Low-Power Neuromorphic Sensor Fusion for Elderly Care

Zheqi Yu
Student Number:

Supervisors:
Dr Qammer H Abbasi
Dr Hadi Heidari
Prof Muhammad Imran

PhD Thesis

James Watt School of Engineering
College of Science and Engineering
University of Glasgow

January 2022

Abstract

Smart wearable systems have become a necessary part of our daily life with applications ranging from entertainment to healthcare. In the wearable healthcare domain, the development of wearable fall recognition bracelets based on embedded systems is getting considerable attention in the market. However, in embedded low-power scenarios, the sensor's signal processing has propelled more challenges for the machine learning algorithm. Traditional machine learning method has a huge number of calculations on the data classification, and it is difficult to implement real-time signal processing in low-power embedded systems. In an embedded system, ensuring data classification in a low-power and real-time processing to fuse a variety of sensor signals is a huge challenge. This requires the introduction of neuromorphic computing with software and hardware co-design concept of the system. This thesis is aimed to review various neuromorphic computing algorithms, research hardware circuits feasibility, and then integrate captured sensor data to realise data classification applications. In addition, it has explored a human being benchmark dataset, which is following defined different levels to design the activities classification task. In this study, firstly the data classification algorithm is applied to human movement sensors to validate the neuromorphic computing on human activity recognition tasks. Secondly, a data fusion framework has been presented, it implements multiple-sensing signals to help neuromorphic computing achieve sensor fusion results and improve classification accuracy. Thirdly, an analog circuits module design to carry out a neural network algorithm to achieve low power and real-time processing hardware has been proposed. It shows a hardware/software co-design system to combine the above work. By adopting the multi-sensing signals on the embedded system, the designed software-based feature extraction method will help to fuse various sensors data as an input to help neuromorphic computing hardware. Finally, the results show that the classification accuracy of neuromorphic computing data fusion framework is higher than that of traditional machine learning and deep neural network, which can reach 98.9% accuracy. Moreover, this framework can flexibly combine acquisition hardware signals and is not limited to single sensor data, and can use multi-sensing information to help the algorithm obtain better stability.

Contents

List of Tables

List of Figures

List of Algorithms

List of Symbols

a Learning rate

dE Vector coordinates in the world coordinate system

dS Vector coordinates in the sensor coordinate system

dt Microvariable with t

f_i Transfer function of the neuron i

H Hankel matrix

i Neuron(i)

I_s Bias current

j Neuron(j)

K Number of stored patterns (memorized patterns)

δk Error value between time i and time j

N Number of neurons (neuron node)

P Pattern

P_{max} Maximum pattern memory capacity

q Sensor attitude

R_i Resistance that represent transmission resistance of the cell membrane

R_{ij} Resistor impedance

s State quantity

t Time

Δt Time increment

U Sub-matrix for m×m

u_i Neurons current state

Σ Sub-matrix for m×n

v_i Neurons current output

V_i Internal membrane potential

V^T Sub-matrix for n×n

V_{bias} External input current

V_{out} Output potential of the neuron

W_{ij} Network weight

x_j Neuron's state

X^p Orthogonal vector

z^{-1} Delay

ω Angular velocity

β The weight to represents the error of the angular velocity

α Constant learning rate for the learning step size

θ_j Threshold of the neuron j

η Learning rate parameter

ϕ Roll angle

θ Pitch angle

ψ Yaw angle

List of Abbreviations

AGI Artificial General Intelligence
AI Artificial Intelligence
AM Associative Memory
ANN Artificial Neural Network
ASIC Application-Specific Integrated Circuit
BAM Bidirectional Associative Memory
BAs Basic Activities
BNN Biological Neural Network
BP Back-Propagation
BPTT Backpropagation Through Time
CA Cellular Automata
CAM Content Addressable Memory
CAs Complex Activities
CHNN Continuous Hopfield Neural Network
CNN Convolutional Neural Network
CSI Channel State Information
DBN Deep Belief Network
DBSCAN Density-Based Spatial Clustering of Applications with Noise
DHNN Discrete Hopfield Neural Network
DLN Deep Learning Network
DNN Deep Neural Network
DSP Digital Signal Processor
EA Euler Angle
FFT Fast Fourier Transformer
FMCW Frequency Modulated Continuous Wave
FN False Negative
FNN Fuzzy Neural Network
FP False Positive
FPGA Field Programmable Gate Array
GA Genetic Algorithm

GAN Generative Adversarial Network
GBW Gain–BandWidth
GMM Gaussian Mixture Model
HL Hebb learning
HMM Hidden Markov Model
HNN Hopfield Neural Network
HAR Human Activity Recognition
IMU Inertial Measurement Unit
IoT Internet of Things
KCL Kirchhoff's Current Law
KNN K Nearest Neighbour
LIF Leaky Integrated-and-Fire
LSTM Long Short Term Memory
MCU Microprogrammed Control Unit
MIMO Multiple Input Multiple Output
MSIF Multi-sensor Information Fusion
NB Naive Bayes
NN Neural Network
OFDM Orthogonal Frequency Division Multiplexing
PCA Principal Component Analysis
PCNN Pulse-Coupled Neural Network
PPA Performance, Power, Area
RC Rate Coding
RF Radio Frequency
RFC Random Forest Classifier
RNN Recurrent Neural Network
SDR Softare Defined Radio
SNN Spiking Neural Network
SFS Sequential Feature Selection
SVD Singular Value Decomposition
SVM Support Vector Machine
TN True Negative
TP True Positive
TSP Travelling Salesman Problem
USRP Universal Software Radio Peripheral

List of Publications

Journal:

1. **Zheqi Yu**, Amir M. Abdulghani, Adnan Zahid, Hadi Heidari, Muhammad A. Imran, Qammer H. Abbasi., 2020. An Overview of Neuromorphic Computing for Artificial Intelligence Enabled Hardware-Based Hopfield Neural Network. IEEE Access, 8, pp.67085-67099. **(Published)**.

2. **Zheqi Yu**, Pedro Machado, Adnan Zahid, Amir M. Abdulghani, Kia Dashtipour, Hadi Heidari, Muhammad A. Imran, Qammer H. Abbasi., 2020. Energy and performance trade-off optimization in heterogeneous computing via reinforcement learning. Electronics, 9(11), p.1812. **(Published)**.

3. **Zheqi Yu**, Adnan Zahid, Shuja Ansari, Hasan Abbas, Amir M. Abdulghani, Hadi Heidari, Muhammad A. Imran, Qammer H. Abbasi., 2020. Hardware-Based Hopfield Neuromorphic Computing for Fall Detection. Sensors, 20(24), p.7226. **(Published)**.

4. **Zheqi Yu**, Adnan Zahid, Shuja Ansari, Hasan Abbas, Hadi Heidari, Muhammad A. Imran, Qammer H. Abbasi., 2021. IMU Sensing-based Hopfield Neuromorphic Computing for Human Activity Recognition. Frontiers in Communications and Networks. **(Published)**.

5. **Zheqi Yu**, William Taylor, Ahmad Taha, Adnan Zahid, Julien Le Kernec, Hadi Heidari, Muhammad A. Imran, Qammer H. Abbasi., 2021. An Intelligent Implementation of Multi-Sensing Data Fusion with Neuromorphic Computing for Human Activity Recognition. IEEE Internet of Things Journal. **(Under Review)**.

6. **Zheqi Yu**, Ahmad Taha, William Taylor, Adnan Zahid, Khalid Rajab, Hadi Heidari, Muhammad A. Imran, Qammer H. Abbasi., 2021. A Radar-based Human Activity Recognition Using a Novel 3D point cloud classifier. IEEE Sensors Journal. **(Under Review)**.

Conference:

1. **Zheqi Yu**, Adnan Zahid, William Taylor, Hasan Abbas, Hadi Heidari, Muhammad A. Imran, Qammer H. Abbasi (2021). Data Fusion for Human Activity Recognition based on RF Sensing and IMU Sensor. EAI BODYNETS 2021, Springer. **(Published)**

2. **Zheqi Yu**, Adnan Zahid, William Taylor, Hadi Heidari, Muhammad A. Imran, Qam-

mer H. Abbasi (2021). Multi-Sensing Data Fusion for Human Activity Recognition Based on Neuromorphic Computing. 2021 IEEE AP-S Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting. **(Published)**

Book Chapter:

1. **Zheqi Yu**, 2021. Using the TULIPP Platform to Diagnose Cancer. In Towards Ubiquitous Low-power Image Processing Platforms (pp. 193-198). Springer, Cham. **(Published, the Book Chapter is invited by Sundance Multiprocessor Technology Ltd on the Feb/2019)**

Acknowledgements

I am very grateful to many people who have helped me during my three and a half years of PhD study. Especially my supervisors, this research would not be possible to complete without their teaching and training.

First of all, I would like to show significant gratitude to my supervisor, Dr. Qammer H. Abbasi, for all the work he has done in my PhD research and all his solid suggestions and support for my project. Significantly when I was writing my thesis, he patiently helped me again and again sort out the paper's logic until there were published. Then, I would like to many thanks to the other members of my supervision team, Dr. Hadi Heidari and Professor Muhammad Ali Imran, who gave me many school resources and help.

I am very honoured to join the Communications, Sensing and Imaging (CSI) research group, and very pleased to get teamwork from Adnan, William, Ahmad, Zakir, Hira, Kazim, Siming, Jin, Yuchen, Paulo, Bruno, etc. Their help in my research work and discussion. Especially Adnan, who participated in all my projects and papers, gave me much sincere advice and helped every time.

In addition, I would like to thank the University of Glasgow and the Transreport Ltd (UK) Company for the joint industrial scholarship to fund my three and a half years of research during my PhD at the University of Glasgow. Meanwhile, I would like to thank Sundance Multiprocessor Technology Ltd for providing me with opportunities for internship in London and research cooperation and inviting me to participate in the cooperation of Book Chapter.

Finally, I would like to thank my parents for their support and encouragement during my PhD study period, which helped me complete my overseas studies.

Declaration of Authorship

I, Zheqi Yu, confirm that this thesis and the work presented in it are my own achievement.

Statement of Copyright

Chapter 1

Introduction

1.1 General Background

With the popularity of smart wearable systems, it has brought more convenient applications to the health-care area. Among them, the development of wearable fall recognition bracelets based on embedded systems has become a hot spot in the market. It is composed of non-invasive and wearable sensors with processors [1]. However, the combination of machine learning and embedded devices still has a technological divide. The low power and real-time health care Artificial Intelligence (AI) platform has great potential in smart wearable systems. In this respect, neuromorphic computing hardware as AI techniques has been proposed by researchers and scientists at various levels in the field of wearable sensors to develop health care applications. Von Neumann structure is a traditional and popular hardware design, it is a memory structure that combines programs, instructions, and data. Von Neumann structure encodes the program into data format, which is then stored in memory along with the general data. The computer processes the data by calling the program in the memory. This means that all programs will eventually be converted into data and stored in the memory, and executing the corresponding program requires fetching instructions from the memory and executing them in turn. However, the CPU memory accesses speed as the main factor affecting the system processing power, which makes the system absolutely dependent on the memory.

Neuromorphic computing as an exploration that is different from Von Neumann's architecture on the hardware processing platform. It is separated from the Von Neumann structure from the bottom design, which adopts the same storage and computing design as the brain. It is the integrated storage and computing architecture, which saves on moving data from a memory unit to a computing unit. Meanwhile, neuromorphic computing uses the biological neurons method to complete the machine learning process. It builds a synapse and neuron-like system by programming electronic components into discrete resistance states and convolving electronic components of different weights with each other. The hardware support low power and real-time for the sensor signal processing, which is a brilliant method in the wearable field [2].

In the realisation of AGI (Artificial General Intelligence) [3], it has two alternative research routes: a. computer science-oriented and b. neuroscience-oriented. There are essential differences in the ideas, concepts and implementation plans of these two ways. Most artificial intelligence applications are based on computer science-oriented, as it relies on the famous Von Neumann architecture [4].

For example, the current popular deep learning algorithms partly simulate the human brain's neural network structure. However, its process depends on the Von Neumann architecture to complete the calculation, which modelling and solving based on optimisation problems. Its essence is still According to the calculation of the time-sharing sequence [5]. So that it will show the high computing power requirements [6], the amount of calculation is large [7], and the algorithm is complicated [8].

Unlike the Von Neumann architecture approach to integrated circuits, the brain's extraordinary processing power from a neuroscience perspective is down to three essential points: a. broad connectivity, b. organisational hierarchies of structure and functionalisation, c. time-dependent synaptic connections of neurons. Neurons as the brain's computational primitive elements, which exchange and transmit information through discrete action potentials, and Synapses are the primary storage elements for memory and learning.

At this point, neuromorphic computing is guided by neuroscience oriented. It works low power consumption to conduct asynchronous, parallel and distributed processing for information by simulating the processing mode of the brain. Neuromorphic computing has a variety of abilities such as autonomous perception, recognition and learning. In this way, it gets free from the Von Neumann architecture on the underlying design, which adopts the memory-computing integrated design as the brain.

It uses neuronal computing architecture to complete the required tasks; after designing a basic neuron, a neural network structure of neuromorphic computing can be built through a flexible combination of the neurons. Such a design breaks through the limitations of traditional computer architecture and can realise parallel data transmission, distributed and real-time processing of massive data with extremely low power consumption.

In contrast, traditional Deep Learning Networks (DLN) are essentially artefacts of hierarchical structure [9] with many layers to represent potential features. It is transformed by representations of different features from multiple layers in the input process. This neural network was developed to match silicon transistor types' hardware computing systems. The digital logic of large-scale computing platforms consists of billions of transistors integrated on a single silicon chip. Therefore, various silicon-based cells are arranged hierarchically for efficient data exchange suitable for the algorithm hierarchy concept.

Currently, some traditional neural networks are widely used, such as Convolutional Neural Networks (CNN) [10], Recurrent Neural Networks (RNN) [11], Generative Adversarial Networks (GAN) [12]. These neural network algorithms are based on the back-propagation algo-

rithm and the various mathematical optimisers that achieves excellent performance in various tasks. However, the computing level of the traditional neural network is divided into processing units and memory storage. The physical separation of processing units and memory hierarchies creates what is known as a "memory wall bottleneck" [13].

Neuromorphic computing is a new generation neural network model that is inspired by biology. It is fundamentally different from these traditional neural networks. Despite the superficial similarities, there are sharp differences between the computing principles of the brain works and silicon-based computers. These include:

• The computing section (processing unit) and storage section (memory unit) are separated in the computer. It unlike neurons and synapses in the brain, which are memory-computing integrated design;

• Transistors are mainly used to construct deterministic Boolean (Digital) circuit switches, which are different from the event-driven random computing based on spike signals in the brain.

• Traditional neural network algorithm is based on the high-precision floating-point numbers for operations, but the brain does not use floating-point numbers for operations. Instead, in the perception system of the brain, information will be in action voltages or electric spikes to transmission, reception, and processing.

• The training process of traditional neural networks relies on the back-propagation algorithm that is mainly on gradient descent. However, the brain's memory and learning depend on the synaptic plasticity that occurs when postsynaptic cells are stimulated. For example, Hebbian learning for

• Traditional neural networks usually require a large number of labelled datasets to drive network fitting. This is different from the brain's experiences, and its perception and learning process in many cases is unsupervised. The brain does not need such a large amount of repeated data to learn the same thing.

To sum up, neuromorphic computing was produced to make neural networks more similar to the brain. It was inspired by the concept that biological brains process information. Neuromorphic computing is not a traditional neural network like CNN and RNN, but it is a new type of neural network that is closer to the brain.

1.2 Research Motivation and Challenges

Currently, artificial intelligence applications have limited usage scenarios due to high power and resource requirements at algorithm runtime. The embedded system is not capable of running a neural network for real-time multi-sensors (such as magnetic, acceleration and RF/Radar sensors) signal processing for activity detection. Neuromorphic chip fit and integrates the neural network into hardware that is an excellent choice to solve this. It can achieve milliwatt level power consumption and process multi-modal data in real-time. Previous works in the area have

not fully utilised the advantages of neuromorphic computing for associative memory features. The data from multiple sensors can be fused together on the neuromorphic chip to detect activity more efficiently and accurately. The data coming from multi-sensors can help make more accurate decisions than single-class data type, for instance, a magnetic sensor for geomagnetic variations in human altitude change in the fall [14], and an acceleration sensor for postural responses when the human fall [15]. The RF/Radar sensor is available in the indoor environment for acquiring human activity by Doppler shift information [16]. When the above multi-sensors data is fused, a wide range of information can be evaluated to avoid a single data effect of activity feature, improving human activity recognition accuracy. The research challenges are to develop real-time Neuromorphic computing hardware, and design feature computing to multi-sensing signal processing for human activity recognition. The designed Hardware/Software system is for real-time and low-power signal processing of human beings with multiple sensors simultaneously and improves the system's robustness in different environments compared to state of the art.

Problem Statement:

In ongoing research work, neuromorphic computing can be work with associative memory features to store signals. However, the key is how to design a hardware module to stimulate neurons for data classification. Neuromorphic computing can be adopted to the Hebbian learning method for training neural networks with the data, the weights of the feature map are obtained and solidified into the hardware design. Hence, the hardware allows signal processing for low power and real-time runtime and is crucial to an embedded system. Meanwhile, following the hardware acceleration advantage, it is can be combined with a more different type of signals into the platform, which helps the neuromorphic computing output more precision and efficiency. The aim of this project is to design a data fusion method to suit the neuromorphic computing to match the multi-hardware input. Finally, the application will focus on the HAR tasks, depending on multi-sensing devices acquire for human being signals, and it can be used in health care areas. Such as more accuracy for fall detection or other activity recognition that help disabled or elder persons achieve a more intelligent and more convenient life.

1.3 Scope of Research

For the machine learning applications, Von Neumann architecture caused power consumption, and computation has surged. A new architecture is crucial for designing a low power and real-time system, which can be used to implement applications from sensor signals to data classification processing. This design should have a tendency to improve wearable devices that are getting out of high energy consumption and low computing power on the runtime. Neuromorphic computing architecture has the potential and is deemed to save energy and accelerate computing on the implementation. It is different from the Von Neumann architecture where storage and com-

puting are separated. Neuromorphic computing can process data more efficiently as a neuron model, which is used to combine storage and processing. Neuromorphic computing can process data in real-time and does not require data interaction between multiple modules during the calculation.

This project focuses on the research of neuromorphic computing hardware with multiple sensors to achieve health care applications. It uses hardware circuits to implement low-power and real-time neuromorphic computing with pre-processing data algorithm as a hardware/software co-design system. The hardware circuit features can implement AI algorithm calculations and avoid high power consumption and the complexity of the computing architecture. The algorithm adopts the integrated structure of storage and computing like human brain processing. The software-based data pre-processing is used to feature extraction on the multiple sensors signal, and then work together with neuromorphic hardware to finish applications. In the Al application, neuromorphic hardware combined with feature extraction method achieves flexible architecture, modular design, and non-declining computing efficiency, and its potential is very significant [17]. Meanwhile, this aforementioned approach using neuromorphic computing can implement into hardware to obtain advancement with low power and real-time processing [18].

1.4 Research Objectives

The research is based on Neuromorphic computing hardware's own design to recognise human beings' status, including vital signs, movement and pose acquired by Sensor, Radar and RF of multi-sensing data. This work aims to develop low power healthcare systems based on the own Neuromorphic computing hardware, which can monitor people activities and vitals simultaneously in real-time, such as Smart bracelets giving them greater independence without needing to rely on carers. This Neuromorphic computing data fusion framework is exploiting the associative memory feature of Neural Networks. A hardware module has been designed to simulate the Neural Network algorithm, which uses multi-sensing information integration and data classification to recognise human activities. By adopting the Hebbian learning method for training neural networks, weights of human activity features are obtained and implemented/embedded into the hardware design. Here, the neural network weight of human activities is achieved through data preprocessing, and then the weight is mapped to the amplification factor setting in the hardware.

Objectives:

1. To study and perform the thorough literature review and recognise the recent research performed on various techniques of Neuromorphic Computing hardware, and understand the working principle with Hopfield neural network.
2. To analyse and fuse Inertial Measurement Unit (IMU) Sensor, Universal Software Radio Peripheral (USRP) and Radio Frequency (RF)/Radar data, for human activity recognition.
3. To design Neuromorphic computing for data classification signal processing based on the

Hopfield Neural Network.
4. To implement Neuromorphic computing for hardware acceleration of data classification with multi-sensing fused data, theoretical analysis on the low power and real-time neuromorphic computing hardware.
5. To design a hardware module to simulate the neuromorphic computing algorithm, which uses data fusion and classification to monitor people's activities for elderly care of fall detection.

1.5 Main Contributions

This thesis proposes a neuromorphic computing data fusion framework to improve the human activity recognition ability from multiple data collection hardware. By comparing the different combinations of sensors and various classification algorithms, it is verified that the framework based on this design can flexibly fuse multiple types of data and achieve high-precision recognition work. Meanwhile, neuromorphic computing is implemented in hardware simulation.

Algorithm Contributions:

• The associative memory function of neuromorphic computing is utilised to help limited sample datasets. As a result, it achieves better HAR task performance than traditional machine learning and deep neural network. Furthermore, the HNN algorithm as neuromorphic computing is used to realise One-shot learning, which dramatically reduces the training cost, and the idea is verified on a limited sample dataset.

• In this thesis, a neuromorphic computing data fusion method is designed based on the attention mechanism to select suitable features and complete feature-level data fusion. It is proved that this method can be flexible fuse human movement information from three different types of hardware: IMU, Radar, and USRP.

Application Contributions:

• The NodeNs sensor is used to collect the point cloud data of human movements, and the different human activities are defined and analysed. Finally, a benchmark dataset of multi-subjects and multi-person is obtained. It can be used to monitor the movement of humans in all aspects, which helps hospitals get better care for disabled persons, such as the elderly. So, this research has completed a health care application, which starts from human being data collection and analyses of human activities. Then the data is preprocessed to obtain activity features to complete HAR task.

