Parthiv Myreddy[1]
Dr. Pragathi Yellanki[2]
M. V. S. Phani Narasimham[3]

RNN-Controlled Neuro-Nanorobots for Enhanced Drug Delivery

Journal of Electrical Systems

Abstract: Neuro-nanorobots offer a novel method for targeted drug delivery and improved medical intervention. This paper discusses advancements in chip fabrication technologies, which are aiding nanorobot capabilities by providing enhanced computational power and real-time decision-making abilities. We simulate nanorobots by integrating synthetic neural signals, enhancing the functionality and autonomy of these microscopic machines. Our design uses recurrent neural networks based on biological neural architectures, achieving improvement in the precision of temperature control during drug delivery over traditional methods. Simulation uses 4000 atoms in a lattice structure simulating the nano robots whose temperature is controlled by signals generated from spike data of 100 to 200 neurons simulator. Through simulations, we demonstrated the enhanced capabilities of these modified nanorobots. Our paper emphasizes the use of advancements in nano chip manufacturing, nano robot standards and AI to build a robust nano robot environiment.

***Keywords*:** Medical Robots and Systems; Nanomanufacturing; Neurorobotics; AI; Robotics; Autonomy for Mobility and Manipulation

1.INTRODUCTION

T The need for autonomous systems from environmental monitoring, healthcare has increased the demand to explore nanorobot technology. Traditionally, biomolecular nanorobots have played a significant role due to their compatibility with biochemical environments. But recent research in synthetic technologies are presenting new areas that could complement or even surpass traditional biological approaches in certain applications.

We integrated recurrent neural networks control signal into nanorobots, marking a significant advance in enhancing their functionality and autonomy. By mimicking biological neural architectures through RNN, we use nanorobots to perform complex tasks with remarkable precision and adaptability. Design of the RNN control spike data is based on the intricate structures and processing of the human brain to use adaptive learning and real-time decision-making, during drug delivery.

Recent innovations in chip manufacturing technologies have significantly expanded the capabilities of our synthetic nanorobots. For instance, Intel's Loihi neuromorphic chip and IBM's TrueNorth chip, along with magnetic memory research, enhance data processing capabilities. TSMC advancements in miniaturization, such as 3 nm technology, support the development of compact and efficient systems. The Graphcore IPU architecture offers enhanced computational power for complex tasks, and MIT's research on carbon nanotube-based processors provides solutions for extreme miniaturization.

In this paper, we review the advancements made by nano chip makers and nano material standards. We focus on leveraging cutting-edge neural network algorithms and chip innovations to enhance the performance and versatility of nanorobots. By simulating the superior capabilities of these synthetic systems, we aim to pave the way for their deployment in critical applications such as targeted drug delivery, precision surgery, and environmental monitoring.

The structure of this paper is as follows:

Section 1: Introduction - Highlights the increasing demand for advanced nanorobots and the potential impact of synthetic brain technology.

Section 2: Related Work - Reviews existing advancements in both biomolecular and synthetic nanorobot technologies.

Section 3: Synthetic Neuron Architecture - Discusses the design and integration of synthetic neural networks and chip technologies.

Section 4: Advancements in Synthetic Nano-Robots - Examines the contributions of key chip manufacturers and their implications for nanorobot development.

Section 5: Integrated Neuro-Nano-Robots - Exploring the broader implications, ethical considerations, and future research directions.

Section 6: Results - Summarizes the key findings and contributions of the work.

By using the neural based spike data with RNN, this research aims to contribute to the emerging field of nanorobotics, offering better results.

[1]Student, GITAM Institute of Technology, Visakhapatnam, India
[2]Professor, Department of Computer Science & Engineering, Stanley College of Engineering & Technology for Women (Autonomous), Hyderabad, India
[3]Senior Architect, Wipro Technologies, Hyderabad, 500082 India

***Corresponding Author:** M V S Phani Narasimham
*(e-mail: phani.shesha@gmail.com).

2. RELATED WORK

Recent advancements in nanotechnology and neural networks have significantly contributed to the development of intelligent nanorobots. This section reviews key contributions in the field, focusing on medical, industrial, and environmental applications.

A. Nanorobots in Medicine

Moreover, nanorobots have shown great potential in medical applications, particularly in targeted drug delivery and precision surgery. Rajendra et al. in their paper "Nanorobotics in Medicine: A Systematic Review of Advances, Challenges, and Future Prospects", reviewed advancements in nanorobotics for medical applications, including targeted drug delivery and minimally invasive surgery. This review highlights the field's growth and current challenges, such as biocompatibility, and provides insights into future research directions [1]. Bogue et al. discussed the role of nanotechnology in creating intelligent sensors, which underpin many functionalities of medical nanorobots, including diagnostics and treatment [2].

B. Neuroscience and Nanotechnology

In their paper, titled "Nanotechnology for Neuroscience: Promising Approaches for Diagnostics, Therapeutics, and Brain Activity Mapping", Kumar A et al. explored how advancements in nanotechnology are transforming neuroscience. The development of sophisticated nanomaterials has led to innovations in diagnostics and therapeutics, including drug delivery, neuroprotection, neural regeneration, and neuroimaging [3]. Silva explored the intersection of neuroscience and nanotechnology, emphasizing the progress and challenges in developing neural interfaces and nanoscale neural systems. This work highlights the importance of biocompatible materials and precise fabrication techniques for integrating synthetic brain structures into nanorobots [4].

C. Nanorobotics and Neural Networks

The "Review of Artificial Neural Network Application in Nanotechnology" explores how artificial neural networks (ANNs) are being utilized in nanotechnology, particularly in medicine and pharmaceuticals. This review summarizes various studies showing the effectiveness of ANNs in diagnosing and disease tracking, with a focus on their application in nanotechnology. This highlights the use of multilayer perceptron ANNs and presents performance metrics, including a high accuracy rate (greater than 90% in 10 out of 12 patients). This paper suggests that ongoing research will enhance the application of ANNs in nanotechnology [5]. Moritz and Krajewska et al. reviewed recent developments in nanorobotics, including the integration of artificial intelligence and neural networks. Their work demonstrated the enhanced decision-making and adaptive learning capabilities of nanorobots equipped with neural network algorithms [6]. Wang and Gao further discussed the biomedical opportunities and challenges associated with nano/microscale motors, providing insights into the practical applications and limitations of current nanorobotic technologies [7].

C. The Neural Simulation Tool (NEST):

Plesser H. E et al in their paper describe NEST simulator. It simulates large networks of simple spiking neurons. It supports a wide range of neuron and synapse models and allows for the creation of spatially structured networks using a Python-based interface. The NEST model is optimized for simulations with neurons described by a small set of differential equations using both fixed time grids and event-driven precision [8]. NEST is a specialized simulation software that models spiking neural network systems by concentrating on the dynamics, size, and overall structure of neural networks rather than the detailed morphology of individual neurons. It is particularly useful for simulating various scales of neural models, such as those that explore information processing in the visual or auditory cortex and network activity dynamics in laminar cortical networks. The NEST serves as a crucial tool in computational neuroscience for understanding complex neural processes and designing experiments [9].

D. Synthetic Brain Structures:

The integration of synthetic brain-like structures within nanorobots is inspired by biological neural architectures. This integration enables advanced functionalities such as real-time processing and adaptive learning. Recent studies, such as the work by Jon Hahnel et al. [10], address challenges in simulating neural networks with gap junctions, which require instantaneous interactions compared to chemical synapses. Their numerical algorithm based on waveform relaxation facilitates the integration of gap junctions into distributed network simulations, maintaining compatibility with delayed communication strategies. In their paper titled “Living Bits: Opportunities and Challenges for Integrating Living Microorganisms in Human–Computer Interaction”, Pataranutaporn P et al. explored how microorganisms can enhance HCIs by integrating living biological systems into technology. The potential of microbes to transform wearables, games, musical instruments, and robots is discussed. This work outlines the possibilities of combining biological and computational elements, examines human-microbe interactions, and provides guidelines for future research on microbial interfaces [11]. Shen L et al. [12] introduced the TROI, a multimedia system designed for comprehensive anatomical study of brain images. The TROI integrates tools for segmentation, training, analysis, and visualization of 3D brain data. It supports the entire process from data collection to presentation, addressing the need for a unified environment in brain imaging research

E. LAMMPS Molecular Dynamics

Aidan P. Thompson et al. [13] reviewed the LAMMPS molecular dynamics package, emphasizing its capabilities and design features that contributed to its popularity. This paper highlights the support of LAMMPS for various particle interaction models, customization options, and integration with other codes for multiphysics simulations. Recent updates include dynamic load balancing, on-the-fly visualization, and machine learning-based interatomic potentials. In their paper, Rajendran S et al. provided a comprehensive overview of the advancements and challenges in medical applications of nanorobotics. The evolution of nanorobotics, their applications in cell therapy, invasive surgery, and drug delivery, and the limitations faced, such as biocompatibility and control, are discussed. This review is based on an extensive literature search and aims to highlight current opportunities and future research directions in this field [14].

