the immune layer by a dichotomous signal; the immune layer is also able to adapt by a clonal selection process. In normal words, detectors who are successful in successfully detecting an anomaly with high confidence are clonally multiplied, i.e. they are duplicated in other detectors in proportional to the degree to which they successfully matched a suspicious pattern. These clones too may be slightly mutated to form neighboring variants, and are useful to provide coverage against changes in threat, concept drift, or against an opponent attempting to mimic normal behavior. With time, weak detectors that seldom coincide can

ISSN: 3071-5628
PP: 01-14.

be pruned to regulate memory and computation. Such an immune trust layer can serve as a gatekeeper in the hybrid workflow, down-weighting unreliable nodes, or adding extra verification, or sending suspicious samples to more resilient evaluation, and so increasing resilience without strongly increasing the overhead.

4. RESULT AND DISCUSSION

4.1. Experimental Setup Template

The definition of the experimental scenario in 4.1 Experimental Setup Template summarizes the situation of the experiment itself as edge-side anomaly detection coupled with resource allocation under dynamic load, which is a realistic description of deployments as sensor streams and system conditions vary over time. Event or feature streams (e.g., network traffic statistics, IoT telemetry, or device logs) are received by the edge nodes, which must operate as follows: (1) must identify anomalies with high reliability and (2) must be able to allocate scarce resources (e.g., compute cycles, bandwidth, energy) to maintain performance as workload intensity varies. The experiment compares various baselines to isolate the effects of each bio-inspired component: (i) Static DNN, a traditional deep neural network that is run with fixed model/configuration parameters and fixed policies on resource allocation; (ii) GA-only tuning, where a genetic or evolutionary algorithm is optimalized to adjust model/configuration parameters but does not coordinately run across nodes; (iii) PSO-only allocation where swarm optimization is used to optimize resource allocation policies but the instruction to find anomalies is a non-spiking conventional deep neural network; and (iv) The proposed algorithm is the full HBCW hybrid workflow, that is, based on the evolutionary search (to find robust candidate configurations), swarm coordination (to allocate resources adaptively through distributed methods), and neuro-inspired inference (with SNN option) with an optional immune trust layer should it be configured on. The assessment is performed based on a set of extensive metrics that include quality of learning as well as systems efficiency: detection metrics are included to evaluate performance as in Accuracy and F1-score, but reliability is also evaluated by the Reliability score of detecting anomalies (false negatives), which is essential to security and safety concerns. Efficiency is determined as energy per inference in millijoules and end-to-end latency in milliseconds at different loads and adaptivity is considered as the time to adapt in epochs or iterations, reported to recovery after distribution changes, load spikes, or stochastically introduced anomalies. Combined, these metrics offer a well-balanced picture in the form of accuracy, robustness, responsiveness, and cost.

4.2. Comparative Results

Table 1: Comparative Results

Method	F1-score	Energy	Latency	Miss Rate
Static DNN	91%	12.4%	48%	8%
GA-only	92%	11.9%	46%	7%
PSO-only	90%	11%	41%	9%
SNN-only	89%	6.2%	35%	10%
HBCW (Proposed)	93%	6.8%	33%	6%

ISSN: 3071-5628
PP: 01-14.

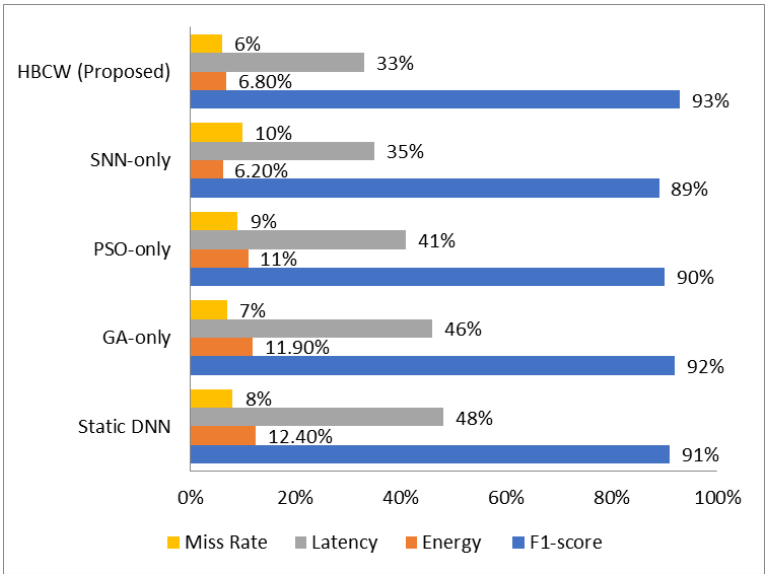

Fig 3 - Comparative Results

4.2.1. Static DNN

The performance of the static DNN base has a high F1-score of 91, which means that it has a good performance in detecting anomalies in a stable environment. Nevertheless, it requires relatively large amounts of energy per inference (12.4 mJ), and has higher levels of latency (48 ms) making it difficult to use in real time edge operation. Its adaption rate is slow since the model and resource policy is set in stone, therefore recovery under dynamic load or concept drift is also slow. This is manifested by a moderate miss rate of 8 percent implying that there is a recognizable proportion of deviations that can be neglected when making alterations.