Hardware Contributions:

• The HNN of neuromorphic computing is successfully built circuits based on the amplifier and the summing unit on the hardware simulation software. Furthermore, the correctness of the data transmission is verified through the input and output voltage, which shows neuromorphic computing obtaining the effect of hardware acceleration.

1.6 Thesis Organization

Chapter 2 reviews Neuromorphic computing technologies for Data Classification utilized in the literature and investigates the possible approach for hardware design with Neuromorphic computing algorithm.

Chapter 3 depicts the IMU Sensor signal processing based on a Neuromorphic computing algorithm to human activity recognition, which is based on Hopfield Neural Network to achieve the data processing for the IMU Sensor. It is built in three different types of Sensors, there are Gyroscope, Accelerometer and Magnetometer, and then complete the recognition task for three activities.

Chapter 4 describes the neuromorphic computing data fusion method. It is designed to fuse multi-sensing signals for data pre-processing. It is used an Attention Mechanism to feature selection for fused data. The first is data pre-processing which is individually feature extraction work for each data collection hardware, and then Attention Mechanism based on TopK to sort the feature, and finally make an activity feature map by the selected features. It is fed in Hopfield Neural Network to complete HAR work.

Chapter 5 presents the hardware implementation for Neuromorphic computing, it designed a circuit by amplifier and summer to represent Hopfield neural network. Then following the hardware simulation tools to verify the algorithm that is achieved human fall detection.

Chapter 6 shows a benchmark dataset for human activity data collection by NodeNs sensor. Its focus on a standard workflow for human activity defines and design. The human activity is stored in a point cloud format, and evaluated by state of the art technology for HAR work.

Chapter 7 summarizes the research on the thesis, and give some advice on possible ways to improve the research field.

Chapter 2

Literature Review

Compared with Von Neumann's computer architecture, neuromorphic systems offer more unique and novel solutions to the artificial intelligence discipline. Inspired by biology, this novel system has implemented the theory of human brain modeling by connecting feigned neurons and synapses to reveal the new neuroscience concepts. Many researchers have vastly invested in neuro-inspired models, algorithms, learning approaches, operation systems for the exploration of the neuromorphic system and have implemented many corresponding applications. Recently, some researchers have demonstrated the capabilities of Hopfield algorithms in some large-scale notable hardware projects and seen significant progression. This chapter presents a comprehensive review and focuses extensively on the Hopfield algorithm's model and its potential advancement in new research applications. Towards the end, the chapter concludes with a broad discussion and a viable plan for the latest application prospects to facilitate developers with a better understanding of the aforementioned model in accordance to build their own artificial intelligence projects.

2.1 Introduction

Nowadays, neuromorphic computing has become a popular architecture of choice instead of Von Neumann computing architecture for applications such as cognitive processing. It is based on biologically inspired methods to build highly connected synthetic neurons and synapses, which achieves theoretical neuroscientific models. The Von Neumann architecture is the computing standard predominantly for machines. However, it has significant differences in organizational structure, power requirements, and processing capabilities relative to the working model of the human brain [19]. Therefore, neuromorphic calculations have emerged in recent years as an auxiliary architecture for the Von Neumann system. Neuromorphic calculations are applied to create a programming framework. The system can learn and create applications from these computations to simulate neuromorphic functions. These can be defined as neuro-inspired models, algorithms and learning methods, hardware and equipment, support systems and applica-

tions [20].

Neuromorphic architectures have several significant and special requirements, such as higher connection and parallelism, low power consumption, memory collocation and processing [21]. Its strong ability to execute complex computational speeds compared to traditional Von Neumann architectures, saving power and smaller size of the footprint. These features are the bottleneck of the Von Neumann architecture, so the neuromorphic architecture will be considered as an appropriate choice for implementing machine learning algorithms [4].

There are some main motivations for using neuromorphic architecture, including Real-time performance, Von Neumann Bottleneck, Scalability, Low power, Fault Tolerance, Faster, Neuroscience [19]. It is particularly in Von Neumann Bottleneck and Neuroscience. The Von Neumann Bottleneck is a problem for the memory wall, which is the growing disparity of speed between CPU and memory outside the Chip. However, neuromorphic architecture is a storage-computation integrated structure that is avoided for the memory wall problem. For Neuroscience, it is different from the computer science-oriented design concept, the neuromorphic architecture is more adapted for biological models of the human brain. Among them, real-time performance is the main driving force of the neuromotor system. Through parallelism and hardware-accelerated computing, these devices are often able to perform Neural Network (NN) computing applications faster than Von Neumann architectures [22]. In recent years, the more focused area for neuromorphic system development has been low power consumption [22] [23] [24]. Biological Neural Networks (BNN) are fundamentally asynchronous [25], and the brain's efficient data-driven can be based on event-based computational models [26]. However, managing the communication of asynchronous and event-based task in large systems is a challenging in the Von Neumann architecture [27]. The hardware implementation of neuromorphic computing is favourable to the large-scale parallel computing architecture as it includes both processing memory and computation in the neuron nodes and achieves ultra-low power consumption in the data processing. Moreover, it is easy to obtain a large-scale neural network based on the scalability. Because of all aforementioned advantages, it is better to consider the neuromorphic architecture than Von Neuman for hardware implementation [28].

The basic problem with neuromorphic calculations is how to structure the neural network model. The composition of biological neurons is usually composed of cell bodies, axon, and dendrites. The neuron models implemented by each component of the specified model are divided into different groups, based on the type of model being distinguished by biologically and computationally driven. It is normally can be separated as follows:

• Biologically Plausible [29]: Specifically simulate the behaviour type present in the biological nervous system. Such as a Hodgkin-Huxley model, the understanding of neuronal activity from the perspective of ions entering and leaving the neuron is based on a four-channel nonlinear differential equation [30]. Another one is Morris Lecar model, which depends on a two-dimensional nonlinear equation for effective and simple calculation and implementation [31].

Meanwhile, a calcium-based model is a simplified biologically plausible implementation, providing the link concept between stimulation protocols, calcium transients, protein denoting cascades and induced synaptic changes [32]. Finally, for a Galves-Löcherbach model, it combines the spiking levels with biological rationality, and a model with inherent randomness [33].
• Biologically-Inspired [34]: Ignore biological rationality to replicate biological nervous system behavior. Such as the Izhikevich model, has both simplicity and the ability to replicate biologically precise behavior [35]. Other one is Hindmarsh-Rose model, which satisfactorily explains the dynamics of pulse firing, cluster firing, and chaotic behavior of neurons [36].

Similarly, synaptic models can be divided into two categories. One of the synaptic models is bio-inspired synaptic implementations that include spike-based systems and feed-forward neural networks [37]. For more complex synaptic models, another common approach is based on plasticity mechanism that depends on the intensity or weight of a neuron to change over time [38].

Neuromorphic systems request different network topologies to complete different work areas. In network models, the more popular implementations are feed-forward neural networks, such as multilayer sensing, and other architectures include Recurrent neural networks, Stochastic neural networks [39], Spiking neural networks [40], Artificial neural network, cellular neural networks [41] and Pulse-Coupled Neural Networks (PCNN) [42], Cellular Automata (CA) [43], Fuzzy Neural Networks (FNN) [44]. Hopfield network as the RNN network architecture is especially common in the early implementation of neural morphology, which is consistent with the neural network research trend, there are more recent implementations now. Such as graph partition [45], fault detection [46] and data classification [47], etc.

For the algorithm, the learning method should match each requirement differences on specific network topology, neuron model or other features of network model. In the algorithm learning process, supervised learning is generally not considered as an online method. Online learning is normally applied for unsupervised tasks of cluster analysis. The widely used algorithm for programming neuromorphic systems is back-propagation technique. In contrast, unsupervised learning is based on self-organizing maps or self-organizing learning rules.

Neuromorphic implementation based on high-level standards is to divide hardware implementation into three categories: digital, analog, and hybrid analog/digital platforms [34]. The analog system takes advantage of the physical characteristics on the electronic device to achieve the computation process, while digital systems tend to rely on logic gates to perform the computation process. In contrast, the biological brain is an analog system that relies on physical properties for computation rather than Boolean logic. Because the neural network can be resistant to noise and faults, it is a good solution for analog implementation [48]. Two major categories of digital systems are processed to neuromorphic implementation that are Field Programmable Gate Array (FPGA) and Application Specific Integrated Circuit (ASIC). The first one for FPGA, which has been used frequently in neuromorphic systems [34]. Another one is

custom or ASIC chips, which is also common neuromorphic implementations [49].

For neuromorphic systems, custom analog integrated circuits have several universal features which make them suitable for each other. There are all properties that occur in both analog circuits and biological systems, such as the conservation of charge, amplification, thresholding and integration [50]. Because of the analog circuits similar to biological systems, widely used in hybrid analog/digital neuromorphic systems for the implementation of neuronal and synaptic components. Moreover, several problems with analog systems about unreliability can be addressed by using digital components. Meanwhile, analog neuromorphic systems of synaptic weights are often stored in digital memories. Neuromorphic system communication includes both intra-chip and inter-chip communication [51].

One of the software tools on the neuromorphic system includes custom hardware synthesis toolset. These synthesis tools usually require a relatively high level of description and conversion, which can be used to implement a low-level representation of the neural circuits on the neuromorphic system [52]. The second set of software tools for the neuromotor system is a programming tool of neuromotor systems, which include two functions: mapping and programming [53]. The software simulator developed to test and verify the neuromotor system that is based on a software-based simulator for hardware performance. For applications, in order to demonstrate the computational and device capabilities of neuromorphic computing, various types of neural networks have been applied to different applications area, including images [54], speech [55], data classification [56], control [57], and anomaly detection [58]. To achieve these types of applications on hardware, neural networks matched lower power consumption, faster calculations, and footprint ratios delivered are superior to those delivered by using Von Neumann architecture.

The rest of the chapter is organized as follows: Section 2.2 introduces the details of Hopfield algorithm. Section 2.3 extends to discrete Hopfield network architecture and hardware implementation. Section 2.4 describes the learning method in Hopfield algorithm. Section 2.5 presents the applications of Hopfield algorithm and shows the application development details. Section 2.6 discusses some of the research open challenges and future trends in the Hopfield algorithm. Section 2.7 summarizes the entire discussion of hardware research on Hopfield algorithms and hardware implementation.

2.2 Hopfield Algorithm

The Hopfield network is an important algorithm in the history of neural network development. Professor J.J Hopfield [59], a physicist at the California Institute of Technology, proposed in 1982 that it is a single-layer feedback neural network. Hopfield Neural Network (HNN) is the simplest and most applicable model in feedback networks [60], because it has the function of associative memory [61], which can accurately identify the object and accurately identify digital

signals even if they are contaminated by noise.

The Hopfield neural network model is a kind of recurrent neural network [62]. There is a feedback connection between the input and the output. Under the input excitation, it will be in a constant state of flux. The feedback network can be divided as stable and unstable, which is by judging its stability. For a Hopfield network, the key is to determine its weight coefficient under stable conditions [62]. If the weight matrix W of the Hopfield the network is a symmetric matrix, and the diagonal elements are 0 then it indicates that the network is stable.

According to the discrete or continuous output of the network, the Hopfield network is divided into two types: Discrete Hopfield Neural Network (DHNN) and Continuous Hopfield Neural Network (CHNN) [63].

DHNN: The output of a neuron takes only 1 and 0, which is respectively indicating the neuron in an activation and inhibition state [64].

CHNN: a topology structure is identical to the DHNN. But the difference is whether its activation function is a discrete step function [65] or a continuous function of sigmoid [66].

Due to the structural characteristics of discrete Hopfield network, the output data is equal to the input mode size and dimension. Meanwhile, it is the neurons that take binary values (1, -1) or (1, 0) as input and output. The synaptic weight between neuron i and neuron j is w_{ij} [67]. So for a Hopfield neural network with N numbers of neurons, the weight for the matrix is NxN size. Its unique associative memory process is through a series of the iterative process until the system is stable [68].

Discrete Hopfield network is a feature that can be used for associative memory. This is one of the intelligent characteristics of human beings, so the Hopfield algorithm can simulate human "want" [69]. By reviewing and thinking about the past scenes, it is used as the associative memory of the Hopfield Neural network. Firstly, the learning training process is determine the weight coefficient of the network, and then the memorized information is stored in the N-Dimensional hypercube of a minimal energy corner [70]. Meanwhile, after the weight coefficient of the network is determined, as long as a new input vector is given to the network, this vector may be local data, incomplete or partially incorrect data, but the network still produces a complete output of the information being remembered [71].

The most prominent feature of the Hopfield neural network concept is designed closely related to circuit hardware deployment [72]. The main idea of the Hopfield is the use of the hardware circuit to simulate neural network optimization process. This process can be fast that takes an analog circuit processing advantage rather than digital circuit [73]. Unlike the software realization of the Hopfield neural network, the hardware implementation of the algorithm makes brain-like computations possible [74].

Hopfield is based on the idea of energy function to create a new calculation method, which is through the nonlinear dynamics method for developing this neural network. It has clarified the relationship between the neural network and dynamics model [75]. Then, established the

stability criterion of the neural network on this algorithm. Meanwhile, it points out that the information is stored in the connection between the neurons of the network, eventually results build a Hopfield network. In addition, Hopfield algorithm compares the feedback network with the Ising model in statistical physics and defines the upward and downward directions of the magnetic rotation as neuron's two states of activation and inhibition [76]. That means the magnetic rotation interaction as the synaptic weight on the neuron. This logicality helped many physics theory and physicists to enter the field of neural networks. In 1984, Hopfield designed and developed the circuit of the network algorithm model [77], it is stating that neurons can be implemented with operational amplifiers, and all neuron connections can be simulated by electronic circuits [78]. One of the continuous Hopfield networks using circuit deployed, which is successfully solved Travelling Salesman Problem (TSP) calculation problem. It proves that the Hopfield circuit can address the optimization problem [79].

Moreover, Hopfield network can convert analog signals into a digital format that is to realise associative memory, signal estimation and combination optimization applications [80]. This solution is similar to the method of the human first layer to achieve signal processing. So, it belongs to the neuromorphic calculation. Due to the algorithm stability of output digital signal, Hopfield neural can withstand the redundant input of analog signal noise or variable [81]. This situation is in contrast to the interface circuit between the traditional analog transmission medium and the digital computing device [82]. It takes the speed advantage of the analog circuit and the noise reduction ability of the digital circuit into account.

2.3 Discrete Hopfield Network

Hopfield algorithm as a single-layer fully interconnected feedback network that includes symmetric synaptic connections to stores information on the connections between neurons. It is forming a discrete neural network that is characterized by parallel processing [83], fault tolerance and trainability [84].

Discrete Hopfield network of function that simulates the memory of biological neural network is often called associative memory network. Associative Memory (AM) is an integral part of neural network theory [85], and it is a significant function in artificial intelligence and other fields that are used for pattern recognition [84], image restoration [86], intelligent control [87], optimal computation [88] and optical information processing [89]. It mainly uses the good fault tolerance of neural networks to restore incomplete, defaced and distorted input samples achieve complete prototypes, which are suitable for recognition, classification purposes [90].

Association is based on memory where the information is stored first, and then retrieved in a certain way or rule. Associative Memory (AM) is also called content-addressed memory, which means the process of AM is the process of accessing information [91]. Information is distributed in the content of biological memory, rather than a specific address. The storage of

information is distributed, not centralized. The storage of information is Content Addressable Memory (CAM) that is distributed, not centralized [92]. Whereas traditional computers are based on addressable memory, which is a group of information with a certain storage unit [93]. In comparison, Hopfield neural networks are more consistent with the information storage mode of biological memory. It is distributed storage of the information in the connection weight matrix of the neural network, which is easy directly recalled the content of the information [67].

According to different memory recall methods, associative memory networks can be divided into static memory and dynamic memory networks. The first advocated a forward mapping of inputs, and the other is the memory process for the interactive feedback of input and output. Since the dynamic network has good fault tolerance, it is the most commonly used associative memory [94]. The common dynamic memory network is the Hopfield model (auto-associative memory) [95] and Kosko's Bidirectional Associative Memory (BAM) model (hetero-associative memory) [96].

The applied classification based on associative memory can be divided into auto-associative memory and hetero-associative memory. Auto-associative memory refers to recovering from the damaged input mode to the complete mode; it can map the input mode in the network to one of the different modes stored in the network. At this point, the associative memory network can not only map the input to the self stored modes, but also have some fault tolerance for the input mode with default or noise [64]. Hetero-associative memory refers to obtaining other relevant patterns from input patterns. When the hetero-associative network is excited by input patterns with certain noise, it can associate the pattern pairs of the original samples through the evolution of the state [97].

In the process of realizing associative memory, discrete Hopfield neural network is divided into two working stages: learning-memory stage and associative memories stage. The task for the learning-memory stage is to adjust the weights on the input samples, which the stored samples become dynamics factors on the Hopfield neural network [98]. The task for the associative memories stage is to make the final steady-state as attractor dynamics after adjusted weights, it is depending on the incomplete or affected information as the associative keyword. [99]. In fact, this associative memory process is the continuous movement of the energy function inside the Hopfield neural network, so that the energy is continuously reduced, eventually reaching a minimum value, and in a steady-state process [100].

From the perspective of learning processing, Hopfield algorithm is a powerful learning system with simple structure and easy programming. The Hopfield network operates in a Neural Dynamics, and its working process is the evolution of the state, which means the evolution from the initial state in the direction of energy reduction until it reaches a stable state, which is the output results. Therefore, the state of the Hopfield network evolves in the direction of decreasing energy function. Since the energy function is bounded, the system will incline to a stable state, which is the output of the Hopfield network [101].

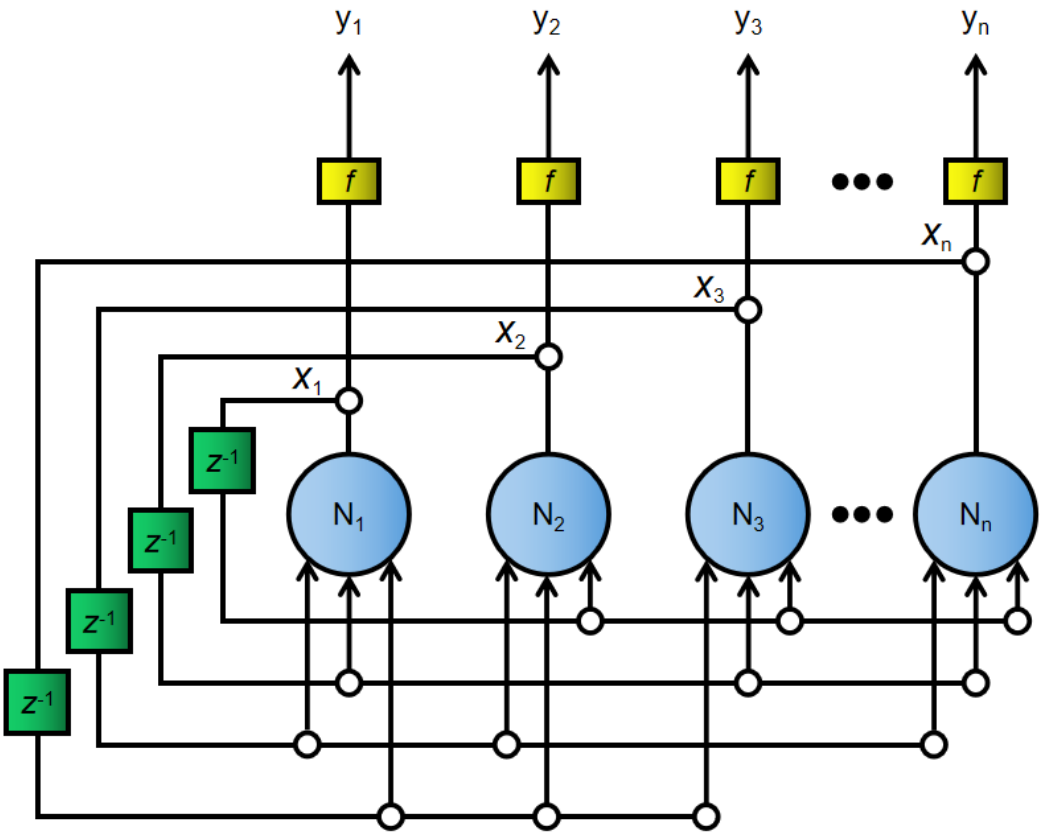


Figure 2.1: The fully connected network architecture of the Hopfield network (adapted from [105].)

From a system perspective, the feedforward neural network model has limited computing power. Comparatively, the feedback neural network has more stronger computing power than a feedforward neural network, it is based on feedback dynamics to enhance global stability [102]. In feedback neural networks, all neurons have the same status and there is no hierarchical difference. They can be connected to each other and also feedback signals to themselves [103]. In contrast, although the Back-Propagation (BP) neural network model can handle learning problems, it is not suitable for combinatorial optimization problems. In theory, if the application is properly set, Hopfield Neural networks can be more robust on the applications. It is a static nonlinear mapping, and the nonlinear processing capability of the complex system can be obtained by the compound mapping of the simple nonlinear processing unit [104].