F. Adaptive Neuro-Fuzzy Inference System (ANFIS)

ANFIS combines neural networks and fuzzy logic principles to solve non-linear problems, integrating the benefits of both frameworks [15]. Ghofrane Rehaiem et al. [16] employed an artificial neural network (ANN) technique based on backpropagation to model real-time task scheduling for embedded systems. Their approach aims to minimize power consumption by optimizing the scheduling process. In their paper [17], AI-ali et al. presented an adaptive neuro-fuzzy inference system (ANFIS) for automatic COVID-19 detection from chest X-ray images. Unlike deep learning methods, the ANFIS approach, which uses texture analysis through the gray level co-occurrence matrix (GLCM), is effective even with small datasets. The proposed method demonstrated comparable performance accuracy to advanced deep learning techniques while handling the complexities of COVID-19 detection.

G. Recurrent Neural Networks (RNNs) and Network Architecture

Olah [18] provided an accessible introduction to long short-term memory (LSTM) networks, a type of RNN known for learning long-term dependencies and handling sequential data. This foundational paper is crucial for understanding advanced applications in network and data processing. Building on this, M V S Phani Narasimham and Y V Sai Pragathi [19] explore the implementation of 5G/6G standards using cloud-native technologies such as Kubernetes. Their study proposed algorithms to enhance user parameters for advanced networks, including automated guided vehicles and car-as-a-cloud services. This paper introduces a modified edge architecture that integrates RNN technologies into the open radio access network (O-RAN) framework, aiming to optimize performance experiences for end users. Babichev S et al. [20] compared different RNN architectures for classifying gene expression data and reported that a single-layer GRU network with 75 neurons achieved the highest accuracy (97.2%). The study also highlights the slight advantage of the GRU model over the CNN and LSTM models in terms of classification performance.

H. NanoRobots

W Wang et al. discussed the potential of swimming nanorobots in biomedical applications, focusing on their ability to mechanically open cell membranes for drug delivery. This approach addresses challenges such as biocompatibility and biodegradability, emphasizing the need for optimized designs to improve the efficiency of overcoming biological barriers [21]. W Si [22] described a nanoparticle-DNA assembled nanorobot powered by electroosmosis and electrophoresis that was designed to move along a graphene membrane surface. This nanorobot, controlled by tuning the surface charge density of nanopores, has potential applications in cargo delivery and nanomanipulation. In their paper, M. Hu et al. [23] discussed the potential of micro/nanorobots for targeted drug delivery, focusing on their ability to autonomously move and deliver drugs to hard-to-reach areas using exogenous and endogenous power sources such as magnetic fields, light, and chemical reactions. This highlights the challenges in developing these systems, particularly the need for additional in vivo research, and discusses the prospects of micro/nanorobots in medical applications. In their paper, Farhadian et al. [24] wrote, "Molecular dynamics simulation of drug delivery across the cell membrane by applying gold nanoparticle carriers" and used molecular dynamics simulations to investigate how gold nanoparticles (AuNPs) affect the delivery of hydrophobic (flutamide) and hydrophilic (glutathione) drugs across a cell membrane. The results showed that AuNPs alter drug interactions with the membrane, potentially enhancing delivery effectiveness and reducing nanotoxicity [25].

3. SYNTHETIC NEURON ARCHITECTURE

Recent biological neurons transmit information through electrical impulses called spikes or action potentials. These spikes travel down a long fiber called an axon and can trigger spikes in other neurons through connections called synapses.

Dendrites: Received signals from other neurons.

Cell body (Soma): The soma contains the nucleus and integrates incoming signals.

Axon: Transmits electrical impulses away from the cell body.

Synapses: axons are connected to other neurons or muscle cells to transmit signals.

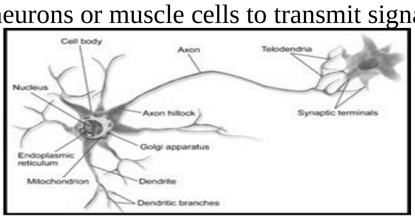

Figure 1: Biological Neurons

In their paper, F. J. Schmitt [26] compared two simulation tools, NEST and GeNN, for spiking neural networks (SNNs). GeNN uses a GPU architecture to increase simulation speed, supporting networks of up to 3.5 million neurons, while NEST parallelizes simulations across CPU cores. The study provides a benchmark for performance and discusses the trade-offs between the tools in terms of cost, speed, and network size. A. Pathak [27] introduced a multiscale computational model of the corticostriatal circuit that generates physiological outputs such as spiking and synaptic changes. It successfully models decision-making and working memory processes, predicting a previously unknown neural code for upcoming erroneous behaviors, which was later confirmed in empirical data.

Nanorobots, also known as nanobots or nanomachines, are engineered at the nanometer scale to interact with materials at the molecular or atomic level. These devices operate within a size range of 0.1 to 10 micrometers and are constructed using nanoscale or molecular components. Nanorobots are often built from advanced molecular building blocks such as buckminsterfullerenes (buckyballs) and carbon nanotubes. Carbon nanotubes, in particular, are cylindrical nanostructures known for their exceptional mechanical, electrical, and thermal properties. The strength, flexibility, and conductivity of these materials make them ideal for nanorobotics, where they can serve as both structural elements and functional components, providing stability and precision at the nanoscale. S.S. Andhari [28] discussed magnetic nanobots designed for targeted cancer therapy utilizing carbon nanotubes and Fe_3O_4 nanoparticles to autonomously navigate biological environments. These nanobots deliver doxorubicin directly to tumor cells, offering improved penetration and drug release in response to the tumor's pH, enhancing therapeutic efficacy while minimizing side effects

Some nanorobots are equipped with molecular switches or motors to perform specific tasks. For example, a single-molecule car developed at Rice University uses buckyballs as wheels and operates through changes in environmental temperature. This car is controlled by advanced techniques such as scanning tunnelling microscopy, demonstrating how functional units enhance the versatility and capabilities of nanorobots.

Additionally, nanorobots can integrate sophisticated sensors designed to detect and analyze chemical compounds at the molecular level. These sensors can measure the concentrations of specific molecules or identify toxic substances with high precision, making nanorobots valuable in fields such as medical diagnostics and environmental monitoring, where accurate detection and measurement are crucial.

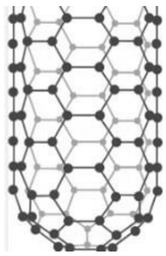

Figure 2: Nanorobot Composition

Advances in nanotechnology and neural networks have significantly contributed to the development of intelligent nanorobots. Nanorobots are also built from advanced molecular building blocks such as buckminsterfullerenes, as shown in Figure 2, commonly known as buckyballs. Buckminsterfullerenes are a type of fullerene that are composed entirely of carbon atoms arranged in a hollow, spherical, or elliptical shape. Named after the architect and inventor Richard Buckminster Fuller, these molecules resemble the geodesic domes he designed. D.K. Patel [29] discussed the synthesis methods of carbon nanotubes (CNTs), such as arc discharge, chemical vapor deposition, and laser vaporization. These findings highlight the potential of CNTs in agriculture and biotechnology, including their applications in promoting plant growth, wastewater treatment, and electrochemical platforms. The high surface area, mechanical strength, and ability to functionalize CNTs make them promising for these fields.

In their paper, Shwetha K.P. [30] explored advancements in nanorobotics for biomedicine, focusing on materials such as carbon nanotubes and Janus particles for drug delivery and diagnostics. The review delves into the design, propulsion mechanisms, and biomedical applications of these materials, particularly in cancer treatment, while also addressing challenges related to biocompatibility and material limitations. Q. Liu [31] reviewed the orthopedic applications of Buckminsterfullerene C60 and its derivatives, highlighting their antioxidant properties and potential in treating conditions such as cartilage degeneration and bone disorders. However, further research on the pharmacokinetics and toxicity of these compounds for clinical use is needed.