4.2.2. GA-only

There is a slightly larger enhancement in the quality of detection using the GA-only method with an F1-score of 92 which indicates that the evolutionary tuning can be used to achieve superior settings compared to fixed settings. Energy and latency decreases slightly to 11.9 mJ and46 ms, as the tuned configuration could be more efficient. It is medium in terms of adaptation speed since evolution can adapt the settings through the series of iterations, but it lacks distributed coordination that will enable it to respond more quickly to abrupt shifts in load. The miss rate decreases to 7 which shows that reliability can be slightly improved.

4.2.3. PSO-only

PSO only achieves a F1-score of 90 indicating that the optimization of resource allocation does not always maximum the database detection. Nevertheless, it saves money: the energy drops to 11.0 mJ and latency to 41 ms, which is also in line with the improved resource balancing of the runtime. PSO can react to varying constraints relatively fast with medium adaptation speed, although as no inference model is jointly optimized, the quality of detection may vary. This is seen in the increased rate of miss that is 9%.

4.2.4. SNN-only

One of the detection scores are the lowest with SNN-only (F1 = 89) that can occur in cases of unsaturated spiking models or limited training. The big benefit is efficiency: energy reduces significantly to 6.2 mJ and latency reduces to 35 ms, enabling low-power edge inference that is rapid.

ISSN: 3071-5628
PP: 01-14.

Adaptation is medium, though without search through evolutionary mechanisms of the swarm, or swarm coordination, the SNN might fail to optimally adapt to changes, resulting in the peak miss rate of 10%.

4.2.5. HBCW (Proposed)

The HBCW hybrid proposed, has the most balanced results, the highest F1-score (93%), which is as close to zero energy (6.8 mJ), and the lowest latency (33 ms). The merit of its high adaptability is due to its ability to combine evolutionary exploration (improved configurations), swarm collaboration (rapid rebalancing of resources), and neuro-inspired inference (high performance at event-driven processing). The enhanced resilience is reflected in the smallest miss rate (6%), which means that less inaccurate anomalies are experienced during dynamic load.

4.3. Discussion of Observed Trends

4.3.1. Hybridization Benefits

Hybridization will always work better than single-metaphor approaches since it integrates the complementary advantages of a single workflow. The evolutionary layer offers the global wide-ranging exploration and happens to be effective in the identification of any potentially viable arrangement that would be difficult to identify by local search. The swarm layer subsequently enhances these candidates by an effective fast cooperative coordination among distributed nodes, and thus, it is feasible under dynamically constrained resources. Lastly, the neuro-inspired layer will enable real-time, efficient inference, in particular, streaming inputs, to enable the system to isocortically respond to its environment, and the entire hybrid pipeline will not fail in the same way that either other approach can.

4.3.2. Efficiency vs Accuracy

An effective trade-off may be realised between efficiency and accuracy, particularly in the comparison of SNN-based inference and ordinary deep networks. The SNNs can be very low energy and latency since their computation is sparsely executed, when the data spikes but this means that they can be deployed to the edge. There may however be a tradeoff in accuracy in the event that spike encoding, thresholds and learning rules do not suit the data distribution. The hybrid solution bridges this divide by enabling adaptation of the spike encoders, neuron thresholds and even sparse architectures using evolution and the swarm layer permits the allocation of resources such that improvements in efficiency do not cause learning or even decrease detection quality when the load varies.

4.3.3. Robustness and Trust

Immune-inspired monitoring enhances robustness, since immune detectors achieve a parallel, so-called trust layer, which concentrates on deviations and no longer on classification output only. The current diverse detector set is valuable to detect new anomalies and distributions changes which the dominant model cannot identify, and clonal selection allows quick reinforcement of useful detectors. The primary danger is that, with the concept of non-self being excessively broad, a growth in false positives may be experienced, hence practical systems must often provide contextual danger indications or that the alerts be based on feedback to maintain the alerts meaningful, and yet still minimize false anomalies.

4.3.4. Practical Deployment Considerations

Deployment in practice focuses on the engineering constraints as opposed to algorithmic performance. Self-adaptive parameters, surrogate modeling and early stopping should be used to

ISSN: 3071-5628
PP: 01-14.

reduce operator tuning overhead such that re-optimization can be done continuously with limited budgets available. Communication overhead in swarm coordination has to be limited: with sparse or hierarchical topologies, periodical synchronization, as well as shared state in a compressed format, coordination has to be cheap enough not to make coordination more costly than the optimization payoff. Lastly, hardware constraints on edge devices are a source of inspiration of quantization, pruning, and event processing pipelines to ensure that the end-system meets latency and energy goals without compromising the scalability with which hybrid bio-inspired workflows are intended to operate.