The discrete Hopfield neural network (DHNN) which is also known as a feedback neural network is fully connected network architecture and is shown in Figure 2.1. The circles represent neurons, and the output of each neuron as the input of other neurons, which means that the input of each neuron comes from other neurons. In the end, other neurons will return the output data to themselves. At this time, each neuron the input and output has a delay z^{-1} [105]. In the Hopfield neural network each neuron is of the same model, and x represents the neuron output at the current time, and y represents the neuron output at the next time.

w_{ij} is weight between neuron i and neuron j. So, in the time t, the output of x in the neuron i can be expressed as [106]:

$$x_i(t) = \sum_{j=1}^{n} w_{ij} y_j(t) \tag{2.1}$$

In the time $t+1$, the output of y in the neuron i can be expressed as [107]:

$$y_i(t+1) = f(x_i(t)) \tag{2.2}$$

Where [108] $f(\bullet)$ is the transfer function:

$$f(a) = \begin{cases} +1 & a \geq 0 \\ -1 & a < 0 \end{cases} \tag{2.3}$$

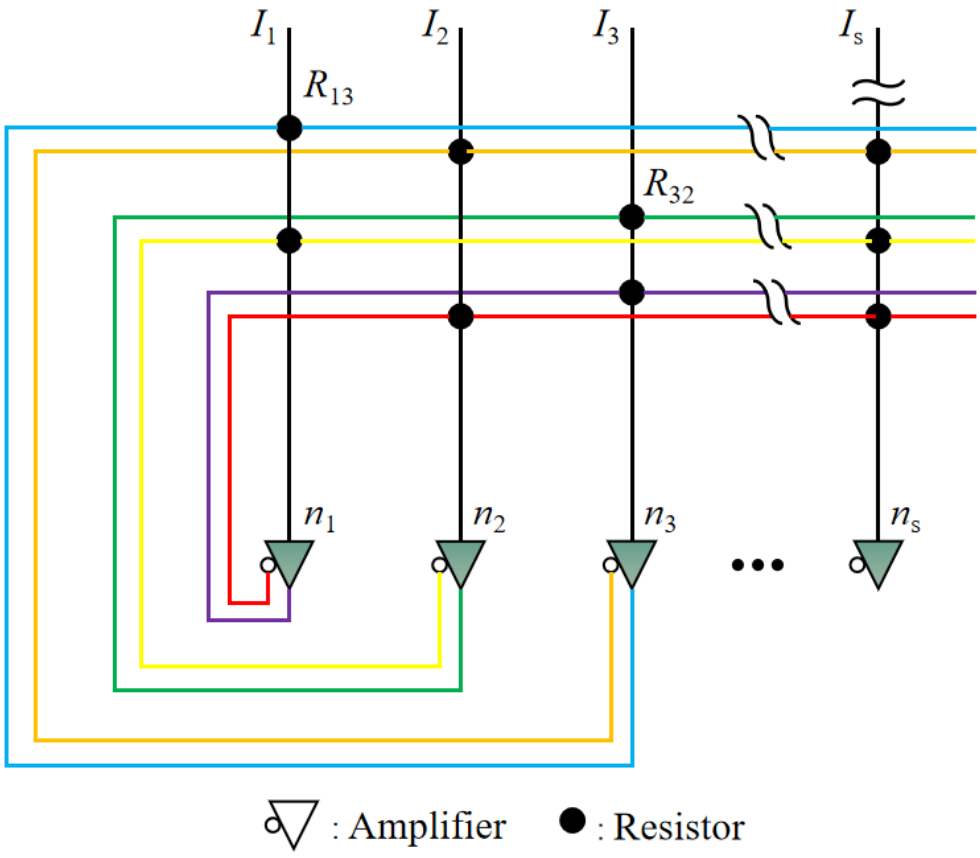


Figure 2.2: Circuit topology of Hopfield network.

Converting the network structure into a circuit topology that is shown in Figure 2.2, Hopfield neural networks are equivalent to amplified electronic circuits. Input data for each electronic

component (neuron), including constant external current input, and feedback connections that to link with other electronic components [109].

As shown in Figure 2.2, each electronic component is based on amplifiers. It includes a non-inverting amplifier and an inverting amplifier (depending on the positive and negative weight of the connection to select corresponding output needs) [110]. All states are feedback to the input of the circuit through the bias current I_s (S=*1,2,3,..., N*) [111]. At the connection point of each neuron has a resistor, which represents the impedance r_{ij}(r_{ij} = $1/w_{ij}$) connected to other neurons [112]. The constant w_{ij} represents the network weight between neuron i and neuron j.

For the bias current calculation is shown in the following [113]:

$$I_i = \sum_{j=1}^{s} \frac{x_i}{r_{ij}} = \sum_{j=1}^{s} x_j w_{ij} \tag{2.4}$$

2.4 Algorithm Learning Method

The neural network learning method can be classified by event sequence and time series [114]. This occurrence is due to the asynchronous timestamps, where the sequence of events depend on the network dynamics, and the time-series is deterministic [115]. For instance, in the RNN (recurrent neural network) architecture of the Backpropagation Through Time Algorithm (BPTT), Forward propagation is calculated each time in sequence order, while the Backpropagation delivers the accumulated residuals starting from the last time sequence number through the multiple layers [116]. In contrast, the event information can be captured based on the event sequence method, thereby, adjusting the time steps of the conditional intensity function [114]. On the other hand, the Hopfield algorithm training is based on the dynamics evolving in discrete time with time steps to discrete learning [117].

According to different learning environments, neural network learning methods can be divided into supervised learning and unsupervised learning [118]. In supervised learning, the data of the training samples are loaded to the network input end. Meanwhile, the corresponding expected output is compared with the network results to achieve the difference value [119]. So, it is to adjust the connection strength on the weights and converge to a certain weight after repeated training [120]. If the sample situation is changed, then weights can be modified to adapt to the new environment after training [121]. Neural network models using supervised learning include back-propagation networks [122], perceptrons [123], etc. In unsupervised learning, the network is directly placed into the environment without giving standard samples, and the learning and the working combined as a stage [124]. At this point, the learning rule transformation is based on the evolved equation of connection weight [125]. The classic example of unsupervised learning is the Hebb learning (HL) rule [126].

Hebb learning rules are the basis of artificial neural networks. The adaptability of neural

networks is realised through learning approaches [127]. It is a behaviour according to the environment changes, which is used to adjust weights and then to improve the system [128]. Hebb rules believe that the learning process occurs in the process of synapse between neurons. At the synapse, the strength of synaptic connections varies with the activity of neurons between synapses [129]. Some artificial neural network learning rules can be regarded as deformation by the Hebb learning rules [130]. Based on this, researchers have proposed various learning rules and algorithms to meet the needs of different network models. Effective learning algorithms enable neural networks to construct different target and object representations, which is through adjustment of connection weights. It is a distinctive information processing method that enables information storage and calculation is processing in network connections [131].

In 1949, D.O. Hebb proposed the "synaptic correction" on the learning mechanism of neural networks by psychology hypothesis [129]. Its means when neuron i and neuron j are excited at the same time, the connection strength between the two neurons should be enhanced. For example, in animal experiments when a bell rings, a neuron is excited, and at the same time the appearance of food will stimulate the other nearby neurons, then the connection between these two neurons will be strengthened, so that there is a relationship between these two things connected. On the contrary, if two neurons are always unable to stimulate simultaneously, the connection between them will become weaker [132].

The neuron stores the learned knowledge on the connection weights of the network. From the biological field, when the A cell's neuron axon is close enough to B cells, it repeatedly and continuously stimulates to cell B. At this point, the connection between the two cells will be strengthened. This means one or two cells in A or B will produce some kind of growth process or metabolic change, thereby enhancing the stimulation effect of cell A into cell B [133].

DHNN follows the Hebb learning method, which is a neural network learning rule proposed by Donald Hebb [134]. It is used to describe the behaviour of the connection relationship between neurons. Such as two connected neurons where both are activated; it can be considered that the relationship between the two neurons should be relatively close. Hence, the weight of the connection between these two neurons is increased. On the contrary, when one of the two neurons is activated, and the other one is inhibited, the weight between the two neurons should decrease.

The Hebb learning rule can be mathematically expressed as follows [135]:

$$w_{ij}(t+1) = w_{ij}(t) + c * e_i * e_j \tag{2.5}$$

The w_{ij} represents the connection weight of neuron j to neuron i, s_i and e_j represent the output of two neurons, 'c' is a constant representing the learning rate. If e_i and e_j are activated at the same time, that means e_i and e_j are positive, and the w_{ij} will increase. If e_i is activated and e_j is inhibited, that means e_i is positive and e_j is negative, then w_{ij} will decrease. This equation

shows that the change in weight w_{ij} is proportional to the product of the transfer function values on both sides of the synapse. This is an unsupervised learning rule, which does not require any information from object outputs [136].

After converting to DHNN, it can be simply expressed as the following weight calculation [113]:

$$W_{ij} = \sum_{s=1}^{n} \left(2V_i^s - 1\right)\left(2V_j^s - 1\right) \tag{2.6}$$

Hopfield neural network weight learning uses the sum of the outer-products method by Hebb rule. Given p pattern samples (n dimensional orthogonal vector) that is X^p , p = 1, 2, 3, ..., $x \in$ {-1, 1}^{n}, and the samples are orthogonal to each other. The $n > p$, then the weight matrix is outer product sum of memory samples [137].

Outer product sum can be expressed as [138]:

$$w = \sum_{p=1}^{p} x^p \left(x^p\right)^T \tag{2.7}$$

Weight binary computing is straightforward using a component-wise manner as follows: [139]: [140]

$$w_{ij} = \begin{cases} \sum_{p=1}^{p} x_i^p x_j^p, & i \neq j \\ 0, & i = j \end{cases} \tag{2.8}$$

At this point, w satisfies the symmetry requirement, and its need to check whether is an attractor on the algorithm.

Since p samples X^p, p = *1, 2, 3,* ..., x $\in$ {-1, 1}^{n} is pairwise orthogonal. The calculation according to the following: [140]

$$\left(x^p\right)^T x^k = \begin{cases} 0 & p \neq k \\ n & p = k \end{cases} \tag{2.9}$$

Due to $n > p$, the attractor x^p will be calculated by the following:

$$f\left(wx^p\right) = f\left[(n-p)x^p\right] = sgn\left[(n-p)x^p\right] = x^p \tag{2.10}$$

The weights computing workflow of Hopfield network [141] is presented in Algorithm 2.1

Algorithm 2.1: Hopfield Algorithm Workflow.

Require: p_n pattern samples (n dimensional orthogonal vector);

1: Set to the initial state of the network x=p;
2: Set the number of iteration steps;
3: Calculate the w weight of the network: w=$\sum_{i=1}^{n}\left[p^T p - I\right]$;
4: Since w_{ij}=0, subtract the unit matrix I;
5: Perform iterative calculation;
6: Until the number of iteration steps is reached or the state of the network is stabled, stop the network learning operations, otherwise iteration continues;

Ensure: Weight matrix of Hopfield Neural Network

However, when the network size is fixed, the number of memory mode is limited. To avoid association error rate on the associative memory, the maximum number of memory modes p_{max} that the network store ability of the capacity is $N/2\ln(N)$ [142]. It is related to the network size, algorithm and the distribution of vectors in the memory mode [143]. When designing a DHNN network using an outer product method, if the memory patterns all meet the pairwise orthogonal condition, the n-dimensional network can memorise at most n patterns [144]. Nonetheless, the pattern samples cannot all meet the orthogonality condition, and the information storage of the network will be greatly reduced for the non-orthogonal patterns. In fact, when the network size n is set, the more patterns to be memorised, which cause the high possibility of errors in associations. On the contrary, if it is required the lower error rate, which needs the smaller information storage capacity of the network. Follow the mentioned storage capacity calculation for $N/2\ln(N)$ [142], which is maximum number of associative memory modes with $0.14N$ for stable storage states. So, it will occur errors during the association on the network [145] when exceeds maximum storage capacity proportion [146]. The error result corresponds to a local minimum of energy, or a pseudo-attractor [147].

DHNN for associating memory function to store patterns that are constrained by the storage capacity, which means storage capacity proportional. This is the number of neurons required for each sample memory capacity. When the memory capacity exceeds the neuron storage capacity, the system will confuse the stored samples. It appears as errors in the neurons signal output. Relative to the steady-state of the correct memorised pattern, spurious states will also be generated during the learning process of associative memory in the Hopfield network. When the pattern samples of learning and memory do not meet the orthogonality condition, the dynamic update of the energy function may produce a false minimum. It will cause the neural network to return wrong or incomplete learning results. Therefore, the associative memory will be wrong

on the neuron; the result corresponds to the local minimum of energy and is called spurious attractors. The current configuration of the neural network is projected into the subspace spanned by the pattern vector with a learning rule. The work should ensure that the memory pattern computing meets the maximum pattern memory capacity (P_{max}) of a learning rule. It can avoid the Hopfield neural network generated spurious states of the pattern learning, which means the local minimum value in the energy function can be reduced. The storage capacity proportion is approximately $0.14N$, which is a max number of memorised patterns. This proportion is from the principle of neuron training, which follows the relationship of neurons N and patterns K in the below computing: [148]

$$K \leq \frac{N}{2lnN} = P_{max} \tag{2.11}$$

2.5 Applications

In daily life, character recognition has high practical application value in the postal [149], transportation [150] and document management process [151], such as the recognition of car numbers and license plates in transportation systems [152]. However, the images captured in the natural environment are often blurred due to the limitations of camera hardware [111], or uncleared by the font is occluded and worn out. At these points, the complete information of the character cannot be obtain and identification of noisy characters become a key issue [153].

At present, there are several methods for character recognition, which are mainly divided into a neural network [154], probability statistical [155], and fuzzy recognition [156]. The traditional character recognition method cannot recognize well under the condition of noise interference. However, the discrete Hopfield neural network has the function of associative memory, which is reasonable for anti-noise processing [157]. By applying this function, characters can be recognized and satisfactory recognition results can be achieved. Besides, the convergence calculation becomes fast for processing.

2.5.1 General workflow for Character recognition case study

The associative memory can be designed based on the discrete Hopfield neural network concept, and the network can recognize the 10 digits that fall in the range from 0-9. Furthermore, despite any disturbance by certain noise due to the specified range of numbers, still has a good recognition effect by network feedback. At this point, the network is composed of a total of 10 stable states that reach to 0-9 numbers. These numbers are represented by 10 x 10 matrices and are directly described by the binary matrix. In the 10 x 10 matrix, the number pixel is represented

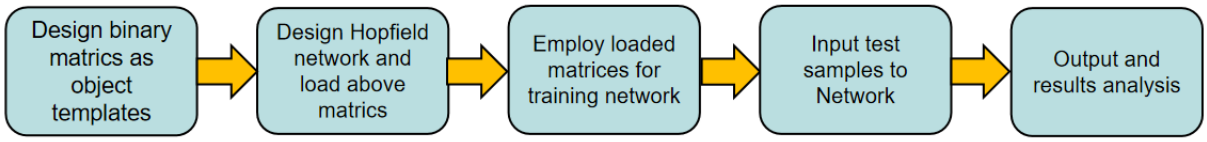


Figure 2.3: Hopfield network of the recognition application workflow.

by 1, and the non-number pixel is defined -1 as blank display.

The network through learning the above matrices on the function of associative memory is performed to achieve 10 steady states reaching 10 numbers. When new input data are applied to the network, the output of the network which is feedback, and used to determine the comparison of 10 steady states such as the object vector. Finally, the purpose of the whole operation is to achieve the effect of correct recognition. The Hopfield network of recognition application workflow is depicted as Figure 2.3

Standard Templates

In this study, standard templates are selected to convert into binarisation matrix. In this application of characters recognition, all the numbers from 0-9 are converted into 10x10 matrices.

Design Hopfield Network

In this case, network T is a matrix size of R with Q target vectors (the element must be binary of -1 or 1). According to the above matrices requirements that to design network architecture, it is a 10 x 10 matrix on number sample size, which means R = 10. Following the matrix size, the Hopfield network will contain 100 neurons, which one by one neuron match with a 10 x 10 matrix point.

Hopfield network Training

The learning processing adopts a neurodynamic method. The working process is based on the evolution of the state. For a given initial state, it evolves of energy decreasing manner, and finally reaches a stable state. The network starts from the initial state X (0) to multiple recursions, where its state does not change to form a stable state, which means the network is stable by $x_(t+1)$ = $x_(t)$.

When the X state is reached to a steady-state, then it is considered as an attractor. The attractors are determined by the final behaviour of a dynamic system. The system requirement of remembers information that is stored in different attractors. When the input sample containing part of the memoried information applied to a network, the evolution process of the network is resumed all the required sample information from the inputted part of the information. The aforementioned procedure is called the process of associative memories.

Input Test Sample to Hopfield Neural Network

In this section, test samples are converted to the matrix, and the size is the same as the training sample. This new test matrix is then placed into a trained Hopfield network, and the network output of the feedback, which is found to the closest of the object vector. In this case, the feedback will be in a range from 0-9 matrices.

2.5.2 Other applications

Based on the discrete Hopfield neural network, it has the function of associative memory. In recent years, many researchers have attempted to apply Hopfield network to various fields in order to replace conventional techniques to address the issues, such as water quality evaluation [158] and generator fault diagnosis [159], and have achieved considerable results by applying aforementioned method. For example, in the Internet of Things (IoT) applications, where multiple links fail and break the real-time transmission services, and due to this reason, the fault cannot be quickly located at that particular point [160]. The relationship between the fault set and the alarm set can be established through the network topology information and the transmission service, which is compatible with the proposed Hopfield Neural Network [161]. The built-in Hopfield algorithm of the energy function is used to resolve fault location, and hence, it is found that integration of aforementioned algorithm with the IoT will improve transmission services in smart cities [162].

However, the application will have a wider framework to suitable more applications in addition to limited applicability for the field. When the Hopfield neural network collaborates with some notable optimisation algorithms, not the network alone, that provides its associative memory stronger but also improves the application efficiency. For example, the existence of many pseudo-stable points in general discrete Hopfield neural networks, limit the proficiency of network. Therefore, a Genetic Algorithm (GA) can be considered for the discrete Hopfield network [163], and the global search capability of the Genetic Algorithm is used to optimise the steady-state of Hopfield associative memory function. So that makes the associative mode jump out of the pseudo-stable point, and then the Hopfield network maintains a high associative success rate under the condition of higher noise to signal ratio.

2.5.3 Application Comparison

The workflow details of Hopfield algorithm are similar to those of above in section B mentioned. However, the main difference is that varied projects need to design suitable templates for different objects to matching data-sets, and then to input object data into the neural network algorithm for learning. Appropriate templates will help the algorithm learning features more easily and improve processing accuracy.

The actual practice of Hopfield neural network involves a surprising number of scientific disciplines, which is the key technical areas it covers include Data Classification, Pattern Recognition, Image Recognition, Feature Match, Image segmentation, Image retrieval. The result of the investigation was reported as shown in Table 2.1. By comparing the applications in various fields, which can find that the application of the algorithm is a different implementation method. It is mainly depending on the design of the neural network weights that the algorithm uses the weights for associative memory to output results. Under the appropriate template of input data, which to corresponding output analysed and predicted results.

2.6 Future Plan and Challenges

For the future of neuromorphic chips, it is the key to break through the development direction of Von Neumann's structure limitations. Because the basic operations of neural networks are the processing of neurons and synapses [171], the conventional processor instruction set (including x86 and ARM, etc.) was developed for general-purpose computing [172]. These operations are arithmetic operations (addition, subtraction, multiplication and division) and logical operations (AND-OR-NOT) [173]. It often requires hundreds or thousands of instructions to complete the processing of neuron computing, making the low processing efficiency of the hardware inefficient.

Currently, neural computing needs a completely different design than the Von Neumann architecture [20]. The storage and processing are integrated into the neural network [28], whereas in Von Neumann's structure, there it is separated and realized respectively by memory and computational unit [174]. There is a huge difference between the two computing when using current classical computers based on the Von Neumann architecture (such as CPUs and GPUs) to run neural network applications. They are inevitably restricted by a separate storage and handling structure, which has caused a lower efficiency over the impacts. Although the current FPGA and ASIC can meet the requirements of some neural network application scenarios, a new generation of architecture like neuromorphic chips and integrated computing design will be used as the underlying architecture to improve the neural network computing in the long-term planning [175].

Among the above, ASIC is a kind of chip specifically designed for this special purpose. Compared to FPGA, it features stronger performance, smaller size, less power consumption, lower cost and more progress in developing hardware design. ASIC needs research, development time and high risks of technology marketing that have become a major obstacle to future promotion. However, some of its advantages such as good mold size, low cost, great energy consumption, great reliability, strong confidentiality, high computing performance and high computing efficiency have become the best choice for current formal nerve chips [176].

Another key point of neural computing is the challenge of holding computing nodes [177]. Generally, the nodes of bit computation are the conduction switches of the transistors [178].

Table 2.1: Algorithm extraction and use cases for Hopfield

Paper	Task	Data	Method	Problem Address
[158]	Data Classification	Seven parameters were selected from the 12 original parameters by Factor analysis. (*BOD5, CODMN, NH3-N, Cu, Zn, Pb, TN*)	Use positive and negative values (+1, -1) to represent whether water quality monitoring under different parameters is passed, which to set different memory patterns for water quality levels corresponding to the network.	Water quality evaluation
[164]		The network is constructed with different genes signatures as nodes, class 1 markers are shown in red, and class 2 markers are shown in blue to store into the model.	First, the expression value of the signature gene in the sample is discretized to +1, -1, after that the value is assigned to the corresponding node. Then, when all values of only one classe are +1 that to evolve the model. Finally, the sample is assigned to the convergent class.	Classify to transcriptomic data of Gene
[165]	Pattern Recognition	Four templates (8 x 8 binary valued pixels)	This network designs three types of neuron blocks, called saccade (S) block, hidden (H) block and output (O) block. The S block is a matrix representation of the position of the input mode neurons. O block of neurons represent the class of the input pattern. H block is a template for the target pattern class.	Speed up saccadic tasks
[164]		Eye's iris information into 150 × 150 pixels image input to network	The input image is first scaled to a 20x20 binarisation matrix, and then multiplied with the 400 x 400 value matrix. The input value of weight is obtained by performing a dot product calculation to the input vector.	Recognition of eye's iris
[166] [167]	Image Recognition	Convert the face image into a binarized matrix and store it in weights using a Hebb rule	Reduce the image to 60 x 60 size, and then transformation to 1D row vector store into matrix. And calculate the similarity as result by measured network output and stable image.	Retrieval of distorted face images
[168]	Feature Match	Representative edge-point that establish by scene edge and the edge of the model object	Feature matching is performed of a two-dimensional array. The row data means the features of the scene, and the columns means the features of the object model. The neuron output the similarity, which compared to the two features the scene and the object model.	Recognise and locate partially occluded 3D rigid objects
[169]	Image Segmentation	A random assignment of N pixels to M classes	Construct a network model of N × M neurons, it is the columns means cluster and the rows means pixel, which to classify the feature space.	Images segmentation for satellite images
[101]		Mammograms image of feature vector that containing gray value and edge information, input each discrete pixel position into N x N matrix.	The network deployed a binarized N × M neurons matrix to achieve image segmentation in the image. N is the number of sampling points of the initial contour, and M is the number of grid points on each grid line.	Tumor Boundary Detection
[170]	Image Retrieval	Subdividing image into L blocks, and extracting features that to generate feature vectors input to network	According to position to calculate weight matrix for each block of all images. Use the method of bootstrapping to get the final weight parameters.	Identify similar images by visual contents

However, formal neural computing requires computational nodes like neurons, which is the penalty for an alternative generalized alternative approach to achieve non-bit computing. This means that artificial synapses and excitement need improvement [179]. Nowadays, there are a lot of explorations on how to simulate or create synthetic synapses. Taken as a whole, for produced formal neuronal chips, industrial circuits are used primarily to simulate synapses that achieve formal neuronal computing [4]. But manufacturing processes and technical costs are high and production efficiency is low, causing neuronal simulation efficiency to low.