4. ADVANCEMENTS IN NANO ROBOTS FROM CHIP MAKERS

Advancements in chip manufacturing technologies are revolutionizing the field of nanorobotics. Traditionally, biomolecular nanorobots have led to their ability to integrate with biological systems due to their ability to interact with biochemical environments. However, recent breakthroughs in synthetic technologies are emerging as compelling alternatives that could complement or even replace biological systems in certain applications.

The paper "Nanowire-Based Sensors for Biological and Medical Applications" by Wang et al. reviews the development and use of silicon nanowires, which are primarily composed of silicon, in creating highly sensitive sensors for biological and medical applications. These nanowires are valued for their electrical controllability and chemically adaptable surfaces, making them suitable for detecting viruses, biomarkers, and DNA, as well as for drug discovery. This paper also discusses recent advancements in nanowire technology, such as improved reusability, sensitivity in challenging environments, and long-term stability, positioning silicon nanowires as crucial components in the future of medical diagnostics [32]. In their paper, G. Zhang [33] reviews the advances in using 1D nanowires for detecting chemical species and biologically relevant molecules. This highlights the superior sensing performance, stability, and low power consumption of these materials, which make them ideal for biological and chemical sensing applications. This paper also addresses challenges such as reproducibility, selectivity, and the development of more advanced nanowire-based sensors.

Intel Loihi neuromorphic chips and advanced research into quantum dots are driving brain-like computing and real-time decision-making. The Loihi chip mimics the neural networks of the human brain, enabling synthetic nanorobots to perform complex decision-making tasks in real time while consuming minimal power. Integrating Intel's Loihi neuromorphic chip technology into molecular nanorobots presents significant advantages and challenges. The Loihi chip's brain-inspired design offers enhanced computational capabilities, enabling nanorobots to perform complex tasks with high efficiency and real-time adaptability. This could revolutionize applications in medicine, such as targeted drug delivery and real-time diagnostics, by providing advanced decision-making abilities and responsiveness to environmental stimuli. However, several challenges need to be addressed for successful integration. Miniaturizing silicon chips to fit within nanoscale robots is complex, and ensuring a consistent power supply while avoiding harm to biological tissues is critical. Biocompatibility is another major concern, as silicon materials must not provoke immune responses or cause tissue damage. Additionally, managing the heat generated by chips is essential for avoiding disruption of biological processes. The integration of silicon technology with nanorobot functionality requires precise engineering and advanced materials science [34]. Intel Lohi 2 in [35] enhances brain-inspired computing, achieving up to a 10x improvement in performance over that of its predecessor. It uses a new open-source framework called Lava, which was designed for community-driven neuromorphic computing. Loihi 2 excels in real-time decision-making while using minimal power, making it suitable for advanced AI applications, including robotics and healthcare.

IBM's TrueNorth chip and innovations in magnetic memory have contributed to brain-inspired computing and advanced data storage solutions. IBM's TrueNorth chip can be integrated into molecular nanorobots to greatly enhance computational ability and efficiency. By embedding TrueNorth's neuromorphic architecture into the control systems of molecular nanorobots, researchers can provide these robots with advanced processing capabilities that mimic the brain's neural networks. This integration allows nanorobots to handle complex data processing tasks at high speed and with low power consumption. TrueNorth's integration involves incorporating its specialized neurosynaptic cores directly into the nanorobot's system, enabling real-time decision-making and adaptive responses. The chip's efficient memory management system supports extensive data storage and retrieval within the limited space of molecular robots [36]. D. V. Christensen et al. [37] explored the current state of neuromorphic computing. It addresses challenges related to traditional von Neumann architectures and highlights the potential of neuromorphic systems to reduce power consumption while offering advanced capabilities such as learning and complex data processing. The roadmap covers key areas such as materials, devices, circuits, and algorithms, in addition to future opportunities and challenges in neuromorphic technology development.

TSMC advancements in chip miniaturization, including 3 nm technology, are critical for developing compact and efficient nanorobot systems. This cutting-edge technology allows for the integration of highly powerful and miniature chips into nanorobots, which significantly enhances their computational power and functionality without increasing their size. By embedding advanced sensors, processors, and communication modules into these small-scale chips, researchers can greatly improve the ability of nanorobots to perform complex tasks such as real-time data processing and intricate biochemical reactions. The 3 nm process allows more transistors to be packed into a smaller space, enhancing the computational power and functionality of nanorobots without increasing their size. This miniaturization facilitates the embedding of advanced sensors, processors, and communication modules within nanorobots, enhancing their ability to perform complex tasks. For instance, nanorobots equipped with 3 nm TSMC chips can execute real-time data processing, control intricate biochemical reactions, and communicate efficiently with external systems [38]. TSMC [39] presented advancements in semiconductor manufacturing. It offers enhanced power efficiency, higher performance, and improved transistor density, making it ideal for integration into nanorobots. These chips allow nanorobots to perform real-time data processing, control biochemical reactions, and communicate with external systems efficiently, enabling complex tasks such as advanced medical diagnostics and drug delivery while maintaining a compact size.

NVIDIA is a leader in AI and GPU technology, and innovations in these areas had significant impacts on various fields, including advanced computing and nanotechnology. They are known for their contributions to high-performance computing, deep learning, and artificial intelligence (AI) research, which can support advancements in nanotechnology and related applications [40]. U. Iqbal in [41] examined GPU-accelerated SBCs, focusing on their role in edge computing environments for computer vision (CV) applications. Hardware improvements, such as NVIDIA's GPU integration, which enhances computational efficiency and real-time processing capabilities for tasks such as urban traffic management and smart cities, are discussed. The paper also highlights software developments in frameworks such as CUDA and provides a comparative analysis of SBC technologies.

Graphcore Intelligence Processing Unit (IPU) architecture introduces new possibilities for adapting control systems in synthetic nanorobots. This architecture provides enhanced computational power, which is essential for executing complex tasks with precision and real-time feedback [42]. In [43], H. Peng evaluated AI/ML accelerators, focusing on the architectural and computational differences between the Graphcore IPU, Sambanova RDU, and NVIDIA/AMD GPUs. By benchmarking these platforms on various AI workloads, insights into their performance and energy efficiency can be obtained, highlighting the potential advantages of data-flow architectures such as the IPU for executing highly parallel tasks, such as those required for AI/ML applications.

Research by MIT on carbon nanotube-based processors offers solutions for extreme miniaturization, enabling the creation of ultrasmall yet powerful synthetic nanorobots. These advanced processors facilitate interactions with biochemical environments at unprecedented scales, potentially surpassing the capabilities of traditional biomolecular robots [44].

Radisys edge servers significantly enhance the functionality and integration of nanorobots within IoT frameworks by providing low-latency processing, AI-driven autonomy, and seamless connectivity. These capabilities are essential for both real-time applications and ongoing research in fields such as healthcare, where precision, scalability, and efficient data handling are critical [45]. A. Bala [46] reviews the integration of AI with edge computing in machine maintenance, focusing on predictive and prescriptive methods. This highlights how edge computing enhances real-time diagnostics, reduces latency, and improves machine health monitoring by processing data closer to the source.

The integration of these cutting-edge chip technologies with synthetic neural networks signifies a major advancement toward developing more versatile and capable nanorobots. While biomolecular nanorobots have laid a strong foundation, the enhanced performance and adaptability of synthetic chip-based systems underscore their growing importance in the field of nanorobotics.

Industry Standards: The ISO 80004-1:2023 standard provides a comprehensive vocabulary for nanotechnologies, offering harmonized definitions and terminology to ensure consistent communication and understanding across various fields involved in nanotechnology. This standard merge previous editions and revises key terms such as "nanostructure," facilitating the development, application, and dissemination of nanotechnologies in industries ranging from health to manufacturing [47]. The ISO standards [48], [49] related to nanotechnologies, particularly those under ISO/TC 229, cover a broad range of topics, including terminology, toxicity assessment, and the characterization of nanomaterials. These standards are essential for ensuring the safe development and application of nanotechnologies across various industries. Specific standards such as ISO/TS 4958:2024 (Nanotechnologies — Vocabulary — Liposomes) and ISO/TS 5094:2023 (Assessment of peroxidase-like activity of metal nanoparticles) provide guidelines for specific applications and evaluations, facilitating consistency and safety in nanotechnology practice.

5. INTEGRATED NEURAO NANO ROBOTS

The integration of synthetic neural networks into nanorobots signifies groundbreaking advancements in nanotechnology. This section delves into the architectural design and implementation of neuro/nanorobots, emphasizing their real-time processing capabilities and potential applications.