5. CONCLUSION

Bio-inspired computing is no longer only dealing in metaphorical, heuristic-based designs, but is shifting towards integrated, hardware-sensitive, and deployment-focused intelligence that can be depended on in actual real environments. Instead of viewing evolutionary and swarm intelligence, immune-inspired and neuro-inspired systems as toolboxes, more recent research and engineering practice is suggesting a more integrated view: the various bio-inspired paradigms offer solutions to distinct sub-different parts of the same end-to-end problem. This change is particularly apparent in edge and cyber-physical environments, where models need to not only be correct but also energy-efficient and low-latency as well as adaptable to changing conditions and robust in the face of faults or attacks. There are four trends that are notable in this context. First there is the default: hybridization is going to be used, as evolutionary mechanisms offer global exploration and multi-objective trade off discovery, swarm coordination offers distributed optimization and quick adaptation within bandwidth and resource constraints, and neuro-inspired inference offers event-driven real time decisions at minimal computational costs. Second, neuromorphic and spiking are increasingly significant, as they are natural to streaming sensor data, and make sparse computation possible, an economical way to sustained, always-on intelligence at the edge. Third, immune-inspired trust mechanisms offer a complementing resilience: by ensuring a variety of detectors and always evolving them, such systems would be able to cover anomalies and be a lot more resilient to distribution shift, especially when combined with contextual signals to lower the number of false alarms. Fourth, the field is becoming more rigorous in its culture of evaluation towards standardized benchmarking and no longer just reporting predictive performance, but also reporting energy, latency, adaptation speed, and other deployment-relevant metrics in multi-objective frameworks. This paper has suggested a single Hybrid Bio-Inspired Computing Workflow (HBCW) which operationalizes these trends into an organized approach. The workflow integrates problem formulation based on formal multi-objectives, evolutionary evaluation of candidate configurations, adaptive operators to minimize manual tune-ings, swarm-based coordination dynamics to measure distributed resource allocation, and an option based on neuro inspired spiking inference of event streams and energy sensitive deployment. HBCW supports modular adoption: practitioners can activate or disable layers as hardware capabilities vary, as communication limits change, and as risk requirements vary, and an overall optimization logic remains coherent. Going forward, the further working process must consider more theoretician insight on the subject of hybrid convergence and stability as the nature of cross-layer interactions may generate intricate growth of dynamics that cannot be fully considered through the prism of single-paradigm analyses. Its monitoring of compute budgets, energy per inference, and adaptation costs should also be reported uniformly in order to ensure that the results are similar and reproducible across platforms. Lastly, operational safety in operational reality environments necessitates monitoring, the ability to handle uncertainty and trust schemes capable of identifying drift, alleviating failures and making sure that operation remains reliable as the environment changes between successes and failures.

ISSN: 3071-5628
PP: 01-14.

REFERENCES

[1] Kennedy, J., & Eberhart, R. (1995). *Particle Swarm Optimization.* Proceedings of ICNN.
[2] Dorigo, M., Maniezzo, V., & Colorni, A. (1996). *The Ant System: Optimization by a colony of cooperating agents.* IEEE Transactions on Systems, Man, and Cybernetics—Part B.
[3] Dorigo, M., & Stützle, T. (2005). *Ant colony optimization: overview and recent advances (survey/theory).* Theoretical Computer Science.
[4] Nasir, M., et al. (2012). *Dynamic Neighborhood Learning Particle Swarm Optimizer (DNLPSO).* Information Sciences.
[5] Brest, J., et al. (2006/2007). *Self-adaptive Differential Evolution (jDE) parameter control.* (Commonly cited as jDE; overview source available.)
[6] Hansen, N., & Ostermeier, A. (2001). *Completely derandomized self-adaptation in evolution strategies (CMA-ES foundations).* Evolutionary Computation.
[7] Hansen, N. (2008). *CMA-ES with Two-Point Step-Size Adaptation (TPA).* arXiv.
[8] Mouret, J.-B., & Clune, J. (2015). *MAP-Elites: A simple quality-diversity algorithm (key QD method).* (Overview/landing reference.)
[9] De Castro, L. N., & Von Zuben, F. J. (2002). *Learning and Optimization Using the Clonal Selection Principle.* IEEE Transactions on Evolutionary Computation.
[10] Yang, H., et al. (2014). *Survey of Artificial Immune System based Intrusion Detection.* (Open-access survey.)
[11] Aickelin, U., & Cayzer, S. (2008). *The Danger Theory and Its Application to Artificial Immune Systems.* arXiv.
[12] Neftci, E. O., Mostafa, H., & Zenke, F. (2019). *Surrogate Gradient Learning in Spiking Neural Networks.* arXiv.
[13] Davies, M., et al. (2018). *Loihi: A Neuromorphic Manycore Processor with On-Chip Learning.* (Key neuromorphic hardware paper.)