There are still many problems in the research of new materials for the neuromorphic hardware. In the future, researchers in the neuromorphic disciplines consider new materials belonging to neuromorphic computing can be found in place of transistors to new hardware design [180]. For example, the array composed of memristor that is a plastic element can be stored and processed to integrate for the neuromorphic hardware. It has a high switching current ratio, a light effective mass, a large adjustable band gap and large electron mobility, which provides a favourable basis for successful preparation of low-power neuromorphic hardware [181].

Eventually, the architecture, algorithm and programming scheme of adaptive neuromorphic computing is in a wide blank and a long way to reach a final goal that replaces to Von Neumann's structure in the artificial intelligence discipline. But the frontiers of neuromorphic computing knowledge are being pushed farther outwards over the time, and the future opens a bright prospects.

2.7 Summary

Although neuromorphic computing has gained widespread attention in recent years, however, it is still considered to be in the infancy stage. The existing solutions mostly focus on a single application at the hardware or software level, and majority of them are only suitable for handling limited applications. In addition, there are many software-based neural network applications that has been deployed, but hardware-based neural network design has been the key to the neuromorphic design. Convention neural network circuit implementation is thought of time-consuming and inconvenient. In order to apply a simple and fast design method to neural network hardware, which can optimise and manufacture neuromorphic computing systems, needs to systematically unificate the requirements of the software calculation process. Furthermore, it can process and improves the final software-level application indicators to quantify hardware attributes. Finally,a testable solution for a specification component can be achieved.

This study gives an overview of the work that is the hardware implementation field on neural networks. In addition, the work has also discussed the various techniques and methods employed in the overall progression and implementation of the Hopefield algorithm. In this regard, it is found that this algorithm has been extensively deployed in various disciplines based on feasibility and efficiency. Moreover, it has also highlighted the existing solutions for neuromorphic

computing which are mainly focused on a single application at the software-hardware level. In this regard, it is discovered that there is significant room for further improvement to achieve the most optimized design with a low computation process. From in-depth research, this chapter provides a significant step toward the hardware implementation of low-power neuromorphic processing systems using the advanced Hopfield algorithm.

Chapter 3

IMU Sensing-based Hopfield Neuromorphic Computing for Human Activity Recognition

Aiming at the self-association feature of the Hopfield neural network, it can reduce the need for extensive sensor training samples during human behaviour recognition. For a training algorithm to obtain a general activity feature template with only one time data preprocessing, this work proposes a data preprocessing framework that is suitable for neuromorphic computing. Based on the preprocessing method of the construction matrix and feature extraction; the work achieves simplification and improvement in the classification output of the Hopfield neuromorphic algorithm. It assigns different samples to neurons by constructing a feature matrix, which changes the weights of different categories to classify sensor data. Meanwhile, the preprocessing realises the sensor data fusion process, which helps to improve the classification accuracy and avoids falling into local optimal value caused by single sensor data. Experimental results show that the framework has high classification accuracy with necessary robustness. Through the proposed method, the classification and recognition accuracy of the Hopfield neuromorphic algorithm on the three classes of human activities is 96.3%. Compared with traditional machine learning algorithms, the proposed framework only requires learning samples once to get the feature matrix for human activities, complementing the limited sample databases while improving the classification accuracy.

3.1 Introduction

Human activities are movement postures with various features that human beings do while at study, work, production and other situations [182], including regular or irregular movement patterns and states such as running, walking, standing, sitting, lying, etc. [183]. As society is consistently changing; science and technology are advancing rapidly, while artificial intelligence is

becoming more and more popular. Research methods for human activity recognition are continually evolving, and it has become the primary technology for many applications such as health care [184] [185], human-computer interaction [186] and robotic control [187]. Meanwhile, with the development of mobile computing and sensing technology, wearable sensor signals have become common data [188]. When an object stays in a different position, performs different activities, or performs different gestures, it will have different effects on the signal characteristics of surrounding sensors [189].

At present, the human activity recognition of wearable devices has strong applicability and use-value. For instance, the intelligent nursing scene system for the elderly detects and analyses the actions in real-time through portable sensors. It can determine whether the elderly has eaten, medicated, or carried out the minimum exercise. To ensure the health and safety of the elderly, information around the amount of exercise and detection of abnormal actions such as falling [190] is vital. For the human-computer interaction system, through the recognition of human activity, timely and accurate responses are highly vital for the different actions [191]. For the robotic control, the rehabilitation training within hospitals by identifying the degree of standardization of movement behaviour, the recovery can be evaluated to provide better rehabilitation guidance [192].

Compared with the activity monitoring of external means such as radar [193] and cameras [194]. The advantages of wearable sensors are more concentrated in open scene applications. The user is not limited to a specific monitoring area, and can freely enter and exit places, and carry-on behaviour without the obstacles of blind spot recording and capturing [195]. Therefore, human activity recognition based on wearable sensor signals overcomes the drawbacks of traditional methods and becomes a promising technology for future mobile computing applications. It will play an essential role in intelligent applications such as smart health [196], smart space [197] and behavioural analysis [198].

Currently, machine learning methods, especially deep learning algorithms, have been widely used for Human Activity Recognition (HAR). However, due to existing machine learning, especially Convolutional Neural Network (CNN) models, induces overwhelming training data collection overhead. The problems caused by massive samples' algorithm requirement to learning for the feature are not suitable for small datasets, which greatly reduced machine learning algorithm practicability. Such as Ayman *et al* [199] realized multi-activity recognition by Support Vector Machine (SVM) algorithm, with preprocessed activity data, and Chavarriaga *et al* [200] used cluster recognition activities by Naive Bayes (NB) and K-Nearest Neighbors (KNN) algorithm. However, these are all traditional machine learning algorithms. Its algorithm capability limits its robustness, and since it focuses on a single operation, it lacks consistency and shows poor recognition effect in complicated and diverse human movement scenes. Lately, there has been a growing trend in improving recognition accuracy and applicability by deep learning. For example, Terry *et al* [201] proposed a CNN model that can better classify to the nonlinear

model, to avoid the hysteresis and time-varying behaviours in sensors with higher frequency strain rate. Francisco *et al* [11] uses the LSTM algorithm to improve the CNN structure and introduces the concept of time series of activity sensor data. This implementation is a deep learning framework composed of convolutional layers and Long Short Term Memory (LSTM) recursive layers, which can automatically learn feature representations and build-time dependencies between activity models. However, the costs of training samples for deep learning methods are not friendly to small datasets and it depends on a wide variety of training datasets to achieve correct activity recognition. The aforementioned factors cause uncertainties, resulting in high computational complexity for training and large dataset requirements for datasets. With limited sample datasets, the classification accuracy cannot meet the requirements [8]. Researchers have also implemented the unsupervised learning method to clustering analysis limited datasets; however, the idea of learning transfer approaches complex algorithms, soft labeling examples increases which also leads to the design missing feature calculations [202]. Following our previous study [203], reducing the need for training data without losing classification accuracy has been a significant problem for researchers in the science community.

In this study, a novel framework has been presented that is based on the Hopfield neural network for wearable multi-sensor data knowledge fusion pre-processing and classification. The Hopfield neural network can be used for a single task, such as human fall recognition, as seen in earlier studies [204]. Following to extend the research on the data fusion of multiple sensors with the Hopfield neural network to a series of human movements. The work builds a dimensionality reduction matrix, place it into the SVD (Singular Value Decomposition) algorithm for feature extraction, also termed as data preprocessing method. Then, it follows the Hebbian learning method for training the Hopfield neural network based on the corresponding human activity feature matrix. Finally, the work achieves multiple human activities recognition result output by similarity calculation. Rather than considering separate single activity data in isolation, it studied the comprehensive data of three activities for comparative output to obtain significantly different feature templates for each human activity. It ensures the Hopfield algorithm can learn the related activities through the corresponding feature templates. Hence, reducing the number of training samples required while maintaining a high classification accuracy.

The rest of this chapter is organized as follows: Section 3.2 presents data collection, preprocessing and feature extraction method. Section 3.3 shows the implementation, evaluation and results. Section 3.4 discussed and compared the project with others. Finally, section 3.5 focuses on conclusions and future work.

3.2 Data PreProcessing and Feature Extraction

The dataset used in this work comes from Li *et al* [205] who works at the University of Glasgow, and the dataset can be requested from the authors. It collects multiple activities of 20 volunteers

aged between 22 and 32, where each activity is repeated three times. Although the sample of this dataset is limited, it is still in the top 3% in terms of the number of subjects compared with other wearable human motion analysis works [205] [206] [207]. All activities are collected by IMU (Inertial Measurement Unit) sensors worn on the wrist. In work, the dataset is 60 data for each activity, and only requests one data in training as one-shot learning.

Ortiz [208] defines how to divide human activity classes. Short events activity generally refer to fast transitions between human body postures and behaviours. The basic activity lasts longer than short events activity, and are usually the most frequently occurring episodes during daily human life, such as sitting, standing, running and other behaviours. Meanwhile, these are also further divided into static activity and dynamic activity, according to the state of the human body. Moreover, there are two ways for complex activity; one is a combined activity that is aggregated through a variety of basic activities, and the other includes multiple users participating during the activity. Therefore, complex behaviours are usually divided into two further strains, i.e., multi-activity and multi-user. Table 3.1 shows the details of selected activities based on the activity principles of duration, complexity, type etc., the work selects three classes and activities data for experimentation which includes: Fall, Carry and Bending to tie shoelaces. These activities have been purposefully selected into different types and include a potential class that can be misclassified as falls, such as the complex activity of bending to tie shoelaces, it involves changing the position of the bow and the complex motion of the hand. This helps to test the robustness of neuromorphic computing classification. Falls a severe impact on the health of the elderly or the disabled, it is incredibly essential to achieve reliable recognition of low false positive alarms and low missed detections.

Table 3.1: Activity selected and details

Duration Complexity	Activity Type	Activity	Detail	Activity Times	Collected Count
Short Events	Transitions	Fall	Fall over while walking	5 seconds	60 times
Basic Activities (BAs)	Dynamic	Carry	Walking while carrying an object	10 seconds	60 times
Complex Activities (CAs)	Multi-activity	Bending to tie shoelaces	Stand first, then bending to tie shoelaces	5 seconds	60 times

IMU (Inertial Measurement Unit) sensors contain three specific sensors: gyroscope, accelerometer and magnetometer. The gyroscope sensor measures the angular velocity, that is the speed at which the object rotates. It multiplies the speed and time to get the angle that the object rotates in a particular period. It captures the angle between the vertical axis of the gyrorotor with

the device in the three-dimensional coordinate system and then calculates the angular velocity. Finally, it judges the movement state of the object in the three-dimensional space (X, Y, Z axis) through the included angle and angular velocity.

The accelerometer measures the gravitational acceleration of the object. It can sense acceleration in any direction, expressed in the three axial (X, Y, Z axis) acceleration magnitude and direction. The magnetometer is used to record the strength and direction of the magnetic field, subsequently locating the orientation (heading direction) of the device. The principle of the magnetometer is similar to the compass, which can measure the included angle between the current device and the four directions of the south, east, north and west. Finally, the gyroscope sensor records the rotation of the device itself, the accelerometer sensor records the force exerted on the device, and the magnetometer positions the device's orientation.

All of the activities will get a 9-axis data by each sensor of $gyroscope_{x,y,z}$, $accelerometer_{x,y,z}$ and $magnetometer_{x,y,z}$ axis.

3.2.1 Sensor Fusion by Quaternion and Euler Angle

The accelerometer sensor's data provides an absolute reference for the horizontal position, but can not provide the azimuth reference. The angle is not accurately measured by the accelerometer sensor alone, which can be addressed by data fusion with a gyroscope sensor. While the magnetic field information is in a stable environment, the sensor achieves the same magnetic field intensity in each activity. Although, it can help correct the gyroscope angular velocity parameter. Therefore, the appropriate algorithm can fuse the data from various sensors to make up for the shortage of a single sensor in calculating the accurate position and direction, thereby achieving high-precision activity recognition. Meanwhile, to reduce the computational complexity of multi-dimensional data, the signal processing needs to consider dimensionality reduction to simplify the calculation. Here, the Quaternion [209] and Euler Angle (EA) [210] computing is used for the sensor data fusion. Depending on the accelerometer, magnetometer and gyroscope sensors' data, the work updates the Quaternion output and then converts it to Euler Angle. Finally, the 9-axis data of three sensors are fused and reduced to a 3-dimensional dataset. This computing is presented as follows explanations.

Following the raw 9-axis data calculated into the coordinate system from the three sensors, the respective three components of the gravity vector and the magnetic field vector in the world coordinate system can be obtained. Subsequently, the accelerometer achieves the three components of the gravity vector in the sensor coordinate system. And finally, the magnetometer measures the magnetic field vector in the sensor coordinate system. Let $\mathrm{d}E$=[0,de_x,de_y,de_z] be the vector coordinates in the world coordinate system. Let q be the sensor attitude. So that it can be used q^{-1} to transform $\mathrm{d}E$ into the vector coordinate $\mathrm{d}S$=[0,ds_x,ds_y,ds_z] in the sensor coordinate system. The gyroscope sensor data achieves the angular velocity (ω) by integrated computation, and then follows the accelerometer and magnetometer attitude ($\mathrm{d}S$) to update the

q as Quaternion at each sampling interval by equation as follows:

$$dS = q^{-1} * dE * q \tag{3.1}$$

where f is the device attitude fitting error, which can be calculated from the current attitude q and the attitude dS.

These are following the Equation 3.2 [211] to achieve Equation 3.3 [211] which is an integrated computation. The $\frac{1}{2}q_t * \omega\Delta t$ is calculated by angular velocity, and $\frac{\nabla f}{\|\nabla f\|}$ is calculated by accelerometer and magnetometer values. The weight β represents the error of the angular velocity.

$$\begin{cases} q_{t+1} = q_t + \frac{1}{2}q_t * \omega\Delta t \\ q_{t+1} = q_t - \mu\frac{\nabla f}{\|\nabla f\|}, \end{cases} \tag{3.2}$$

$$q_{t+1} = q_t + \frac{1}{2}q_t * \omega\Delta t - \beta\frac{\nabla f}{\|\nabla f\|}\Delta t \tag{3.3}$$

Since combined rotation and vector transformation are frequently used in attitude calculation, the Quaternion can simplify the calculation and smoothen interpolation. Quaternion parametrisation (of the rotation) only needs a four-dimensional quaternion to express arbitrary rotation, which is more efficient than the matrix method [212].

Euler Angle is used to define the rotation of the device in space. It rotates as a fixed angle to sequence around the coordinate system of Z, Y and X axis. Using the Roll angle (ϕ), Pitch angle (θ) and Yaw angle (ψ) to represent the rotation angle around X, Y and Z axis on the coordinate system of the object [213]. The attitude matrix determined by Euler Angle is the product of the cubic coordinate transformation matrix.

For the sensor fusion, this work first converts the accelerometer and magnetometer sensor's data to the coordinate system, and then calculates the deviation with the corresponding reference gravity vector and magnetic field vector. This deviation is used to correct the output of the gyroscope, after that it uses the gyroscope data to update the Quaternion to the 4-dimensional feature expression of activity. Finally, the work follows the Euler Angle computing to convert the Quaternion. Figure 3.1 shows 9-axis sensor's raw data (Figure 3.1a) which is dimensionally fused and given to the 3-dimensional Euler Angle feature (Figure 3.1b), without affecting the expression of the corresponding action.

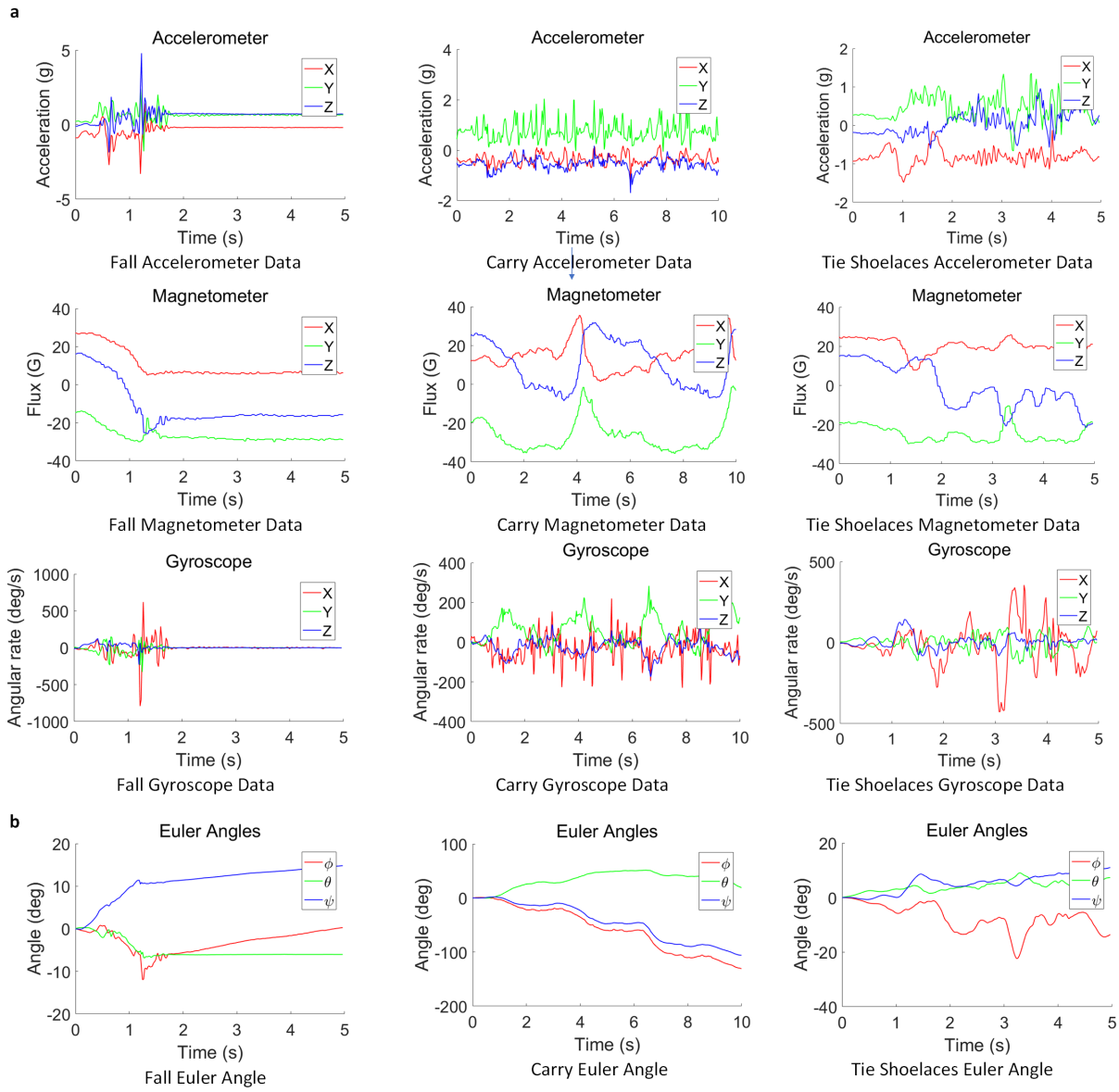


Figure 3.1: Three activities (First column:Fall, Second column:Carry and Third column:Tie Shoelaces) calculated from IMU raw sensor data to Euler Angle feature: a, IMU 9-axis sensor data. b, Euler Angle Feature

3.2.2 Feature Extraction by SVD: Singular Value Decomposition

When performing dimensionality reduction processing on a high-dimensional vector space model, there are usually two methods that are feature selection and feature extraction. Feature extraction has changed the original feature space, which is created and refines new features on the basis of the original features. It converts the original data into a set of apparent physical features, or kernel features. Feature selection only filters the original features that are selecting a set of statistically significant features from a set of predefined features, and hence it can also be called feature subset selection. Therefore, feature extraction is different from feature selection

solution. Therefore, feature extraction is different from feature selection solutions. The feature extraction computing transforms the data into a new feature space. In contrast, the feature selection maintained original features from the data. Its extract the abstract features contained in the original features according to a specific algorithm.

The simple way to obtain a low-dimensional feature subspace expression is to perform a linear transformation on the original high-dimensional feature space. Here, the chapter chooses the SVD (Singular Value Decomposition) [214] to extract features of different human activities. It is a crucial matrix decomposition in linear algebra, which is based on the SVD method to calculate the feature spaces and feature vectors of the matrix, so that low-dimensional features can represent the original data. SVD used three submatrices U, Σ, V^T to decomposes the original matrix as following: [215]

$$A_{m*n} = U_{m*m}\Sigma_{m*n}V^T_{n*n} \tag{3.4}$$

The U is an $m*m$ size matrix, Σ is an $m*n$ size matrix, and V^T is an $n*n$ size matrix, which is the conjugate transpose of V. U and V^T are two sets of orthogonal vectors, which means the work gets two sets of orthogonal basis. The A matrix as an original data, which rotates a vector from orthogonal basis vector space of V^T to orthogonal basis vector space of U, with a certain scaling in each direction. Then it achieves the scaling factor as singular values of Σ. Σ being a diagonal matrix, which should sort the diagonal values in descending order. It is corresponding to the singular values of the original data, and the matrix values are all 0 except the values on the main diagonal, where each element on the main diagonal is called a singular value.