A. REAL-TIME CLOUD ARCHITECTURE

The incorporation of real-time cloud architectures into neuro/nanorobots significantly enhances their functionality and performance. By leveraging cloud computing, these nanorobots can offload intensive computations and data storage to remote servers, thus enabling real-time processing and decision-making even with limited onboard resources. The core components of this architecture include the following:

Cloud Servers: These high-performance servers manage the heavy computational tasks and data storage necessary for processing the information gathered by the nanorobots.

Communication Interface: A robust and secure communication interface ensures seamless data exchange between the nanorobots and the cloud servers.

Real-Time Processing Framework: An advanced framework processes data in real time, allowing nanorobots to make quick, informed decisions. With this architecture, neuro/nanorobots can perform complex tasks such as adaptive learning, environmental monitoring, and targeted medical interventions with enhanced efficiency and precision. A.W. Donnatti [50] reviews the impact of virtual machine (VM) state changes on the network management domain in OpenStack cloud environments. The authors conducted extensive tests across different operating systems and VM tasks to characterize the administrative network traffic. This study provides valuable insights into how user-generated tasks (e.g., VM creation, deletion, suspension) can affect cloud network performance and how administrators can better manage resources by predicting network traffic through linear regression models. These findings can aid in optimizing bandwidth and ensuring the efficient design of OpenStack cloud infrastructures.

B. NANO ROBOTS USING NEURAL SIGNALS

The Neural Simulation Tool (NEST) is a powerful simulation tool for spiking neural network models. This approach plays a crucial role in simulating and optimizing synthetic neural networks within nanorobots, thereby enhancing their functionality and performance. The key aspects of utilizing NEST include the following:

Optimization of Neural Network Architectures: Architecture of synthetic neural networks to achieve optimal performance for specific tasks.

Simulation of Real-World Scenarios: The NEST allows for modeling and simulating real-world environments, enabling the testing and validation of neuro-nanorobot performance. Enhancement of adaptive learning: Advanced learning algorithms can be implemented to allow neuro/nanorobots to adapt to changing environments and improve their performance over time.

Using NEST in developing neuro/nanorobots ensures their autonomous and efficient operation, making them suitable for various applications in medical, industrial, and environmental fields. M.D. Pham in [51] reviews the role of biologically plausible spiking neural networks (SNNs) in neuromorphic systems, emphasizing their application in neurorobotics and the balance between computational efficiency and biological accuracy. This highlights how SNNs contribute to embodied cognition in robotics, especially through neuromorphic hardware

The simulation in Figure 3 begins with running an NEST simulation to model neural activity, generating spike data. These data are subsequently used to construct and train a RNN model, which enables it to recognize patterns and predict outcomes. The trained RNN model generates a control signal, which serves as an input for running a LAMMPS simulation to model physical processes. The output from the LAMMPS simulation is analyzed to assess the accuracy and effectiveness of the model. The results are then compared with expected outcomes or benchmarks, and comparison graphs are plotted to visualize the findings. This integrated approach combines computational neuroscience with molecular dynamics, allowing for detailed analysis and comparison of simulation results to validate the models used.

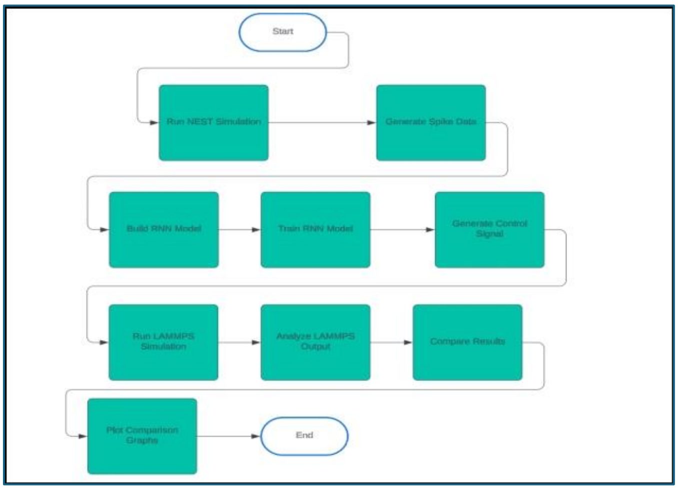

Figure 3: Flowchart of Nano Robots for Drug Delivery Efficiency Using an RNN-Generated Control Signal

The lower section describes the modules in the flow chart in Figure 3.

NEST Simulation: Start by running a simulation with NEST to model spiking neural networks. This simulation generates spike data, which capture the neural activity within the network.

Data Utilization: The spike data from NEST were used to build and train an RNN model. The RNN is trained to recognize patterns in the data and predict control signals based on these patterns.

Control Signal Generation: Once the RNN is trained, it generates control signals that guide the behavior of the nanorobots. These control signals are crucial for directing nanorobots toward their target, optimizing their movement and efficiency.

LAMMPS Simulation: The RNN-generated control signals are fed into a LAMMPS simulation. LAMMPS models the physical processes and interactions of nanorobots within a simulation environment. This step evaluates how well the nanorobots follow the control signals and achieve their objectives.

C. SIMULATION ALGORITHM

Directional Influence Calculation

Input: positions (list of nanorobot positions), source_position (position of drug source)

Output: influences (list of calculated influence values based on distance)

Steps:

1. Initialize an empty list influences.
2. For each position,
   - Calculate the Euclidean distance between the position and source_position.
   - The influence is computed as 1/(1 + distance).
3. The influence is applied to the influence list.
4. Return the influence list.

Drug Effect Application

Input: neurons (list of neuron IDs), drug_effect (drug effect factor), target_precision (precision of target), positions (list of nanorobot positions), source position (position of drug source)

Output: Adjusted neurons with new tau_m values

Steps:

1. A neuron collection was created from the neuron list.
2. Directional influence is calculated using the calculate_directional_influence function.
3. For each neuron and its corresponding influence:
   - The tau_m parameter was adjusted using the formula 20.0 * drug_effect * (1.0 + target_precision * influence).
   - Update the neuron's tau_m value with the new calculated value.

NEST Simulation with Drug Effects

- **Input**: drug_effect (default: 1.0), target_precision (default: 0.5)
- **Output**: spike_times (array of spike times from the simulation)

Steps:

1. Reset the NEST kernel.
2. Random positions were generated for neurons, and the source, position was defined for the drug.
3. Neurons were created, and drug effects were applied using the apply_drug_effects function.
4. Create a spike generator and a voltmeter.

- The spike generator was connected to the neurons, and the voltmeter was connected to the neurons.
- The simulation lasted for 200 milliseconds.
- The spike times were retrieved and returned from the voltmeter.

RNN Model for Control Signals

Build the RNN Model

Input: input_shape (shape of the input data)

Output: Compiled RNN model

Steps:

1. Initialize a sequential model.
2. Two LSTM layers with 50 units each were added.
3. A dense layer with 1 unit was added.
4. The model was compared using the Adam optimizer and mean squared error loss.
5. Return the compiled model.

Train the RNN Model:

Input: model (RNN model), spike_data (data used to train the model)

Output: Trained RNN model

Steps:

1. Reshape the spike_data to fit the RNN input shape.
2. Define y as the mean of the differences in spike_data.
3. The model is trained with X and y for 10 epochs.

Generate the Control Signal:

Input: Model (trained RNN model) and spike_data (data used to generate the control signal)

Output: control_signal (generated control signal)

Steps:

1. Reshape the spike_data to fit the RNN input shape.
2. The control signal is predicted using the model.
3. Return the predicted control signal.

LAMMPS Simulation Function

Input: None

Output: None (runs LAMMPS simulation)

Steps:

1. Initialize LAMMPS.
2. Define A LAMMPS script with simulation parameters.
3. Execute the LAMMPS script using lmp.commands_string().
4. Close the LAMMPS instance

D. Execution Snapshot

W. Ponghiran [54] converted the computationally intensive fully connected layers of traditional LSTMs into low-power spiking neural networks (SNNs), and the authors significantly reduced the power consumption without compromising the accuracy. The hybrid model achieves more than 55x improvement in energy efficiency for image recognition tasks on Intel's Loihi processor, making it ideal for energy-constrained platforms. The execution snapshots of the simulation process, including time logs and performance metrics, are provided below. A sample snapshot is used for the LAMMPS simulation and RNN model training:

NEST Simulation Logs

Y. Nasir in [55] reviewed the performance of the Genn and Spike Excel packages in single-threaded frames, while the NEST algorithm achieved better results with multithreaded CPU configurations. We used 64 core Intel icelake CPU for simulation.

```
Aug 11 19:51:36 NodeManager::prepare_nodes [Info]:
Preparing 12 nodes for simulation.
Aug 11 19:51:36 SimulationManager::start_updating_ [Info]:
Number of local nodes: 12
Simulation time (ms): 200
Number of OpenMP threads: 1
Not using MPI
Aug 11 19:51:36 SimulationManager::run [Info]:
Simulation finished.
```

RNN Simulation Logs

Kjaerrain [56] reviews how each epoch represents a complete pass through the dataset during training, with weights updated at each iteration based on error backpropagation. The authors also describe how LSTMs handle sequential dependencies, emphasizing the importance of tuning parameters across multiple epochs to improve loss reduction, as shown in their experiments on solar power production and natural language processing.

```
Epoch 1/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 4s 4s/step - loss: 0.3143
Epoch 2/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step - loss: 0.1176
Epoch 3/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step - loss: 0.0186
Epoch 4/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step - loss: 0.0021
Epoch 5/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step - loss: 0.0300
Epoch 6/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s/step - loss: 0.0573
Epoch 7/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step - loss: 0.0670
Epoch 8/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step - loss: 0.0600
Epoch 9/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step - loss: 0.0434
Epoch 10/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step - loss: 0.0249
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 384ms/step
```

lammps Simulation Logs

LAMMPS course in [57] describes users how to configure the system environment, submit jobs using the Slurm scheduler, and improve simulation performance. Key topics include managing multicore processing, MPI task division, and performance profiling using timing breakdowns of different simulation components [57]. The topic *of running LAMMPS on HPC* systems teaches users how to optimize LAMMPS for high-performance computing [58].

In [59] G Giri et al explores the challenges in design of nano robot sensors, navigating, actuators, powering and transmission. H.Zhou et al uses magnetic fields for actuation of nanorobots. Fuel free control is achieved with embedding ferro particle into nano robots [60]. M Urso et al in their paper "Smart micro- and nanorobots for water purification" have used magnetic nano particles, catalytic materials using platinum and light responsive materials like titanium dioxide. Nanorobots made of this material are used in removing microplastics, heavy metals and pathogens from water [61]. In [62] X. Kong et al has used Fe3O4, multi walled carbon nanotubes, biodegradable polymers for easy navigation.

A snapshot of the nanorobot lammps simulation logs is shown below.

```
The OMP_NUM_THREADS environment was not used. Defaulting to 1 thread. (src/comm.cpp:98)
using 1 OpenMP thread(s) per MPI task
Lattice spacing in the x,y,z direction = 1.6795962 1.6795962 1.6795962
Created orthogonal box = (0 0 0) to (16.795962 16.795962 16.795962)
1 by 1 by 1 MPI processor grid
Created 4000 atoms
using lattice units in orthogonal box = (0 0 0) to (16.795962 16.795962 16.795962)
create_atoms CPU = 0.000 seconds
Generated 0 of 0 mixed pair_coeff terms from geometric mixing rule
```

```
Neighbor list info...
update: every = 20 steps, delay = 0 steps, check = no
max neighbors/atom: 2000, page size: 100000
master list distance cutoff = 2.8
ghost atom cutoff = 2.8
binsize = 1.4, bins = 12 12 12
1 neighbor lists, perpetual/occasional/extra = 1 0 0
(1) pair lj/cut, perpetual
attributes: half, newton on
pair build: half/bin/atomonly/newton
stencil: half/bin/3d
bin: standard
Setting up Verlet run ...
Unit style    : lj
Current step  : 0
Time step     : 0.005
Per MPI rank memory allocation (min/avg/max) = 3.222 | 3.222 | 3.222 Mbytes
Step          Temp          Press          TotEng
0   1.44          -5.0199732     -4.6139081
50   0.7438899      0.30822425    -4.6215047
100   0.75798698     0.2092058     -4.6214464
150   0.75372058     0.22865363    -4.6207481
200   0.75473375     0.25644226    -4.6182748
250   0.76475493     0.22169717    -4.6153306
Loop time of 0.578915 on 1 procs for 250 steps with 4000 atoms

Performance: 186555.834 tau/day, 431.842 timesteps/s, 1.727 Matom-step/s
99.4% CPU use with 1 MPI tasks x 1 OpenMP threads

MPI task timing breakdown:
Section |  min time  |  avg time  |  max time  |%varavg| %total
---------------------------------------------------------------
Pair    | 0.48156    | 0.48156    | 0.48156    |   0.0 | 83.18
Neigh   | 0.070976   | 0.070976   | 0.070976   |   0.0 | 12.26
Comm    | 0.0078094  | 0.0078094  | 0.0078094  |   0.0 |  1.35
Output  | 0.00034296 | 0.00034296 | 0.00034296 |   0.0 |  0.06
Modify  | 0.016538   | 0.016538   | 0.016538   |   0.0 |  2.86
Other   |            | 0.001689   |            |       |  0.29

Nlocal:           4000 ave        4000 max        4000 min
Histogram: 1 0 0 0 0 0 0 0 0 0
Nghost:           5723 ave        5723 max        5723 min
Histogram: 1 0 0 0 0 0 0 0 0 0
Neighs:         149901 ave      149901 max      149901 min
Histogram: 1 0 0 0 0 0 0 0 0 0

Total # of neighbors = 149901
Ave neighs/atom = 37.47525
Neighbor list builds = 12
Dangerous builds not checked
Total wall time: 0:00:00
LAMMPS (27 Jun 2024)
Total wall time: 0:00:00
```

6. RESULTS

Neuro/nanobots accurately direct drugs to specific target sites within the body, reducing side effects and increasing therapeutic efficacy. In LAMMPS' Lennard–Jones (LJ) units, energy is expressed relative to the depth of the Lennard–Jones potential (ε), and temperature is normalized to ε/k_B.

$E_{Joules}=E_{LJ}\times\epsilon\times1.602\times10^{-19}$J/eV

E_{LJ} is the energy in the LJ unit

ϵ is the depth of the Lennard–Jones potential in eV

The conversion factor from eV to joules is 1.602×10^{-19}J/eV.

In LJ units, the temperature is normalized by ϵ/k_B, where kB is the Boltzmann constant. The formula for converting the temperature from LJ units to Kelvin is:

$T_K = T_{LJ} \times \epsilon / k_B$

where:

- T_{LJ} is the temperature in the LJ units,
- ϵ is the potential depth in Joules,
- $k_B = 1.38 \times 10^{-23}$ J/K.

If ϵ is given in eV, multiply by 1.602×10^{-19} times to convert to Joules. Thus:

$T_K = T_{LJ} \times \epsilon \times 1.602 \times 10^{-19} / 1.38 \times 10^{-23}$

Figure 4 illustrates the temperature regulation of nanorobots in a LAMMPS simulation, comparing the results when controlled by an RNN with a precision of 0.5 against a default control signal. The RNN control signal achieves more stable and lower temperature values, indicating its superior ability to manage the thermal conditions of the nanorobot.

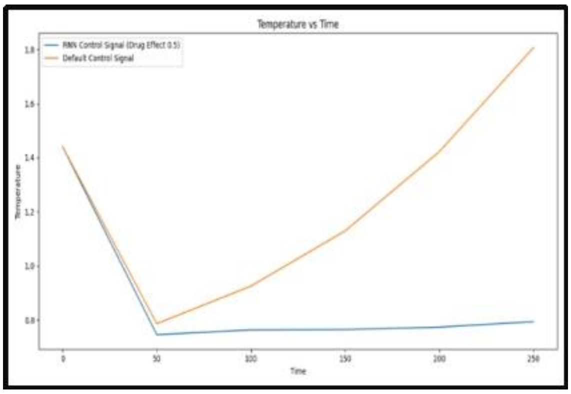

Figure 4: Lammps' nanorobot temperature comparison with the RNN control signal with a precision of 0.5

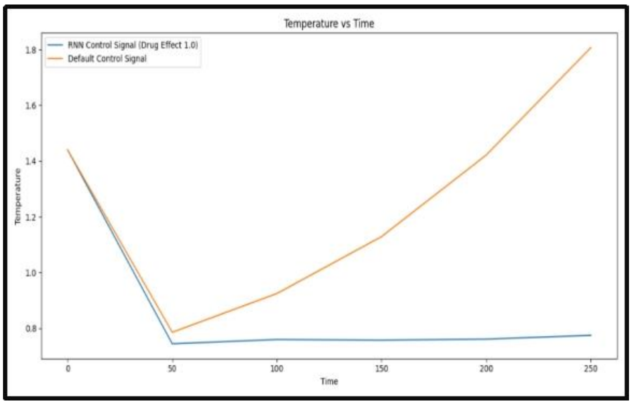

Figure 5: Lammps nanorobots temperature using an RNN control signal with precision

Figure 5 shows the temperature behavior of the nanorobots under RNN control signal with a precision of 1.0. A higher precision allows for even finer temperature control, further reducing fluctuations and maintaining a lower overall temperature, which is crucial for ensuring the stability and effectiveness of nanorobots.