The singular value can be used as a feature to express a rectangular matrix or a singular matrix, which can be regarded as mapping from one feature space to another feature space. Whereby, the project complete the feature extraction of the Euler angle feature to each activity matrix pattern. Figure 3.2 shows the Euler Angle data through 3 times of SVD computing for dimension reduction and feature extraction that is scaled to a 5*5 feature matrix to distinguish different human activities. Figure 3.2a is a construction matrix from Euler Angle value, and the matrix size is the product of time and frequency (where sensor's sampling frequency is 50Hz). The roll angle (ϕ) data is loaded in the first row, the pitch angle (θ) data is put into the first column and the yaw angle (ψ) data on the diagonal of the construction matrix. Then, Figure 3.2b, 3.2c and 3.2d are following the three times SVD processing for dimension reduction and feature extraction on the construction matrix. Carrying out matrix decomposition through SVD computing, the large matrix can be decomposed into a product of three small matrices. The middle Σ matrix of the upper left corner are the most important feature values, which represents the main component of the matrix decomposition. At this point, the work keeps the upper-left data as a new matrix, and the rest of the positions are regarded as zero for data dimension reduction. It still retains the significant features of the data. After three times SVD computing, the Hankel computing to compress the singular value matrix into 5*5 as the following calculation [216]:

$$H(i,j) = H(i+1, j-1) \tag{3.5}$$

For Hankel computing, it specifies the first column is the first n values of the diagonal on the matrix and the last row is the last m values of the diagonal on the matrix ($n \times m$ = final matrix size). All other elements in the Hankel matrix are equal to the adjacent position of the lower-left corner. Figure 3.2e shows the singular values obtained after Hankel computing. Finally, Figure 3.2f converts the Hankel matrix into Binary Pattern by the manual threshold value, which is significant to distinguish the three activities as different feature matrices.

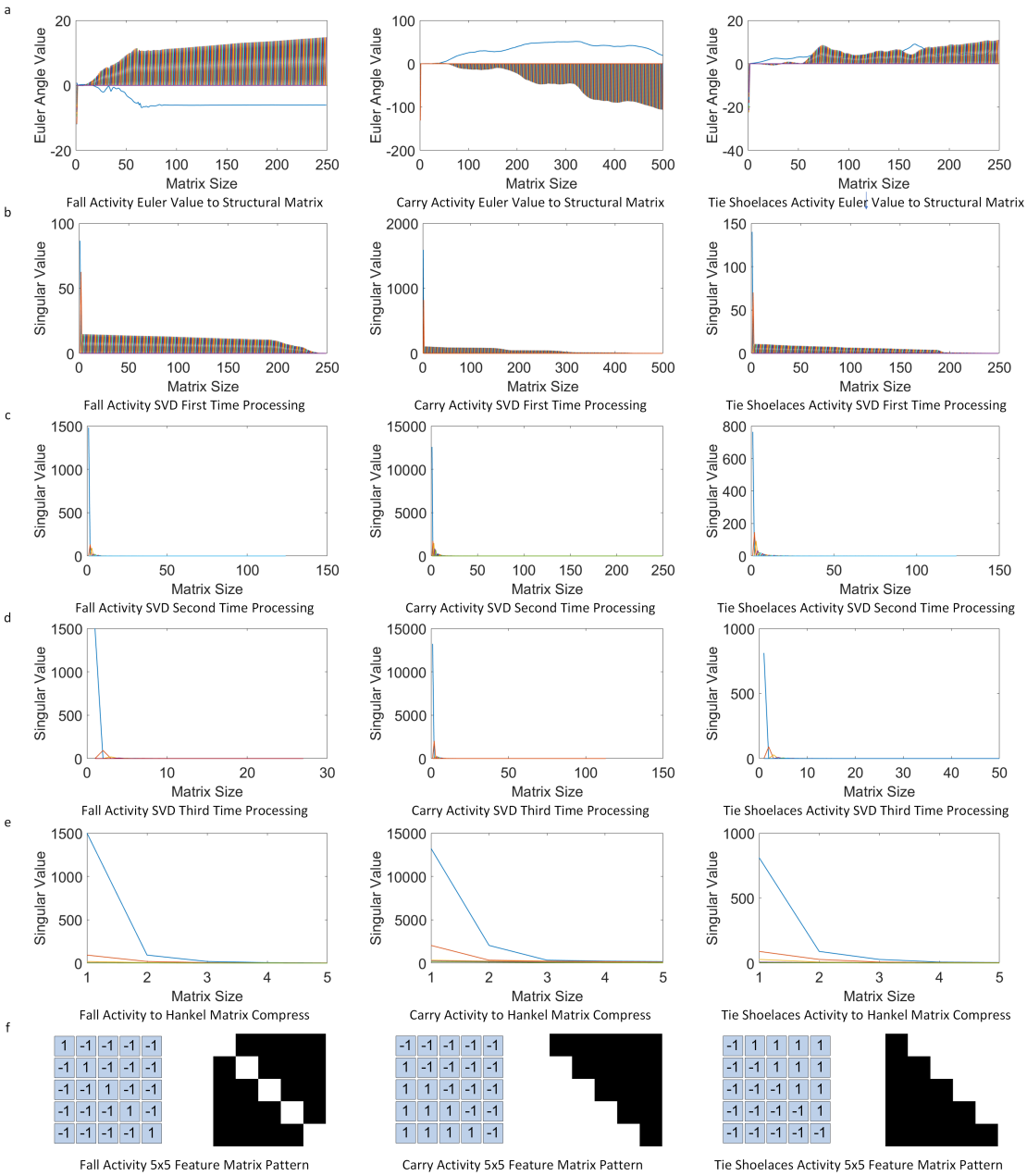


Figure 3.2: The workflow for Euler Angle Value to Feature Matrix of Activity Binary Pattern

The achieved feature matrix of each human activity can improve the accuracy of recognition, construct a faster and less expensive classification model, and a better understanding and interpretation of the model than Euler Angle. Meanwhile, the binarised pattern is convenient for further transmission to the Hopfield neural network to a training classification model.

3.2.3 Hopfield Neural Network: Binary Pattern

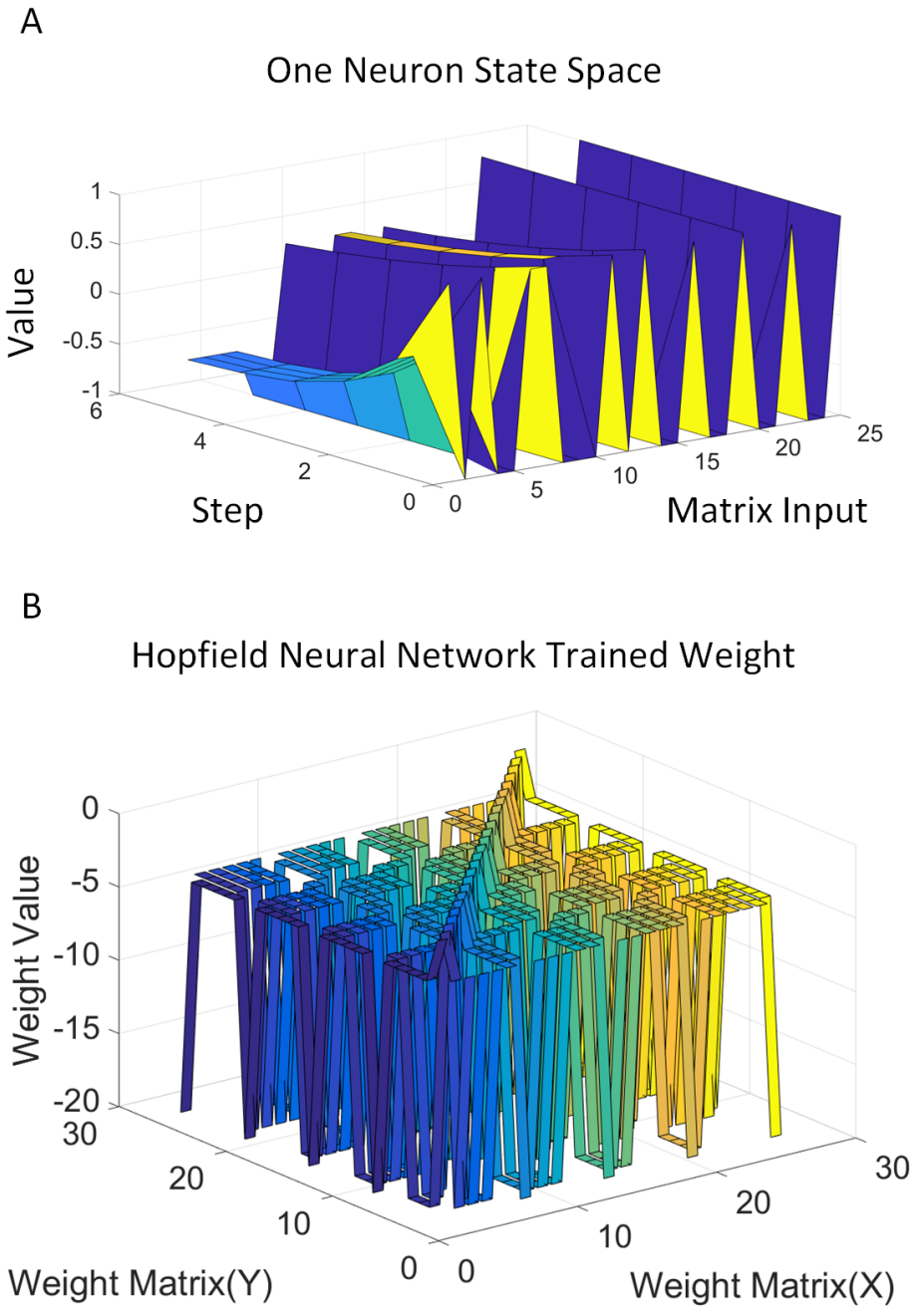


Figure 3.3: Discrete Hopfield Neural Network Neuron and Weight trained by three activities

The Hopfield neural network training phase is illustrated in Figure 3.3 with the three activities. Figure 3.3a depicts one Hopfield neural network state-space neuron that processes the training. The weight of DHNN is calculated using a feature matrix (5*5 feature pattern of human activity) and the constructed Hebbian learning law, and it indicates that the neuron has reached a stable state after the 5 steps of training. Figure 3.3b displays the entire DHNN weight output after the algorithm trained the three preprocessed binary patterns of activities. Since the DHNN has the smallest space, it memorizes three patterns for Fall, Carry and Tie Shoelaces activities: 5*5=25 (storage capacity equal to $0.14N$) [142] neurons, hence the above data preprocessing stage is used to obtain the 5x5 feature matrix as the behavior pattern.

3.2.4 Cosine Similarity work for Recognition

When the Hopfield algorithm is used for associative memory, it only needs to give part of the information of the input mode, and then the algorithm can associate itself to the complete output mode. At this point, it is fault-tolerant, which is conducive to the input of the sensor data at different time frames into the algorithm. Finally, the work obtain a usable matrix output while reducing the time-sensitivity of the algorithm with data. After the output the matrix is obtained by the Hopfield algorithm, furthermore, the vectors representing each matrix feature can be calculated by the cosine of the angle of the inner product space. These measure the similarity between the matrices, and achieves the classification of the data. Cosine similarity [217] is based on cosine distance computing, which is a measure of the difference between two individual matrix data. It uses the cosine of the angle calculation between two vectors in vector space. When the cosine value is closer to 1 it implies that the angle is closer to 0 degrees, meaning there is a relatively higher level of similarity between the vectors. This calculation method of cosine distance is suitable for n-dimensional vectors. Following the Equation 3.6, A and B are two n dimensional vectors (n=1,2,3,...), and i is components of vector A and B. The cosine of the angle θ between A and B can be calculated as follows: [218]

$$\begin{aligned} cos(\theta) &= \frac{A \cdot B}{\|A\| \cdot \|B\|} \\ &= \frac{\sum_{i=1}^{n} A_i * B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} * \sqrt{\sum_{i=1}^{n} (B_i)^2}} \end{aligned} \tag{3.6}$$

In this regard, all sensor signals will undergo the previous feature extraction to output different feature patterns. In the subsequent neural network, except for the trained 3 activity models, any other data cannot activate the neurons. Then, the output of the Hopfield neural network linked to the cosine similarity, achieves human activity recognition. Three trained activities will get a high probability of similarity output, and other data signals will be output low cosine simi-

larity since the neurons are not activated. Finally, a classifier based on neuromorphic computing achieves effective human activity recognition as elaborated in the following Results Evaluation Section.

3.3 Results Evaluation

The Figure 3.4 is the workflow of the algorithm framework, which shows each working step from the sensor's raw data into the algorithm until the final output of the classification result. According to the profile report of the Pie Chart in Figure 3.4, it depicts the relative importance of each function in the entire system and explains how to distribute work and coordinate with one another. The system's primary focus is on Hopfield neural network, which explains that the recognition work is the most essential and complex calculation in the system. At this point, the first challenging is how to make a suitable feature map input as a training template on the Hopfield neural network. The system's second challenging task is SVD processing, which mainly depends on the Discrete Hopfield Neural Network input requirement. Since the neural network uses binary information instead of raw data, the SVD algorithm can perform data pre-processing for feature extraction, resulting in a binary feature map with activities. It requests feature extraction computing to raw sensor data and then converts it into a binary feature matrix controlled by a threshold value. The following are additional details:

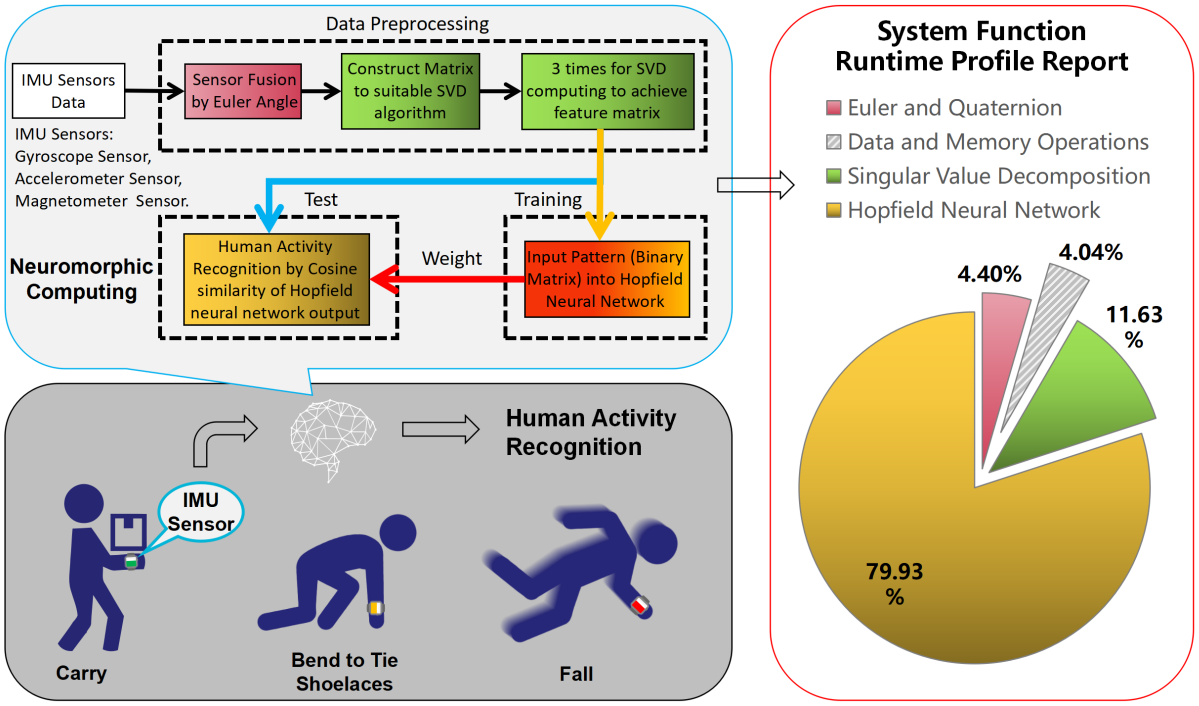


Figure 3.4: Workflow for Sensor data to achieve human activity recognition

Sensor Fusion: The purpose of this step is to fuse the raw data of the sensors. Using Eu-

ler Angle to reduce the 9-axis data of the three sensors with increasing data differentiation of activities.

Construction matrix: Based on the Hopfield neural network for memory storage of three activities, the processing on 25 neurons of 5x5 binarised matrix input becomes the optimal design. The fused data can then be constructed as a matrix to achieve feature extraction and binarisation pattern. The feature matrix is output to the template of different activities to suitable Hopfield neural network processing.

SVD algorithm: Depending on the three times singular value decomposition, it is mainly working for dimension reduction and feature extraction. After the computing, there are three activities that can achieve totally different feature matrices on the 5x5 size. This can then be efficiently input into Hopfield neural network training.

Hopfield neural network: This step is to obtain the advantages of neuromorphic computing by loading the designed binarised feature matrix into the Hopfield algorithm. The neural network weight is calculated by standardised activity data without the massive training samples, and the three activities memories are stored in neurons.

Classification function: During the test, all the input signal work with the data preprocessing is fed into the Hopfield algorithm achieving our desired output. At this point, the work calculates cosine similarity between the output signal of the neural network and the standardised feature matrix. The Hopfield algorithm acts as a filter blocking the data outside the memory and only outputs the activity data that satisfies the memory. Through the threshold of cosine similarity, the accuracy confusion matrix can then be obtained to complete the human activity recognition.

In the algorithm, feature extraction determines the input of the network, which is one of the most critical aspects of the framework. The extracted features must reflect the category differences of the object data. It needs excellent tolerance to the randomness and noise of sensor data. Meanwhile, it should be convenient to add new categories as a flexible framework. Depending on the specific project, it can include steps such as removing unique attributes, processing missing values, attribute encoding, data normalisation, feature selection and principal component analysis. The data for this project is processed using coarse grid feature extraction and binarisation, which is suitable for the DHNN algorithm training.

Table 3.2: Confusion Matrix of Classification Precision

		Target Class		
	%	Fall	Carry	Tie
Output Class	Fall	94.4%	0	5.3%
	Carry	5.6%	100%	0
	Tie	0	0	94.7%

Table 3.3: Confusion Matrix of Classification Recall

		Target Class		
	%	Fall	Carry	Tie
Output Class	Fall	94.4%	0	5.6%
	Carry	5.6%	94.4%	0
	Tie	0	0	100%

The two confusion matrices representing the algorithm's classification presents the results for the three classes of human activities. The confusion matrix summarises the results of the dataset in the matrix form by the real category and the predicted category. The columns of the matrix represent the true values, and the rows of the matrix represent the predicted values. Table 3.2 is the precision results of the algorithm classification, which indicates the percentage of samples that are classified to be positive. That is calculated by True Positive (TP) / (True Positive (TP) + False Positive (FP)). Among the three human activities, the classification of 'Carry' activity is correct, which indicates the algorithm's classification has the highest success rate for 'Carry' activity, and it will not misidentify other activities as 'Carry'. Table 3.3 is the Recall rate of the classification, which represents the percentage of correctly classified positive samples in the truly positive sample and is calculated by True Positive (TP) / (True Positive (TP) + False Negative (FN)). The result shows that the algorithm is most sensitive to shoelace tie activity. Finally, the algorithm classification accuracy calculated as (True Positive (TP) + True Negative(TN)) / (True Positive (TP) + False Positive(FP) + True negative(TN) + False negative(FN)), for the three human activities comes to be 96.3%.

Table 3.4 shows a comparison against traditional machine learning algorithms and proves that better results are achieved through the proposed neuromorphic algorithm under a suitable feature extraction model. As discussed before, this depends on the associative memory function of the Hopfield neural network and generates activity weights after learning the one training sample, based on the SVD and computation of the feature matrix. Li *et al* [205] based on traditional machine learning algorithms work on the same dataset, which achieved classification result by Support Vector Machine (SVM) and Artificial Neural Network (ANN). Although existing machine learning a good recognition effect has been obtained, there is overwhelming training data collection overhead. Spiros *et al* [219] propose a method from a change detection algorithm along with deep learning, which is wide generalization by operation upon raw sensor accelerometer signal. Then, use it as an activity feature input to the CNN algorithm to achieve fall recognition. Ashry *et al* [220] following the Bi-LSTM (Long Short Term Memory) algorithm to input autocorrelation, median, entropy, and instantaneous frequency as stream features that achieves IMU sensing human activity recognition. However, as existing machine learning, especially deep learning a good recognition effect has been obtained, there induces overwhelm-

ing training data collection overhead. Taylor *et al* [221] tried some neural network method on the same activities for recognition. Meanwhile, they used the Fast Fourier Transformer (FFT) method to improve the classification accuracy of limited datasets, which is based on Orthogonal Frequency Division Multiplexing (OFDM) for 64 points of FFT producing 64 frequency carriers. However, these use Universal Software Radio Peripheral (USRP) radar signal data, and the method cannot extend to sensor data. Comparing their work, we believed that the recognition findings are preferable, demonstrating that Neuromorphic computation is effective in recognizing human behavior. Furthermore, our proposed framework has greater robustness and can adapt to more different types of matrix data. Meanwhile, the Hopfield neural network benefits from only one training sample to achieve good classification accuracy, which addresses the limited datasets problem.

3.4 Discussion

Simple feature maps of different classes are realised by feature extraction for sensors data, while a discrete Hopfield neural network is utilised to compensate similar data that achieves robustness in human activity features. For example, some drift in the angle calculations of the IMU sensor sometimes makes it impossible to keep clear of errors from acquiring activities data. Traditional deep learning has weak learning ability and often requires massive data and repeated training to generalise sufficient accuracy. Such deep neural networks are usually good at learning features from high-dimensional data; however, they require training with a large sample dataset.