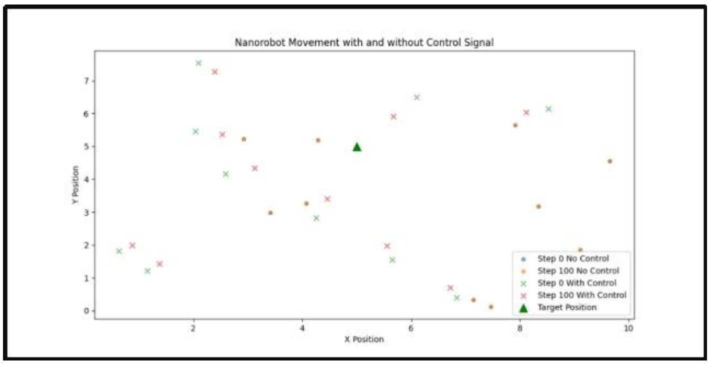

Figure 6: Nanorobots using RNN neural signals for drug target delivery

Figure 6 depicts the enhanced precision and accuracy of drug delivery using nanorobots guided by RNN-generated neural signals. The integration of synthetic brain capabilities into nanorobots significantly improves their ability to target specific sites, minimize side effects and boost therapeutic outcomes.

Table 1. Comparison of Temperatures in lammps simulation using an RNN Neural Control Signal

Time	Temp with RNN Control Signal	Temp with Default Control Signal	Difference
0	1.44	1.44	0
50	0.745919	0.785567	-0.039649
100	0.766101	0.924695	-0.158594
150	0.771977	1.128814	-0.356837
200	0.787189	1.421567	-0.634378
250	0.815467	1.806682	-0.991215

Table 2. Energy comparison with and without RNN

Time Step	Total Energy (LJ)	Total Energy (No RNN)
0	-4.6139081	-4.6139081
50	-4.6184628	-4.5590048
100	-4.6092786	-4.3714465
150	-4.5933708	-4.0582485
200	-4.5696039	-3.6182753
250	-4.5392824	-3.0528315

In Table 1, the temperatures obtained using the RNN control signal remain lower than those obtained using the default signal, with the differences increasing over time. This indicates that the RNN control achieves better temperature regulation. In Table 2, the total energy values obtained using RNN control are lower across all time steps than those obtained using default control, suggesting RNN control not only maintains a more stable temperature but also reduces the system's total energy. This improved energy efficiency highlights the effectiveness of RNN-based control in optimizing simulation conditions.

7. CONCLUSION

We reviewed the latest efforts by nano chip makers and the standards for implementing an efficient nanorobot environment. We simulated neural nanorobots that represent a significant advancement over traditional designs, offering enhanced precision in drug delivery and superior directional control. By integrating RNNs into nanorobot designs, these innovations pave the way for effective and targeted therapeutic applications. These neural nanorobots achieve enhanced control and precision, suggesting substantial improvements in areas such as targeted drug delivery, precision surgery, and environmental monitoring, marking a noteworthy step forward in the field of nanorobotics and precision medicine.
Future work could involve exploring the integration of 3D directional control mechanisms within these nanorobots to further enhance their spatial navigation capabilities and adaptability in complex environments. Additionally, incorporating cutting-edge advancements from chip manufacturers using MIT's carbon nanotube-based processors will improve sensing capabilities of nano robots while interreacting with biological fluids. Future work will involve utilizing open stack edge clouds, offloading to nearby cloud servers, reducing latency and enabling seamless integration of nanorobots in critical applications such as healthcare, environmental monitoring, and advanced industrial wastage control.

Funding Declaration This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgments

We are grateful to GITAM Biotechnology H.O.D., Sarat babu I., for his assistance in helping with the complex biotechnology concepts. We are thankful to the managers Srikanth G, Vishal K, Deepak D, Rupesh K, and Prakash S for their unwavering support throughout this research. We are grateful to Stanley College of Engineering & Technology for Women, Principal Dr. Satya Prasad Lanka, and College Management for their encouragement in exploring new research areas.

8. REFERENCES

[1] Rajendran S, Sundararajan P, Awasthi A & Rajendran S, "Nanorobotics in medicine: A systematic review of advances, challenges, and future prospects with a focus on cell therapy, invasive surgery, and drug delivery". Precision Nanomedicine,2024, https://doi.org/10.33218/001c.92224.
[2] Bogue, R, "The role of nanotechnology in the development of intelligent sensors". Sensor Review, 35(3), 219-223,2015, DOI - 10.1108/02602280410545362.

[3] Kumar A, Tan A, Wong J, Spagnoli J.C, Lam J, Blevins B.D, Thorne L, Ashkan K, Xie J and Liu, H,

[4] Silva, G. A. Neuroscience nanotechnology: progress, opportunities, and challenges. Nature Reviews Neuroscience, 12(3), 151-161, 2011.

[5] Jakupović, A., Kovačević, Ž., Gurbeta, L., & Badnjević, A, "Review of artificial neural network application in nanotechnology. Proceedings of the 7th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro", 2018, IEEE. https://doi.org/10.1109/MECO.2018.8406006

[6] Moritz, M., & Krajewska, A. Nanorobots: recent developments and future perspectives. Materials Science and Engineering: C, 109, 110651, 2020.

[7] Wang J and Gao, W, "Nano/microscale motors: biomedical opportunities and challenges", *ACS Nano*, 6(7), 5745-5751,2012.

[8] Plesser, H. E., Diesmann M., Gewaltig M O and Morrison, A, " NEST: The Neural Simulation Tool. In Encyclopedia of Computational Neuroscience" (pp. 1849–1852). Springer, 2015. https://doi.org/10.1007/978-1-4614-6435-8_306.

[9] NEST (Neural Simulation Tool). GitHub - nest/nest-simulator: The NEST simulator.

[10] Jan Hahne, Moritz Helias, Susanne Kunkel, Jun Igarashi, Matthias Bolten, Andreas Frommer and Markus Diesmann, "A unified framework for spiking and gap-junction interactions in distributed neuronal network simulations", DOI: 10.3389/fninf.2015.00022, frontiers in Neuroinformatics.

[11] Pataranutaporn P., Vujic A., Kong D. S, Maes P, and Sra, M, "Living Bits: Opportunities and Challenges for Integrating Living Microorganisms in Human-Computer Interaction",ACM ISBN 978-1-4503-7603-7/20/03. https://doi.org/10.1145/3384657.3384783

[12] Shen L, Cheng L, Teng F, Makedon F, Ford J, Steinberg T and Saykin A. J, "A multimedia system for the anatomical study of brain images", Proceedings of the ACM Symposium on User Interface Software and Technology (UIST). Dartmouth College, Hanover, NH, USA

[13] Aidan P.Thompsona, H. MetinAktulgab, RichardBergerc, Dan S.Bolintineanua, W.MichaelBrownd, Paul S.Croziera, Pieter J.int Velde, AxelKohlmeyerc, StanG.Moorea, TrungDacNguyenf, RayShang, Mark J.Stevensa, JulienTranchidaa, ChristianTrotta, Steven J.Plimptona, "LAMMPS -a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales", Computer Physics Communications, vol 271, Feb 2022, https://doi.org/10.1016/j.cpc.2021.108171.

[14] Rajendran S, Sundararajan P, Awasthi A. and Rajendran S, "Nanorobotics in Medicine: A Systematic Review of Advances, Challenges, and Future Prospects with a Focus in Cell Therapy, Invasive Surgery and Drug Delivery. Precision Nanomedicine, 7(1), 1221-1232, 2024.

[15] Jang J. S. R and Sun C. T "Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence", 1995, Prentice Hall.

[16] Rehaiem G, Gharsellaoui H and Ben Ahmed S, "A Neural Networks Based Approach for the Real-Time Scheduling of Reconfigurable Embedded Systems with Minimization of Power Consumption." Journal of Embedded Systems, 12(1), 89-102, 2019.

[17] Al-ali A, Elharrouss O, Qidwai U and Al-Maaddeed, S. "ANFIS-Net for automatic detection of COVID-19. Scientific Reports, 11, 17318", https://doi.org/10.1038/s41598-021-96705-2, 2021.

[18] Olah, C. (2015). "Understanding LSTM Networks." arXiv preprint arXiv:1506.00019. Available at: https://arxiv.org/abs/1506.00019.

[19] M.V.S Phani Narasimham, Y.V.S Sai Pragathi. "Modified O-RAN 5G Edge Reference Architecture Using RNN." International Journal of Wireless & Mobile Networks (IJWMN), Vol.16, No.3, June 2024, DOI:10.5121/ijwmn.2024.16301, 2024.

[20] Babichev S, Liakh I and Kalinina I, "Applying a Recurrent Neural Network-Based Deep Learning Model for Gene Expression Data Classification", Applied Sciences, 13(21), 11823. https://doi.org/10.3390/app132111823, 2023.

[21] W. Wang, Z. Wu, and Q. He, "Swimming nanorobots for opening a cell membrane mechanically," *View*, vol. 1, no. 3, pp. 20200005, Jul. 2020, doi: 10.1002/VIW.20200005

[22] W. Si, M. Yu, G. Wu, C. Chen, J. Sha, Y. Zhang, and Y. Chen, "A Nanoparticle-DNA Assembled Nanorobot Powered by Charge-Tunable Quad-Nanopore System," *ACS Nano*, vol. 14, no. 11, pp. 15518-15528, Nov. 2020, doi: 10.1021/acsnano.0c07123

[23] M. Hu, X. Ge, X. Chen, W. Mao, X. Qian, and W.-E. Yuan, "Micro/Nanorobot: A Promising Targeted Drug Delivery System," Pharmaceutics, vol. 12, no. 7, p. 665, Jul. 2020, doi: 10.3390/pharmaceutics12070665

[24] Q. Zhao and L. Lin, "Modeling and Simulation of a Bio-Inspired Nanorobotic Drug Delivery System," in *Bio-inspired Information and Communications Technologies*, pp. 64-76, Sep. 2023, doi: 10.1007/978-3-031-43135-7_7.

[25] N. Farhadian, M. S. Kazemi, F. M. Baigi, and M. Khalaj, "Molecular Dynamics Simulation of Drug Delivery Across the Cell Membrane by Applying Gold Nanoparticle Carrier: Flutamide as Hydrophobic and Glutathione as Hydrophilic Drugs as the Case Studies," *J. Mol. Graph. Model.*, vol. 116, p. 108271, Nov. 2022, doi: 10.1016/j.jmgm.2022.108271.

[26] F. J. Schmitt, V. Rostami, and M. P. Nawrot, "Efficient parameter calibration and real-time simulation of large-scale spiking neural networks with GeNN and NEST," *Frontiers in Neuroinformatics*, vol. 17, Art. no. 941696, Feb. 2023, doi: 10.3389/fninf.2023.941696

[27] A. Pathak, S. L. Brincat, H. Organtzidis, H. H. Strey, S. Senneff, E. G. Antzoulatos, L. R. Mujica-Parodi, E. K. Miller, and R. Granger, "Biomimetic model of corticostriatal micro assemblies discovers new neural code," *bioRxiv*, Nov. 2023, doi: 10.1101/2023.11.06.565902.

[28] S. Andhari, R. D. Wavhale, K. D. Dhobale, B. V. Tawade, G. P. Chate, Y. N. Patil, J. J. Khandare, and S. S. Banerjee, "Self-propelling targeted magneto-nanobots for deep tumor penetration and pH-responsive intracellular drug delivery," *Sci. Rep.*, vol.10, Art. no. 4703, 2020. doi: 10.1038/s41598-020-61586-y

[29] D. K. Patel, H.-B. Kim, S. D. Dutta, K. Ganguly, and K.-T. Lim, "Carbon nanotubes-based nanomaterials and their agricultural and biotechnological applications," *Materials (Basel)*, vol. 13, no. 7, p. 1679, Apr. 2020. doi: 10.3390/ma13071679.

[30] Shwetha K. P., Amogh P. N., Shrish Shrinath Vaidya, Hariprasad N., Sri Krishna, and Manjunatha C., "Nanorobotics for advancing biomedicine: Progress in materials, design, fabrication, opportunities and applications," *IEEE Access*, vol. XX, pp. 1-25, 2024. doi: 10.1109/ACCESS.2024.3416818

[31] Q. Liu, Q. Cui, L. Jin, and X. Li, "The Applications of Buckminsterfullerene C60 and Derivatives in Orthopedic Research," *Connective Tissue Research*, vol. 55, no. 2, pp. 81–87, Jan. 2014, doi: 10.3109/03008207.2013.877894.

[32] Wang, Z., Lee, S., Koo, K., & Kim, K.. Nanowire-Based Sensors for Biological and Medical Applications. IEEE Transactions on Nanobioscience, 15(3), 186-199. doi: 10.1109/TNB.2016.2528258, 2016.

[33] G. Zhang, Z. Hao, J. Liu, K. Nagashima, et al., "Nanowire-Based Sensor Electronics for Chemical and Biological Applications," *The Analyst*, vol. 146, no. 17, pp. 6684-6725, Oct. 2021, doi: 10.1039/D1AN01096D

[34] Intel Corporation. Loihi Neuromorphic Research Chip. Neuromorphic Computing and Engineering with AI | Intel®.

[35] Intel, "Taking Neuromorphic Computing to the Next Level with Loihi 2 Technology Brief," Intel, 2021. https://www.intel.com/content/www/us/en/research/neuromorphic-computing-loihi-2-technology-brief.html

[36] IBM Corporation. TrueNorth: Brain-Inspired Computing. TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip for IEEE TCADIS - IBM Research.

[37] D. V. Christensen, R. Dittmann, B. Linares-Barranco, A. Sebastian, M. Le Gallo, A. Redaelli, S. Slesazeck, T. Mikolajick, S. Spiga, and S. Menzel, "2022 roadmap on neuromorphic computing and engineering," *Neuromorphic Computing and Engineering*, vol. 2, no. 2, Art. no. 022501, May 2022, doi: 10.1088/2634-4386/ac4a83

[38] Advances in Chip Miniaturization. 3nm Technology - Taiwan Semiconductor Manufacturing Company Limited (tsmc.com).

[39] TSMC, "TSMC Holds 3 nm Volume Production and Capacity Expansion Ceremony, Marking a Key Milestone for Advanced Manufacturing," *TSMC Press Release*, Dec. 29, 2022. Available: TSMC

[40] NVIDIA Corporation. Advances in AI and GPU Technology. NVIDIA Official Website.

[41] U. Iqbal, T. Davies, and P. Perez, "A Review of Recent Hardware and Software Advances in GPU-Accelerated Edge-Computing Single-Board Computers (SBCs) for Computer Vision," *Sensors (Basel)*, vol. 24, no. 15, Art. no. 4830, Aug. 2024, doi: 10.3390/s24154830

[42] Graphcore. Intelligence Processing Unit (IPU) Architecture. Graphcore: Accelerating machine learning in the cloud

[43] H. Peng, C. Ding, T. Geng, S. Choudhury, K. Barker, and A. Li, "Evaluating Emerging AI/ML Accelerators: IPU, RDU, and NVIDIA/AMD GPUs," *arXiv preprint arXiv:2311.04417v3*, Mar. 2024, doi: 10.1145/3629527.3651428

[44] MIT Research. Carbon Nanotube-Based Processors. MIT engineers build advanced microprocessor out of carbon nanotubes .

[45] Radisys, "Multi-Access Edge Computing (MEC) Solutions," Radisys, 2024. https://www.radisys.com/solutions/multiaccessedge

[46] A. Bala, R. Z. Jusoh, A. Rashid, I. Ismail, D. Oliva, N. Muhammad, S. M. Sait, K. A. Al-Utaibi, T. I. Amosa, and K. A. Memon, "Artificial intelligence and edge computing for machine maintenance-review," *Artificial Intelligence Review*, vol. 57, article 119, Apr. 2024. DOI: 10.1007/s10462-024-119

[47] ISO, "ISO 80004-1:2023: Nanotechnologies - Vocabulary - Part 1: Core Vocabulary," International Organization for Standardization, Geneva, Switzerland, 2023

[48] ISO, "ISO/TS 4958:2024: Nanotechnologies — Vocabulary — Liposomes," International Organization for Standardization, Geneva, Switzerland, 2024.