In contrast, when SVD pre-processes the new input data, the calculated activity features are improved by the associative memory function of the Hopfield neural network, which then disassociates corrupted data and outputs correct activity feature information. This makes it one-shot learning for different activities to achieve prior knowledge and form a knowledge structure and is based on associative memory to expand the data generalisation processing. The resulting associative memory helps in generalising data to quickly match the correct activity feature map and, finally, accurately classify human activities. One-shot learning was completed to implement HAR for limited datasets, and high-precision results were obtained. Meanwhile, randomly selecting a single training sample that is verified the robustness of the designed algorithm, and is convenient to promote more datasets.

Based on the neuromorphic computing of the Hopfield neural network to realise one-shot learning, the associative memory of data works somewhat like the human brain; it aims to achieve feature information about the object classes from one training sample. Randomly select one data in the dataset as the training sample for the Hopfield Neuron Network, And then the feature matric obtained after the feature extraction calculation, which is can be mapped into the feature template of the corresponding activity. The associative memory function of the Hopfield Neuron Network with the similarity of the distance calculation that is composition Learn the

Table 3.4: Comparison table with other machine learning methods

Project	Algorithm	Feature Extraction	Training Method	Activity Classes	Accuracy
Our Work	Hopeld Neural Network	SVD preprocessing Euler Angle data to achieve activity binary pattern	One-Shot Learning: 1 Sensor sample to generate a standard pattern for each activity	3	96.3%
Li *et al* [205]	Support Vector Machine	Sequential Feature Selection (SFS) for 30 features of IMU Sensor data	Machine Learning: Using a 70% Sensor dataset as training data (20 volunteers * 3 repetitions * 70% = 42 training samples for each activity)	10	92%
	Articial Neural Network	Sequential Feature Selection (SFS) for 30 features of IMU Sensor data, and 10 features for the Radar data	Machine Learning: Combine the Sensor and Radar dataset and then use 70% samples to training (20 volunteers * 3 repetitions * 70% = 42 readings for each Sensor and Radar, there are totally 84 training samples for each activity)		96%
Spiros *et al* [219]	Convolutional Neural Network	Wide generalization by operation upon raw sensor accelerometer signal	Deep Learning: In the MSB dataset, the number of observations for the raw data was 294,679 of which 8,516 were labeled as falls	1	92%
Ashry *et al* [220]	Bi-LSTM (Long Short Term Memory)	Stream features for autocorrelation, median, entropy, and instantaneous frequency	Deep Learning: The total number of streams in the dataset is 470 samples, and 70% of the data are randomly chosen for training while the remaining 30% is used for testing	10	91%
Taylor *et al* [221]	K Nearest Neighbours	Used USRP Radar to collect 30 samples of each activity, which each contain 64 subcarriers by Fast Fourier Transformer (FFT) produced	Machine Learning: 70% USRP radio signal dataset training algorithm (70% * 64 * 30 = 1344 samples each activity)	2	90.71%
	Neural Network mode				93.40%
	Ensemble Classier				93.83%

required elements once. These constitute one-shot learning elements, and are finally achieved HAR by the associative memory function of the Hopfield Neuron Network with the similarity of the distance calculation. Of course, this method also has certain limitations. For instance, the number of learning activities is limited to the neurons. The more activity classes require more neurons to remember patterns to avoid spurious patterns in Hopfield neural networks. The feature extraction process (achieve the memory pattern to the Hopfield neural network) also has a significant disadvantage. The SVD algorithm works with massive matrix data and then reduce it to useful feature maps of each activity. It should design an elaborate construction matrix to feature map for adapting Hopfield neural network memory learning, which then achieves suitable binary patterns for different activities without interference. Nevertheless, as the number of learned activity classes increases, the task becomes more difficult. This is because the associative memory function of the Hopfield Neuron Network has the limitation of the storage capacity proportion. Generally, it can only store 0.14N data (N is the number of neurons) [142]. When a large number of feature templates need to be memorized, more neurons need to be added into network to meet the demand.

3.5 Summary

In this chapter, first use the Quaternion and Euler Angle to fuse multiple sensors data, followed by extracting the features against each human activity with the SVD algorithm. Finally, following the designed activity feature matrix to train and test on the Hopfield neural network achieves human activity recognition. The proposed approach shows 96.3% classification accuracy after one training sample of each activity while improving performances and robustness compared with traditional machine learning approaches. The research suggests that Hopfield neural network can avoid large training dataset requirements when preprocessing is used for designing the activity feature matrix. The neural network weight specified by the Neural Dynamics operation follows the Hebbian learning method to train the Hopfield algorithm. It is one-shot learning that only requests one sample of information to calculate the feature template, which is simple and fast. The algorithm starts to associate memory and feedback with the output matrix information, making it much less dependent on training samples of the dataset. Following the results, it is verified that the proposed framework is suitable for general datasets to reduce the training samples request, which is limited datasets can also build a high-accuracy recognition model.

Chapter 4

An Intelligent Implementation of Multi-Sensing Data Fusion with Neuromorphic Computing for Human Activity Recognition

The increasing demand for considering multi-sensor data fusion technology has drawn attention for precise human activity recognition over standalone technology due to its reliability and robustness. This paper presents a framework that fuses data from multiple sensing systems and applies Neuromorphic computing to sense and classify human activities. The data is collected by utilizing Inertial Measurement Unit (IMU) sensors, software-defined radios, and radars and feature extraction and selection are performed on the data. For each of the actions, such as sitting and standing, an activity matrix is generated, which is then fed into a discrete Hopfield neural network as a binary feature pattern for one-shot learning. Following the Hopfield network neurons' feedback output, the conformity to the standard activity feature pattern is also determined. Following the Hopfield network neurons' feedback output, the training of neurons is completed after 2 steps under the Hebbian learning law, and the conformity to the standard activity feature pattern is also determined. According to probabilistic statistics on inference predictions, the proposed method that Neuromorphic computing of the three data fused framework achieved the Box-plot for highest lower quartile output of 95.34%, while the confusion matrix classification accuracy of the two activities was 98.98%. The results have shown that Neuromorphic computing is most capable for multi-sensor data fusion-based human activity recognition. Furthermore, the proposed method can be enhanced by incorporating additional hardware signal processing in the system to enable the flexible integration of human activity data.

4.1 Introduction

In recent years, the application of multi-sensor data fusion technology has become popular for military, industry, and emerging technology development applications [222]. Multi-sensor Information Fusion (MSIF) is an information processing technique in which the data from multi-sensor or multi-source hardware are fused and analyzed to complete the required decision-making and estimation [223].

MSIF technology is widely used in industrial control [224], robotics [225], object recognition [226], traffic control [227], inertial navigation [228], agriculture [229], remote sensing [230], medical diagnosis [231] and other fields [232]. Research studies have proven that compared with single-sensor systems, the use of MSIF technology results in accurate detection and tracking of subjects' activities [233]. Moreover, it can enhance the validity, reliability, and robustness of the entire system, improve data credibility to increase accuracy, expand the time and space coverage, and reinforce the system's real-time performance and information utilization [234].

Muhammad *et al* [235] proposed a data fusion-based system for ensemble computing with the Random Forest Classifier algorithm to predict results from multiple sensors. The results of the study was promising as it recorded an average accuracy of more than 90% after performing data fusion. The authors in [205] used the Sequential Feature Selection (SFS) method to fuse the Inertial Measurement Unit (IMU) and Radar information to form time series data, which can be used as features to train the Support Vector Machine (SVM) and Artificial Neural Network (ANN) algorithm for classification computing, which increases the accuracy by approximately 6% compared to using a single type of data.

In view of the uneven data quality of different hardware platforms [236], Huang *et al* [9] used multi-scale features by three sparsity-invariant operations. It depends on a hierarchical multi-scale encoder-decoder Neural Network, which is used to process sparse input and feature maps for multi-hardware data. The features of multiple sensors can be fused further to improve the performance of deep learning algorithms. However, a multi-sensing system normally requests hardware platforms to work synchronously to ensure the collected data time axis is unified in the coordinate system.

A current research focus revolves around the development of high accuracy Human Activity Recognition (HAR) systems using the limited datasets available. Traditional machine learning (especially deep learning models) has achieved practicable results in the HAR field [237], but it has also led to a large amount of training data collection overhead [7]. On the upside, deep neural networks are friendly to high-dimensional data learning and it completes the end-to-end calculation without the more cumbersome process of feature engineering. On the downside, it causes problems such as huge demand for training samples, complex model structure, and time-consuming training [238]. Moreover, it loses the cognition of features, and there are challenges to knowing the importance of data features [239]. On the other hand, neuromorphic computing

has required fewer training samples to achieve high accuracy recognition results [240]. It is based on the combination of feature engineering for the abstract expression on the object and the associative memory function of neuromorphic computing, which achieves one-shot learning for HAR.

This chapter first explores a new multi-hardware data fusion method that makes use of IMU sensors and Universal Software Radio Peripheral (USRP) for human motion recognition. Our approach is based on using building a constructing feature matrix to fuse different hardware information as unified data. Different hardware signals are difficult to match due to the signal shapes between the object. The difference in location and time axis is the main disadvantage of this data fusion. In order to overcome this limitation, this work constructed matrices from vectors based on principal component analysis to combine IMU and USRP signals, then helps multi-hardware data fusion, and finally achieve better classification and recognition effect than traditional data fusion results.

And then, the research has been extended to a novel multi-sensing HAR system, which is a Neuromorphic Computing-based data fusion method. It is based on the above data fusion idea with IMU sensors and USRP, and adds Radar signals are complete the HAR task. The method is to construct a feature matrix to fuse different hardware information as a unified data input to a Hopfield Neural Network. Constructed activity feature matrices depend on attention mechanisms to combine IMU, Radar and USRP signals for feature extraction and selection. The multi-hardware data are then fused for better classification and recognition accuracies using the Hopfield Neural Network as compared to traditional data fusion results.

The chapter is organised as follows: Section 4.2 outlines how the human motion data from the IMU, Radar, and USRP are collected and modelled. Section 4.3 is a case study to show the traditional machine learning workflow for the IMU sensor and USRP data fusion method. Follows Section 4.4 is the ANN algorithm's HAR accuracy for the fused signals. In Section 4.5, there are details of the proposed feature matrix for the data fusion method, and the algorithm calculation workflow. Section 4.6, presents a quantitative evaluation of the application of neuromorphic computing on the fused dataset in the context of existing studies in the literature. Finally, Section 4.7 summarizes the multi-sensing data fusion implementation of the Hopfield neuromorphic computing to HAR and outlines the potential future direction.

4.2 Materials and Methods

At present, there are many types of sensing hardware that can capture human movement information, but the acquisition of signals by a single device is relatively limited. In general, IMU sensors are low-cost, easy to use, and less restricted by usage scenarios, and have been integrated into many wearable devices. However, its serviceable range and accuracy are inferior to those of USRP and Radar, and its performance is constrained by other components such as

batteries and microprocessors. USRP can achieve higher precision object detection through the Doppler frequency shift principle. Depending on high power support, hardware performance can be better released. However, it is generally used in fixed scenes that cannot move quickly, which means that the capture of object signals is easily affected by some factors such as occlusion and limited angle. Comparatively, radar mainly transmits electromagnetic waves and receives echoes to obtain the distance, speed and angle of objects. It has good penetration and a strong resolution ratio. However, it is bulky and complicated to install. Therefore, following the advantages of these different types of devices, we can integrate them together to form a multi-sensing human activity perception system, which complements each other and realizes a more stable and reliable human activity recognition task. After the data fusion method is adopted, the Sensor, USRP and Radar will provide different perception information, which can overcome the limitations and discrepancy of a single device in terms of geometric, spectral and spatial resolution. Finally, it improves the data quality, and thus facilitates the positioning, recognition and interpretation of human movement information.

4.2.1 Experimental Setup and Data Collection

The data collection of human body movements was performed using three sensing hardware platforms, as listed below:

- Shimmer 3 IMU sensor [241].
- Walabot Radar DIY model [242].
- Universal Software-defined Radio Peripheral (USRP) [243] X300 unit.

First, the IMU sensor was worn on the wrist where the three axes of the coordinates system of each sensor (gyroscope, accelerometer and magnetometer) have the spatial coordinate information of X, Y and Z respectively. Then, the Radar and the USRP were positioned at a distance of 2 meters from the fixed human activity position (Shown on Figure 4.1).

The IMU [244] constitutes a gyroscope, an accelerometer and a magnetometer, used in measuring the attitude angle of an object. The gyroscope detects the angular velocity signals relative to the three degrees of freedom (X, Y and Z) in the coordinate navigation system, and the accelerometer monitors the acceleration signals of the independent three axes of the object carrier coordinate system in X, Y and Z directions. The magnetometer can obtain the surrounding magnetic field information. It can calculate the angle between the module with the north direction through the geomagnetic vector and help correct the angular velocity parameters of the gyroscope. The real-time output that includes the three-dimensional angular velocity signal, acceleration signal, and magnetic field information is used to calculate the object's posture. To capture this information, the voltage signals of the X, Y, and Z axes in the IMU sensor are digitised at sampling frequencies of $20Hz$ for the magnetic field and $400Hz$ for the accelerometer

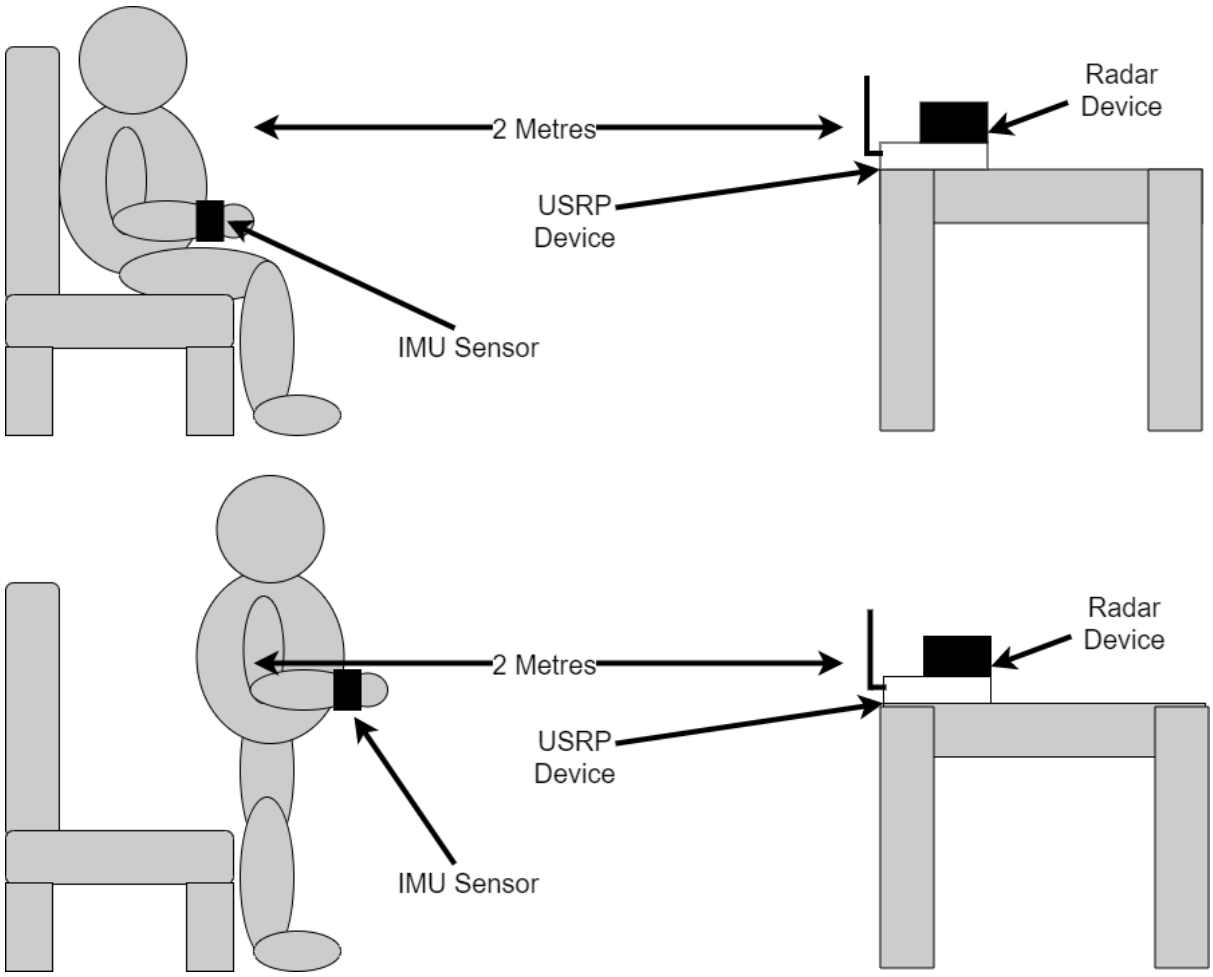


Figure 4.1: Device Setting-up and Environment for Human Activity Data Collection.

and gyroscope. The working current of the sensor is 500μA with a power supply voltage of 3.3V, resulting in a total power consumption of 1.65mW.

The Radar device used in this chapter is an off the shelf "Walabot DIY" device. The device is designed to use radar technology to detect metal and wooden studs as well as electrical wires inside of a wall to assist users with DIY tasks around the home. However, it can also be used to detect human movements [193, 245]. The Walabot radar is a Multiple Input Multiple Output (MIMO) device and does not allow for its preset parameter to be tuned. Hence, the data for this experiment was collected using the predefined settings of the product.

The USRP device is a Softare Defined Radio (SDR) used to enable Radio Frequency (RF) communication between two antennas. Two Omnidirectional antennas are connected to a single USRP device, that is, one as a transmitter and one as a receiver. The data collection window was set to 5 seconds during which the activity took place. During the 5 second communication window, the Channel State Information (CSI) are captured, reflecting the activity performed. This process is repeated multiple times to capture several samples for each activity, where the amplitude of the RF signals is extracted from the CSI. The USRP was configured to operate at 2.4 Ghz frequency similar to Wi-FI, with a 20 MHz bandwidth.

In this chapter, the USRP is set up to communicate using Orthogonal Frequency Division Multiplexing (OFDM) [246]. Channel estimation is an important feature of OFDM as it monitors the state of the channel for the purpose of improving performance. Channel estimation does this by using a specified set of symbols known as pilot symbols. These symbols are used in the transmission of the data and once the receiver antenna receives the data the received pilot symbols are compared to the expected pilot symbols and this provides details of the state of the channel.

4.2.2 IMU Sensor, USRP and Radar Modeling

• IMU Sensor Modeling: The IMU [244] constitutes a gyroscope, an accelerometer and a magnetometer, used in measuring the attitude angle of an object. The gyroscope detects the angular velocity signals relative to the three degrees of freedom (X, Y and Z) in the coordinate navigation system, and the accelerometer monitors the acceleration signals of the independent three axes of the object carrier coordinate system in X, Y and Z directions. The magnetometer can obtain the surrounding magnetic field information. It can calculate the angle between the module with the north direction through the geomagnetic vector and help correct the angular velocity parameters of the gyroscope. The real-time output that includes the three-dimensional angular velocity signal, acceleration signal, and magnetic field information is used to calculate the object's posture. To capture this information, the voltage signals of the X, Y, and Z axes in the IMU sensor are digitised at sampling frequencies of $20Hz$ for the magnetic field and $400Hz$ for the accelerometer and gyroscope. The working current of the sensor is 500μA with a power supply voltage of 3.3V, resulting in a total power consumption of 1.65mW.
• Radar Modeling: The Radar device used in this paper is an off the shelf "Walabot DIY" device. The device is designed to use radar technology to detect metal and wooden studs as well as electrical wires inside of a wall to assist users with DIY tasks around the home. However, it can also be used to detect human movements [193, 245]. The Walabot radar is a multiple-input and multiple-output (MIMO) device and does not allow for its preset parameter to be tuned. Hence, the data for this experiment was collected using the predefined settings of the product.
• USRP Modeling: The USRP device is a softare Defined Radio (SDR) used to enable Radio Frequency (RF) communication between two antennas. Two Omnidirectional antennas are connected to a single USRP device, that is, one as a transmitter and one as a receiver. The data collection window was set to 5 seconds during which the activity took place. During the 5 second communication window, the channel state information (CSI) are captured, reflecting the activity performed. This process is repeated multiple times to capture several samples for each activity, where the amplitude of the RF signals is extracted from the CSI. The USRP was configured to operate at 2.4 Ghz frequency similar to Wi-FI, with a 20 MHz bandwidth.