[49] ISO, "ISO/TS 5094:2023: Nanotechnologies — Assessment of Peroxidase-Like Activity of Metal and Metal Oxide Nanoparticles," International Organization for Standardization, Geneva, Switzerland, 2023.

[50] A. W. Donatti, C. C. Miers, G. P. Koslovski, M. A. Pillon, and T. C. M. B. Carvalho, "Characterization of Network Management Traffic in OpenStack based on Virtual Machine State Changes," in *Proc. 11th Int. Conf. Cloud Computing Services Sci. (CLOSER 2021)*, pp. 232-239, 2021, doi: 10.5220/0010447102320239.

[51] M. D. Pham, A. D'Angiulli, M. M. Dehnavi, and R. Chhabra, "From Brain Models to Robotic Embodied Cognition: How Does Biological Plausibility Inform Neuromorphic Systems?," *Brain Sci.*, vol. 13, no. 9, p. 1316, Sep. 2023, doi: 10.3390/brainsci13091316.

[52] K. Yamazaki, V.-K. Vo-Ho, D. Bulsara, and N. Le, "Spiking Neural Networks and Their Applications: A Review," *Brain Sciences*, vol. 12, no. 7, p. 863, Jun. 2022. Available: https://doi.org/10.3390/brainsci12070863.

[53] S. Rajendran, P. Sundararajan, A. Awasthi, and S. Rajendran, "Nanorobotics in Medicine: A Systematic Review of Advances, Challenges, and Future Prospects," *Precis. Nanomed.*, vol. 7, no. 1, pp. 1221-1232, Jan. 2024. DOI: 10.33218/001c.92224.

[54] W. Ponghiran and K. Roy, "Hybrid Analog-Spiking Long Short-Term Memory for Energy Efficient Computing on Edge Devices," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021, pp. 581-586. DOI: 10.23919/DATE51398.2021.9473953.
[55] Y. Nasir, "Comparison of Spiking Neural Network Simulator Performance," *SNNSimulatorComparison*, GitHub repository, 2024. https://github.com/nasiryahm/SNNSimulatorComparison.
[56] A. Kjærran, E. S. Bugge, and C. B. Vennerød, "Long Short-Term Memory RNN for Time Series Forecasting: Iterations and Epochs," *TDT4173 Assignment*, Norwegian University of Science and Technology, Mar. 2024.
[57] "LAMMPS Course for Intermediate Users: Measuring and Improving LAMMPS Performance," *EPCC Archer2 Training*, Dec. 2023. https://www.github.com/EPCCed/archer2-advanced-use-of-lammps.git.
[58] "Running LAMMPS on HPC Systems," HPC Carpentry, Software Carpentry Workshop, 2024. https://hpc-carpentry.github.io.
[59] G. Giri, Y. Maddahi, and K. Zareinia, "Brief Review on Challenges in Design and Development of Nanorobots for Medical Applications," *Appl. Sci.*, vol. 11, no. 21, p. 10385, Nov. 2021, doi: 10.3390/app112110385.
[60] H. Zhou, C. C. Mayorga-Martinez, S. Pané, L. Zhang, and M. Pumera, "Magnetically Driven Micro and Nanorobots," *Chem. Rev.*, vol. 121, no. 7, pp. 4999−5041, 2021. doi: 10.1021/acs.chemrev.0c00780.
[61] M. Urso, M. Ussia, and M. Pumera, "Smart micro- and nanorobots for water purification," *Nature Reviews Bioengineering*, vol. 1, pp. 236–251, Feb. 2023.
[62] X. Kong, P. Gao, J. Wang, Y. Fang, and K. C. Hwang, "Advances of medical nanorobots for future cancer treatments," *Journal of Hematology & Oncology*, vol. 16, no. 74, 2023.

8. Appendix – NanoRobots Simulation Framework

The Python Nano Robot Library supports simulations and control tasks for neuro nano-robots. Below are key functions of the simulation.

```
#Directional Influence Calculation
def calculate_directional_influence(positions, source_position):
    influences = []
    for pos in positions:
        distance = np.linalg.norm(np.array(pos) - np.array(source_position))
        influence = 1 / (1 + distance)  # Example influence calculation based on distance
        influences.append(influence)
    return influences

#Drug Effects Application
def apply_drug_effects(neurons, drug_effect, target_precision, positions, source_position):
    neuron_collection = nest.NodeCollection(neurons)
    directional_influence = calculate_directional_influence(positions, source_position)

    for i, neuron in enumerate(neuron_collection):
        influence = directional_influence[i]
        tau_m_adjusted = 20.0 * drug_effect * (1.0 + target_precision * influence)
        nest.SetStatus([neuron], {"tau_m": tau_m_adjusted})

#NEST Simulation with Drug Effects
def run_nest_simulation(drug_effect=1.0, target_precision=0.5):
    nest.ResetKernel()

    positions = np.random.rand(10, 2) * 10  # Random positions in a 2D space
    source_position = np.array([5, 5])  # Example drug source position

    neurons = nest.Create('iaf_psc_alpha', 10)
    apply_drug_effects(neurons, drug_effect, target_precision, positions, source_position)

    spike_generator = nest.Create('spike_generator', params={"spike_times": np.arange(10., 100., 10.)})
    voltmeter = nest.Create('voltmeter')
    nest.Connect(spike_generator, neurons, syn_spec={'weight': 1.5})
    nest.Connect(voltmeter, neurons)
    nest.Simulate(200.0)

    events = nest.GetStatus(voltmeter, "events")[0]
    spike_times = events["times"]
    return np.array(spike_times)
```

```
#RNN Model for Control Signals
def build_rnn_model(input_shape):
    model = Sequential()
    model.add(LSTM(50, input_shape=input_shape, return_sequences=True))
    model.add(LSTM(50))
    model.add(Dense(1))
    model.compile(optimizer='adam', loss='mean_squared_error')
    return model

def train_rnn_model(model, spike_data):
    X = spike_data.reshape((1, -1, 1))  # Reshape for RNN
    y = np.array([np.mean(np.diff(spike_data))])  # Control signal as target
    model.fit(X, y, epochs=10)

def generate_control_signal(model, spike_data):
    X = spike_data.reshape((1, -1, 1))  # Reshape for RNN
    control_signal = model.predict(X)[0, 0]
    return control_signal

#LAMMPS Simulation Function
def run_lammps_simulation():
    lmp = lammps()
    lammps_script = """
    units lj
    atom_style atomic
    lattice fcc 0.8442
    region box block 0 10 0 10 0 10
    create_box 1 box
    create_atoms 1 box
    mass 1 1.0
    velocity all create 1.44 87287
    pair_style lj/cut 2.5
    pair_coeff 1 1 1.0 1.0 2.5
    fix 1 all nve
    run 100
    """
    lmp.commands_string(lammps_script)
    lmp.close()

Targeted Drug Delivery -  Python library
def simulate_nanorobots(control_signal, num_robots=10, steps=100, influence_factor=0.1, target_position=None):
    positions = np.random.rand(num_robots, 2) * 10
    source_position = np.array([5, 5]) if target_position is None else target_position
    position_history = [positions.copy()]

    for _ in range(steps):
        for i, pos in enumerate(positions):
            direction = (source_position - pos) + np.random.normal(scale=0.5, size=2)
            distance = np.linalg.norm(direction)
            influence = control_signal * influence_factor / (1 + distance)
            positions[i] += influence * direction / distance

        position_history.append(positions.copy())

    return np.array(position_history), source_position

#Comparison Graphs
def plot_comparison_graphs(position_history_no_control, position_history_control, target_position, effect, precision):
    plt.figure(figsize=(12, 6))

    for i, positions in enumerate(position_history_no_control):
        if i == 0 or i == len(position_history_no_control) - 1:
```

```
        plt.scatter(positions[:, 0], positions[:, 1], label=f'Step {i} No Control', alpha=0.6, s=20)

for i, positions in enumerate(position_history_control):
    if i == 0 or i == len(position_history_control) - 1:
        plt.scatter(positions[:, 0], positions[:, 1], marker='x', label=f'Step {i} With Control', alpha=0.6, s=40)

plt.scatter(target_position[:, 0], target_position[:, 1], color='green', marker='^', s=100, label='Target Position')
plt.xlabel('X Position')
plt.ylabel('Y Position')
plt.title('Nanorobot Movement with and without Control Signal')
plt.legend()
plt.savefig(f'/tmp/nanorobot_movement_comparison_{effect}_{precision}.png')
plt.close()
```