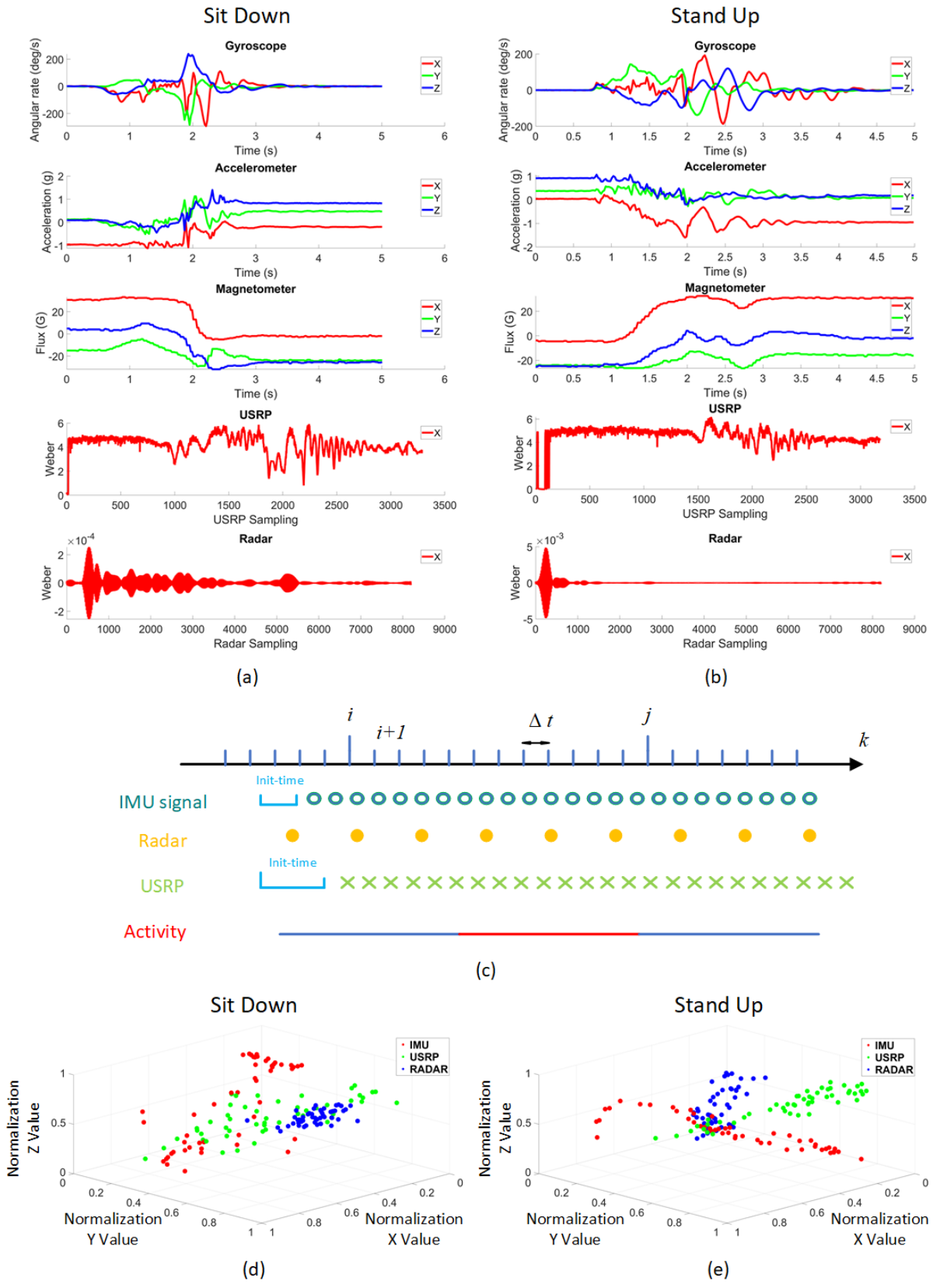


Figure 4.2: The multi-sensing raw activity signal data.

Figure 4.2 shows the raw data as captured by the IMU sensor, Radar and USRP devices

where Figure 4.2.a and Figure 4.2.b represent those of the sitting and standing activities, respectively. It is worth mentioning that the data collected from all three devices was not synchronised due to the difficulty of controlling the start and end of the data collection window and the sampling of each sensor was independent and different from each other. This resulted in an inconsistent time stamp of the collected actions, as shown in Figure 4.2c. Due to different devices requiring different initial times, the collected data of time series are non-uniform. It means data length and interval are different that are difficult directly to process.

The work summarized the error formula of the action state variable inheres:

$$\delta k = \sum_{i}^{j} \left(\int s(t)' \triangle t \cdot dt \right) dt \tag{4.1}$$

Here δk is the error value between time i and time j, and time i should be the first activity time timestamp, s is the state quantity, t is the time difference, and dt represents the microvariable with t as the variable. Following the normalization process of the raw data from each sensing unit, the measured values are then converted to unified coordinate system, which eliminates the time stamp of the information. This is shown for the sitting and standing activities in Figures 4.2d and Figure 4.2e, respectively.

4.3 Case Study: Data Fusion for IMU and USRP Signals

Data fusion utilises comprehensive and complete information about the object and environment obtained by multiple hardware devices, which is mainly reflected in the data fusion algorithm. Therefore, the signal processing's core point on a multi-hardware system is to construct a suitable data fusion algorithm. For multi-type sensor hardware, which acquired information is diverse and complex. Moreover, the basic requirements for information data fusion methods are robustness and parallel processing capabilities. There are also requests for the speed and accuracy of the method, and the previous preprocessing calculations and subsequent recognition algorithms interface compatibility that to coordinate with different technologies and methods; reduce information sample requirements, etc. In general, data fusion methods are based on nonlinear mathematical computing. It can achieve fault tolerance, adaptability, associative memory, and parallel processing capabilities.

There is two separate single hardware to acquire activity data, and how to fuse the two hardware's signal information becomes the key point. The hardware's different sampling rate makes the time axis difficult to achieve with the unified coordinates system. At this point, the raw data need to reduce the time dimension of the activity feature. The PCA [247] algorithm used for data dimension reduction can be achieved a time-independent activity features. Furthermore, after analysing the two types of signal, it can find that the sensor data has more dimensions than

USRP data. Therefore, this research designed a big sub-matrix to represent the sensor signal of human activity after data dimension reduction and a small sub-matrix to represent the USRP signal after data dimension reduction. The big sub-matrix can more accurately represent the sensor's data feature that is can keep more origin data information. Comparatively, USRP data is simple and can be represented as a small sub-matrix, it is enough to save all of the features on the small matrix.

Meanwhile, this design also facilitates subsequent machine learning, which requests normalise the fused data to obtain a standard feature map pattern [248] of each activity, and then loaded the feature map template into the Neural network algorithm for training. It gets the combination matrix of the feature map pattern. Its activity feature matrix of the sensor and USRP is combined by a direct row arrangement method of previous sub-matrices. Finally, the neural network obtains the classification function of the two activities based on the training. The neural network as a classifier to output recognition result. Figure 4.3 shows the entire calculation workflow from feature extraction to recognition.

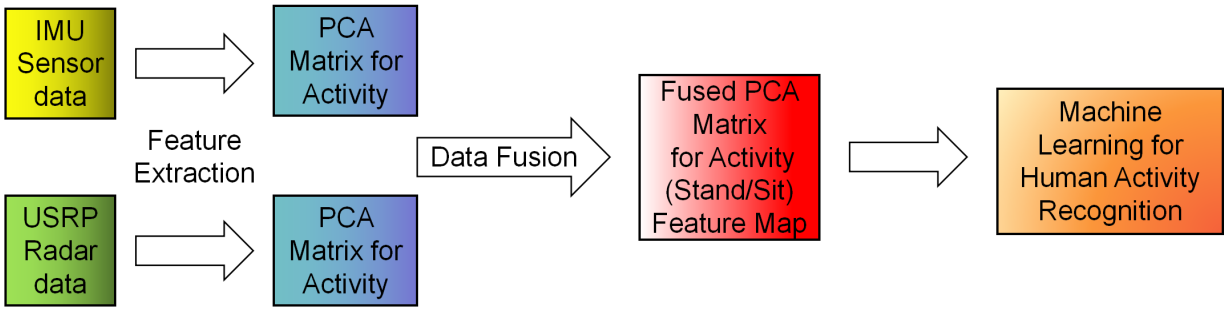


Figure 4.3: The entire calculation workflow of feature extraction to human activity recognition.

4.3.1 Principal Component Analysis for Feature Extraction

The PCA algorithm for data dimensionality reduction calculates the covariance matrix of one dimension with sample information, and then solves the feature value and corresponding feature vector of the covariance matrix. It arranges these feature vectors according to their corresponding feature values from large to small as a new projection matrix. In this case, the projection matrix as the feature vector pattern after sample data transformation. It maps n-dimension features to k-dimension space, which is a brand new orthogonal feature as the principal component. It is a re-constructed k-dimension feature based on the original n-dimension features. Following the first K-dimension vectors are the essential features of the high-dimension data retained to remove noise and unimportant interference factors to improve data information quality.

4.4 Experimental Evaluation for Two Sensing Accuracy

After PCA feature extraction of the hardware signal, it is evaluated the training and testing performance following by the Artificial Neural Network (ANN). Figure 4.4 illustrates the recognition accuracy of sensors data applied to the machine learning algorithm. The evaluation for the performance of a single sensor and fused data modality in activity recognition. It is the two layers ANN algorithm classification results of the confusion matrix for Sit down and Stand Up activities. Following feature extraction through the PCA algorithm, the ANN algorithm presents the results in terms of classification accuracy for different hardware (IMU Sensor and USRP) of direct processing and data fusion when fused the different features.

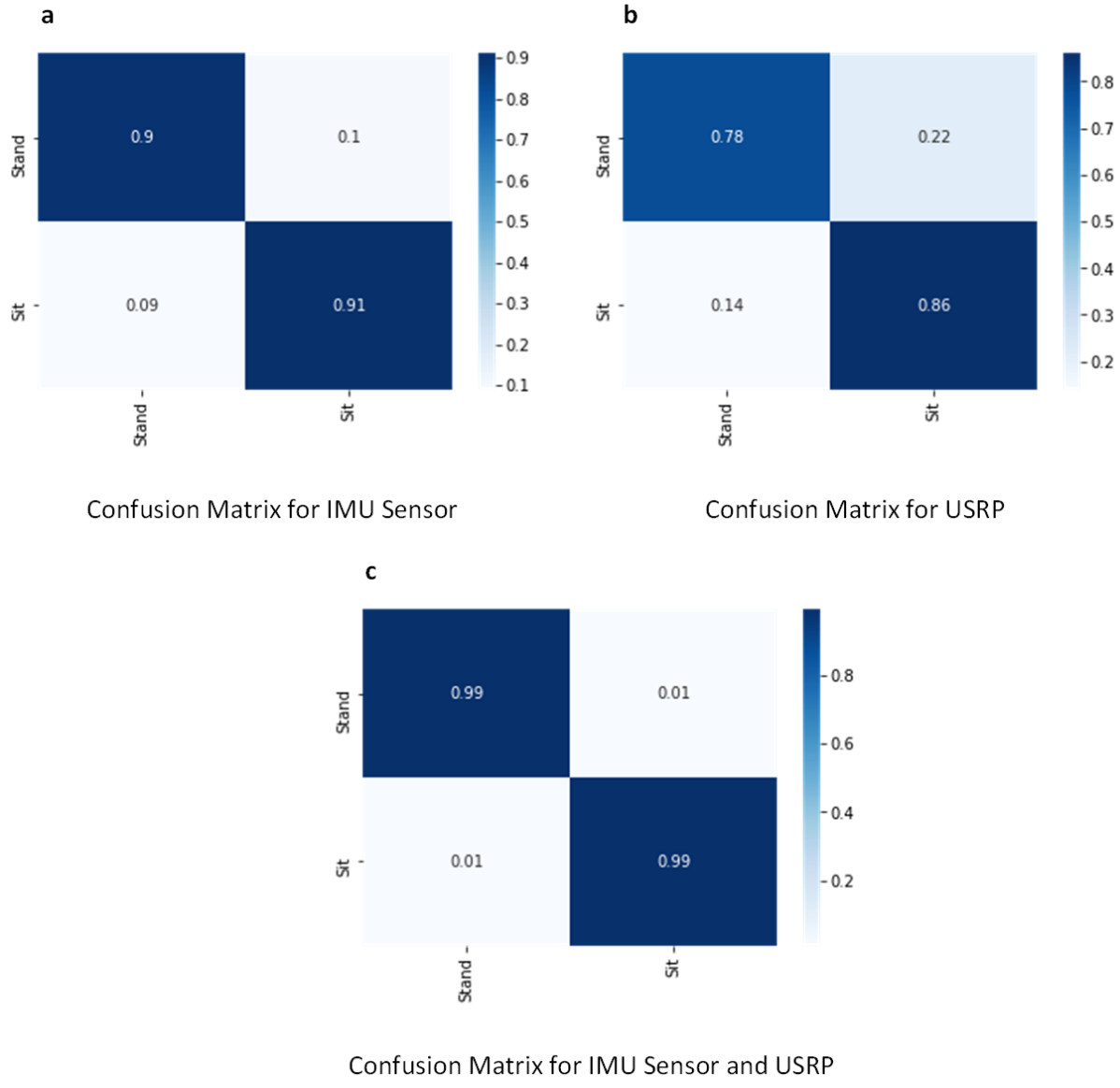


Figure 4.4: The ANN algorithm classification confusion matrix of IMU Sensor, USRP and IMU Sensor fused USRP data.

Table 4.1: Comparison table with other data fusion methods.

Project	Hardware	Algorithm	Data Fusion	Activity Classes	Accuracy
Our	IMU sensor (magnetometer + accelerometer + gyroscope) + USRP	Artificial Neural Network	Using PCA algorithm for feature fusion	2	99.2%
Chen *et al* [249]	IMU sensor (accelerometer + gyroscope)	K Nearest Neighbor	two-stage genetic algorithm-based feature selection algorithm with a fixed activation number	9	99.1%
		Random Forest Classifier			99.1%
		Support Vector Machine			98.9%
Chung *et al* [250]	IMU sensor (magnetometer + accelerometer + gyroscope)	LSTM network	Using various combinations of sensors, and two voting ensemble techniques adopting all sensor modalities.	9	94.47%
Calvo *et al* [251]	Kinect + IMU + EMG	Hidden Markov Model Classification	From each sensor, it is keep track of a succession of primitive movements over a time window, and combine them to uniquely describe the overall activity performed by the human.	5	98.81%
Zou *et al* [252]	WiFi-enabled IoT devices and camera	C3D model and CNN model ensemble DNN model	Performs multimodal fusion at the decision level to combine the strength of WiFi and vision by constructing an ensembled DNN model.	3	97.5%

By comparing the single hardware data's classification performance with the data fusion method on the machine learning of neural network algorithm, this work designed the fusion method to increase the activity classification accuracy from single signal data of 90.5% from IMU sensor and 81.8% from USRP signal to 99.2% after the IMU and USRP data fusion. This evaluation proves that our solution can pass the method of constructing the matrix helps the data fusion between different hardware, and the fused data can obtain higher accuracy. When the sensors are used individually, the IMU Sensor is more suitable for measures human activity. For multi-hardware of both IMU sensor and USRP, it improves single type signal quality with more angle and dimensionality on the feature extraction.

Table 4.1 shows a comparison against traditional machine learning algorithms' accuracy and proves that better results are achieved through the proposed data fusion method. Such as Chen *et al* worked on the IMU Sensor with accelerometer and gyroscope, and they used traditional machine learning algorithms (K Nearest Neighbor (KNN), Random Forest Classifier (RFC) and Support Vector Machine (SVM)) to classify human activities. Furthermore, Chung *et al* improve the data fusion method to suitable for 9 axes IMU Sensor (magnetometer, accelerometer and gyroscope) and achieve results from the LSTM network. Based on the Kinect, IMU and EMG of the multi-hardware platform, Calvo *et al* implements the Hidden Markov Model Classifier to recognise human activity signal, and Zou *et al* design a Deep Neural Network (DNN) framework to ensembled the C3D and Convolutional Neural Network(CNN) model to processing fused data of WiFi-enabled IoT devices and camera. However, through comparing accuracy, our implementation is more accurate than their classification. We believed that the recognition findings are preferable, demonstrating that the PCA model to fuse multi-hardware signal features effectively recognises human behavior. Furthermore, our proposed workflow has greater robustness and can adapt to more different types of matrix data.

4.5 The Proposed Structure Matrix to Data Fusion for IMU, Radar and USRP Signals

Figure 4.5 shows the framework and the data flow from the multi-sensing stage to the neuromorphic computing stage, for HAR. Firstly, human motion information is respectively collected on different hardware platforms, and features are extracted from the collected raw data.

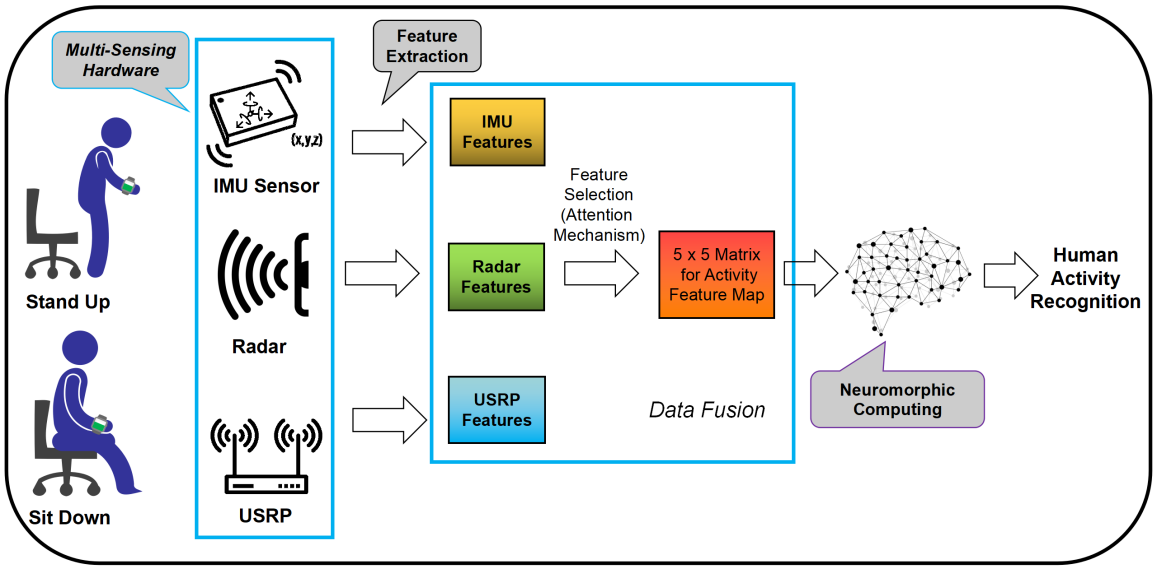


Figure 4.5: The multi-sensing entire calculation workflow to human activity recognition.

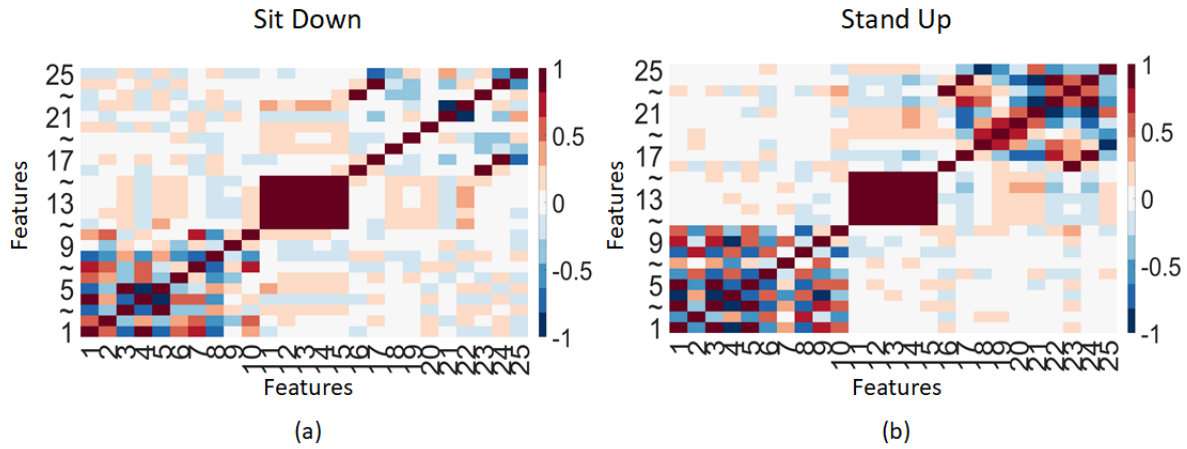
4.5.1 Feature Extraction and Feature Selection

Feature selection and feature extraction are two important sub-contents of Feature Engineering. Among them, feature extraction can find the attributes that best represent the uniqueness of the data [253]. Feature extraction obtains a new feature space by transforming or mapping the original raw data, such as mapping from three-dimensional space to two-dimensional space. The purpose of feature extraction is to use fewer features to represent most of the information in the original data space. Thus, it can improve computing efficiency and reduce dimension disasters.

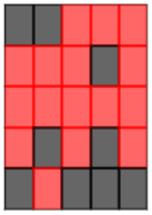
However, after observing the raw data of multi-sensing devices, it can be found that the data dimensionality reduction calculation of previous used PCA and SVD feature extraction algorithms is no longer suitable for feature expression on the multi-sensing signal processing. Because with the increase of sensing devices, directly reducing the dimension of the raw data to obtain the feature matrix will cause uneven feature distribution. In extreme cases, the obtained activity feature matrix will lose part of the data of the sensing devices, and only rely on one or a few devices' information to express the feature matrix without all of the multi-sensing devices' signals. In this case, not only feature extraction algorithm is used, but also feature selection methods are needed. Feature selection is to select the appropriate feature from the candidate features [254]. It can reduce the dimension of the data, and improve and optimize the ML model's performance.

Figure 4.6 shows the process from Raw data feature extraction to the Attention Mechanism [255] of TopK [255] feature selection [256], and binarization for human activity features map. Figure 4.6a is the raw multi-sensing data calculated by a tree-based prediction model that can be

used to list features and obtain the heat map after TopK order [255] [256] for a feature selection method. That is, all elements of the feature array are sorted in ascending order, and then the first k numbers are taken. Figure 4.6b is the 5*5 feature matrix after extracting the best 25 features of TopK computing. Finally, Figures 4.6c and d are the human activity feature pattern after binarization (-1/1) by features values (Following positive and negative values to binarization).

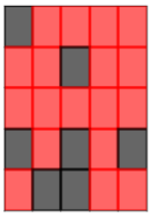


0.0000	0.4985	−4.3935	−14.4429	−31.2570
987.7537	48.8553	0.0440	0.4163	−87.6179
5.0324	5.4020	5.1412	5.0044	5.2213
0.0000	0.0001	−0.0001	0.0000	−0.0001
0.0001	−0.0001	0.0000	0.0001	0.0001



(c)

0.0201	0.5464	−4.3924	−16.6435	−30.4196
987.7571	55.7694	0.0858	0.6495	−91.3581
4.6325	5.2254	5.0369	4.9262	4.9702
−0.0001	0.0001	−0.0001	0.0000	−0.0001
0.0001	−0.0001	−0.0001	0.0001	0.0001



(d)

Figure 4.6: Feature Extraction for HeatMap.

The Attention Mechanism [255] of neural networks is a resource allocation scheme. In neural network learning, the stronger expression ability of the model requests more parameters on the neurons. Meanwhile, more information can store on neurons, but this will bring information overload. Therefore, depending on the attention mechanism, the neuron network pays more attention to the high critical information on the current task. Meanwhile, filtering out irrelevant information and reducing attention to other information. As a result, information overload can be solved, and the accuracy and efficiency of task operation can be improved, by allocating computing resources to high important tasks.

Inhere, the Attention Mechanism selectively ignoring unimportant information by the following activity features‘ importance. Then, it focuses on these features to express the corresponding activity. The focusing process is reflected in the calculation of feature weight coefficients. The weight shows the essential features of data. Through the heat map of feature correlation, there are TopK [257] (K=25) features selected to represent the original information of the activity.

The formula of the Attention mechanism's distribution probability is represented below: [258]

$$Attention(Query, Source) = \sum_{i=1}^{L_x} Similarity(Query, Key_i) \bullet Value_i \tag{4.2}$$

Source is the stored data, and Query is for fetching the corresponding value in the memory of stored data as the attention value. The Lx denotes the length of the Source, it is a series of <Key, Value> data pairs. In this case, the weight coefficient of the corresponding Value of each Key can be obtained by element Query in the Target. First, it calculates the correlation or similarity between Query and each Key and then, the Value is weighted and summed to get the final Attention value. Essentially, the Attention mechanism is a weighted sum for the values of elements in the Source, while Query and Key are used to calculate the weight coefficient of corresponding values.

4.5.2 Hopfield Neural Network and Euclidean Distance

Designing neuromorphic computing for end-to-end signal processing. Firstly, the raw data is feature extracted through data preprocessing, and the feature map of the corresponding activity is obtained as explained earlier in section 4.5.1. The binary feature pattern is then fed to the Hopfield neural network [203] for training. Finally, the output signal is compared by the Hopfield neural network and the corresponding activity feature map. It can recognise the input signal that has been trained or not to achieve the inference result of the activity. Depending on Hopfield neural network that is a fully connected structure of recurrent feedback neural network to achieve the associative memory of neuromorphic computing.

The Discrete Hopfield Neural Network (DHNN) [259] is based on binary feedback to realise associative memory work. Following the step function of activation calculation to each neuron, its input and output of the neuron are binary values of -1 and 1. The Hopfield neural network training phase is illustrated in Figure 4.7 for the sitting and standing activities. Figure 4.7a depicts one Hopfield neural network state-space neuron that processes the training for both activities. The weight of the DHNN is calculated using a binary feature matrix (5*5 feature pattern achieved by above feature extraction and selection of human activity), and trained by Hebbian learning law [203]. It indicates that the neuron has reached a stable state after the 2

steps of training. Figure 4.7b displays the entire DHNN weight output after training the Hebbian learning algorithmusing the two preprocessed binary patterns of activities. It shows the weight value distribution for each neuron's connection, which is easy to analyse the weight value range.

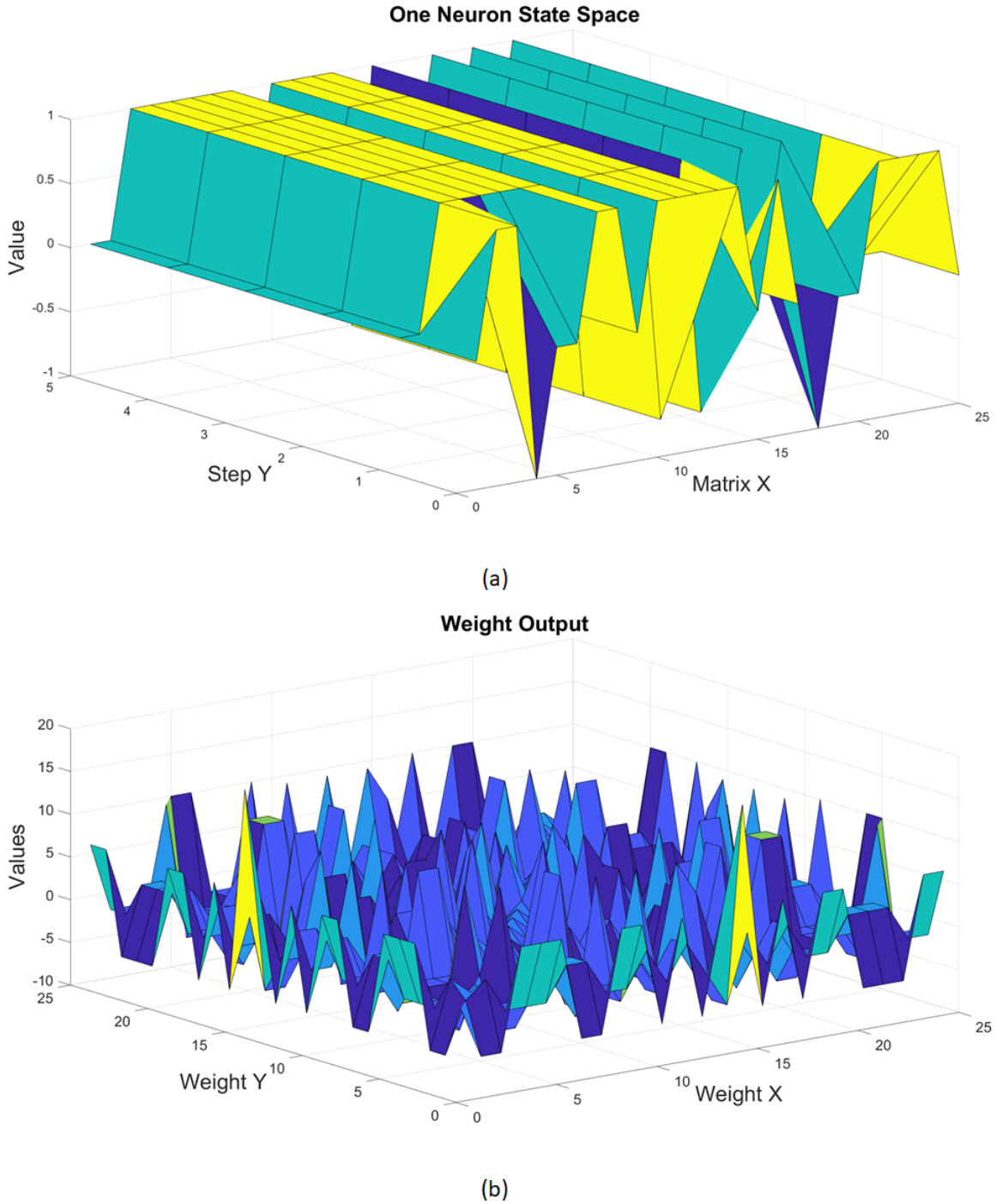


Figure 4.7: Hopfield Neural Network Neuron State and Weight Output.

All device signals will go through the feature extraction to output different feature patterns. The input data is transferred to the neurons, and it is like a filter that only passes the data for trained two activity feature patterns on the activation state. Then, the Hopfield neural network output links to the Euclidean distance algorithm. It is based on the similarity to estimate the recognition result of the HAR, which compares the output of the neural network and trained feature patterns. The Euclidean distance to calculate the similarity is the distance between two points and it is always a non-negative number [260]. Thus, the similarity value range is between [-1,1], and its reciprocal will control the result between [0,1]. At this point, distance is negatively correlated with similarity. Two trained activities will get a high probability similarity output, while other data signals will output a low probability similarity because the neuron is not activated. Finally, the classifier for neuromorphic computing is completed to realise effective HAR.

Algorithm 4.1: Multi-Sensing Data for Human Activity Recognition.

Load Multi-Sensing Hardware data:
[G_x, G_y, G_z] = Gyroscope [:column_1, column_2, column_3]
[A_x, A_y, A_z] = Accelerometor [:column_1, column_2, column_3]
[M_x, M_y, M_z] = Magnetometor [:column_1, column_2, column_3]
[U_x] = USRP data Matrix [:1]
[R_x] = Radar data Matrix [:1]
Require: :
Feature Extraction: Tree-based prediction model
$F_{IMU}(G,A,M) = (f(G_x)) + (f(G_y)) + (f(G_z)) + (f(A_x)) + (f(A_y)) + (f(A_z) + (f(M_x)) + (f(M_y)) + (f(M_z))$;
$F_{USRP}(U) = f(U_x)$;
$F_{Radar}(R) = f(R_x)$;
Feature Selection:Attention Mechanism
F‘ = Sort $F_1(G)$, $F_1(A)$, $F_1(M)$, $F_1(U)$, $F_1(R)$
M_f = TopK(F‘) : K = 5*5
Binarization Matrix: Depending on a threshold value
Threshold value = t
for $n_i = 0{:}24$ **do** : $out[n_i][n_j] = (M_f > t[n_i])$? 0 : 1;
return **Matrix B_M(5:5)**;
HNN = Hopfield(B_M)
Out(Confidence) = Euclidean_Similarity(HNN-B_M)
return **Recognition Result**;

Table 4.2: Algorithm Hyper-Parameter and Value Setting

	Hyper-Parameter	Value				
1.	G_x, G_y, G_z	Initial to 0, and then load to sensors data for 1-3 column				
2.	A_x, A_y, A_z	Initial to 0, and then load to sensors data for 4-6 column				
3.	M_x, M_y, M_z	Initial to 0, and then load to sensors data for 7-9 column				
4.	U_x	Initial to 0, and then load to USRP data				
5.	R_x	Initial to 0, and then load to USRP data				
6.	Threshold t for 5x5 matrix	0.084158	0.169579	13.23165	41.138027	68.868009
		0.001983	0.11178	3939.6904	0.089959	5200.0895
		0.871498	0.74435	0.703459	0.800664	0.696558
		1.36E-08	1.49E-07	7.80E-09	3.09E-09	8.75E-09
		1.66E-09	1.07E-09	3.21E-09	3.73E-10	9.38E-10

4.5.3 Proposed Algorithm Implementation Scheme

Algorithm 4.1 verifies the feasibility of the whole framework theoretically, and shows the specific calculation process of each step in the workflow. In order to avoid interference between the different types of hardware signals in the calculation, feature extraction will be performed separately, first and then, work on the feature level fusion. This processing helps the different types of signals keep the original information. Depending on the attention mechanism [258] of TopK computing, the most important sub-features can be extracted from the fused feature-set. In order to make the Hopfield neural network get better processing results, the activity feature matrix is converted into the Binarized feature pattern by calculating the threshold values. Finally, following the calculation of the similarity between the Hopfield neural network's output and feature pattern. The similarity output can work as the confidence of the activity classification to be achieved in the HAR process, which means the similarity directly corresponds to confidence. All of the algorithm 4.1 hyper-parameter and value settings have been shown in the Table 4.2.

4.6 Experimental Evaluation for Multi-Sensing Neuromorphic Computing Accuracy and Discussion

As compared to the classification performance of data collected using a single hardware platform, the data fusion methodology adopted in this chapter increases the activity classification accuracy through feature-level fusion of the IMU sensor, Radar and USRP signal, which recorded an accuracy of about 98.98% (see the multi-class confusion matrix in Figure 4.8). This is further

shown in Figure 4.9 where a box and whiskers plot is use to compare the inference probability when using single devices, as well as fusion of two and three of the devices together. As can be seen, applying neuromorphic computing to fuse HAR data from three hardware devices is the most stable compared to machine learning results from a single device and data fusion of two hardware devices. Therefore, this evaluation proves that our solution can pass the method of constructing the matrix to help the data fusion between different hardware, and the fused data can obtain higher accuracy performance by the Neuromorphic computing algorithm.

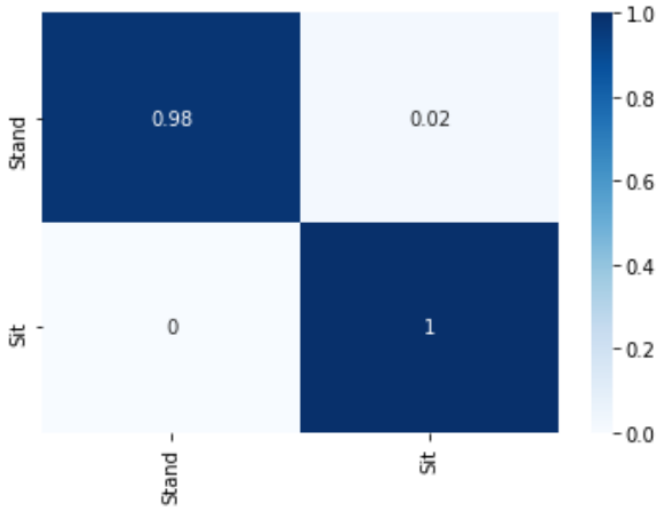


Figure 4.8: Confusion Matrix for IMU, USRP and Radar sensor fusion of HAR accuracy.

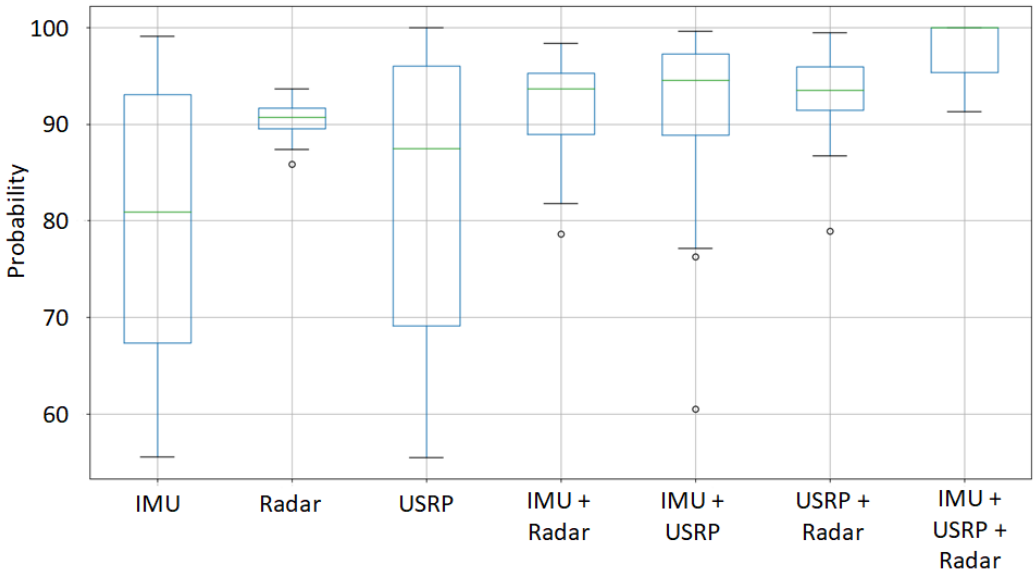


Figure 4.9: Box and whiskers plot to compare the machine learning accuracy obtained from data collected using single devices and fusion of two and three devices.

Table 4.3: Comparison table with multi-sensing data fusion methods between neuromorphic computing with traditional machine learning

Project	Hardware	Algorithm	Feature	Activity Classes	Accuracy
Our Method	IMU sensor (magnetometer + accelerometer + gyroscope) + USRP + Radar	Hopfield Neural Network(Neuromorphic Computing)	Tree-based prediction model for feature extraction and feature level fused of feature selection by Attention Mechanism of TopK	2	98.98%
Bangaru *et al* [261]	EMG and IMU sensors	Artificial Neural Network (ANN)	The EMG and IMU sensor data is normalized by z-score standardization to data fusion	15	93.29%
Chung *et al* [250]	IMU sensor (magnetometer + accelerometer + gyroscope)	Long Short Term Memory (LSTM) network	Using various combinations of sensors, and two voting ensemble techniques adopting all sensor modalities.	9	94.47%
Cao *et al* [10]	Frequency Modulated Continuous Wave (FMCW) Radar	Convolutional Neural Network (CNN)	The coordination of both labelling methods by the neighbor-aggregating-based labeling method and incorporates with clustering-based labelling method that is motivated to be implemented in the weighted combination form.	6	93.60%
William *et al* [221]	Universal Software Radio Peripheral (USRP)	K Nearest Neighbours	Used USRP Radar to collect activity, which each contain 64 subcarriers by fast Fourier transformer (FFT) produced	2	90.71%
		Neural Network mode			93.40%
		Ensemble Classier			93.83%

Table 4.3 shows a comparison against traditional machine learning algorithms' accuracy and proves that better results are achieved through the proposed data fusion method. For instance, Bangaru *et al* worked on the EMG and IMU Sensor, and they used Artificial Neural Network (ANN) to classify human activities. Furthermore, Chung *et al* improve the data fusion method to be suitable for 9 axes IMU Sensor (magnetometer, accelerometer and gyroscope) and achieve results from the LSTM network. Based on the Frequency Modulated Continuous Wave (FMCW) Radar, Cao *et al* implements the Convolutional Neural Network Classifier to processing fused data recognise human activity signal, and William *et al* design a framework to ensemble the KNN, Neural Network and Ensemble Classier model to processing of USRP human activity data. However, through comparing accuracy, our implementation is more accurate than their classification. We believed that the recognition findings are preferable, demonstrating that the Hopfield Neural Network of Neuromorphic computing to fuse multi-hardware signal features effectively recognises human behaviour. Furthermore, our proposed workflow has greater robustness and accuracy performance.

4.7 Summary

This chapter proposed an architecture for a HAR system that uses Neuromorphic Computing to integrate different hardware signal data for sensing and classifying human behaviours swiftly and efficiently. The Hopfield Neural Network of associative memory function was applied for one-shot learning to Human activities. This approach not only addresses the issues with traditional machine learning for large training sample requirements, but it also allows for greater flexibility in fitting multi-sensing hardware signals. The suggested technique has the great potential to assist the different types of measurement devices in achieving system-level data fusion without affecting the accuracy of classification and recognition. Furthermore, validation methods are employed throughout to demonstrate that the method yields a significant improvement in accuracy when a sensor, USRP, and radar data are fused. The proposed approach has shown a classification accuracy of approximately 98.98% and has demonstrated the strong potential of Neuromorphic Computing of multi-sensing data in human activity recognition.

Chapter 5

Hardware-Based Hopfield Neuromorphic Computing for Fall Detection

With the popularity of smart wearable systems, sensor signal processing poses more challenges to machine learning in embedded scenarios. For example, traditional machine-learning methods for data classification, especially in real time, are computationally intensive. The deployment of Artificial Intelligence algorithms on embedded hardware for fast data classification and accurate fall detection poses a huge challenge in achieving power-efficient embedded systems. Therefore, by exploiting the associative memory feature of Hopfield Neural Network, a hardware module has been designed to simulate the Neural Network algorithm which uses sensor data integration and data classification for recognizing the fall. By adopting the Hebbian learning method for training neural networks, weights of human activity features are obtained and implemented/embedded into the hardware design. Here, the neural network weight of fall activity is achieved through data preprocessing, and then the weight is mapped to the amplification factor setting in the hardware. The designs are checked with validation scenarios, and the experiment is completed with a Hopfield neural network in the analog module. Through simulations, the classification accuracy of the fall data reached 88.9% which compares well with some other results achieved by the software-based machine-learning algorithms, which verify the feasibility of the hardware design. The designed system performs the complex signal calculations of the hardware's feedback signal, replacing the software-based method. A straightforward circuit design is used to meet the weight setting from the Hopfield neural network, which is maximizing the reusability and flexibility of the circuit design.

5.1 Introduction

With the ever-improving living conditions, sensors-based healthcare has been widely adopted. Embedded systems are used as monitoring tools for well-being or preventive purposes. Such systems are composed of non-invasive and wearable sensors with processors [262]. Sensors

acquire a variety of physiological signals and can be used for life-saving implants, medical treatments, and long-term health monitoring of disabled or elderly persons [263]. For example, real-time, reliable and accurate monitoring results provided by the sensor system are used for fall recognition [264], blood glucose monitoring [265] and asthma tracking [266].

Presently, machine learning is being widely used for various applications in diverse fields and health care is no exception. It can be used to improve the accuracy of monitoring and detection, which is a popular method in the wearable field [267]. However, the combination of machine learning and embedded devices still has a technological divide. Primarily, it is a huge challenge to design low-power and real-time healthcare Artificial Intelligence (AI) systems. For instance, an embedded system is not capable of running an AI algorithm of neural network for low-power and real-time processing of multi-sensor signal [268], such as data fusion of magnetic, acceleration and RF/Radar sensor signal processing for activity detection. Furthermore, if real-time cloud solutions were used to collect and process the wearable device signals on the host computer, it would use more data and become inconvenient and a risk for customers [269].

The problem is that the neural network algorithm requires a large number of differential equations to be solved, to compute gradient descent, such as mean square error and regularization calculations. The instruction set and architecture of a microprocessor constrain these software-based calculations on a general-purpose processor, and it requires many consumption thread resources to complete. In this respect, hardware-based neuromorphic computing systems present a more efficient AI technology, that has been proposed by researchers and scientists at various levels. It aims to stimulate neurons using specialized hardware that uses the discrete and sparse nature of neuron pulse behavior to achieve desired results.

Neuromorphic computing hardware refers to a hardware system that supports the scale of simulated neural network models and the speed of neural computing. Its initial hardware implementation includes Field Programmable Gate Array (FPGA) [270], Neuromorphic Chip (Based on ASIC) [271] and Digital Signal Processor (DSP) [272]. The core of hardware implementation research is the construction of neural devices, which can be of electronic [273], optical [274] and biological [275] nature. In the research of neuromorphic computing, for neural network algorithms to be effectively applied in applications, the neural network implementation technology request to support the neuromorphic computing of scalable network architecture. Meanwhile, the system shall minimize the cycle time of the neuromorphic computing process to match real-time processing. However, the existing system based on the software environment of various AI algorithms is difficult to support the large-scale and running time of the neural network model, which cannot meet the application requirements. Therefore, the development of hardware-implemented neural network computers is imperative to replace the software environment and achieve hardware acceleration.

Traditional neuromorphic computing of Hopfield neural network algorithms has been applied to data classification that relies on associative memory model. Data is processed directly

by storing and memorizing standard data models. For example, Rong and Junfei [276] tried to use the network for water quality detection. They designed a memory template by selecting the water quality parameters for data preprocessing, and then passed the actual data back into the network. They used the binary values 1 and -1 to indicate the water quality monitoring and finally output the feedback matrix to match the classes of the water quality information in the memory models. Cantini et al. [277] used the memory retrieval of the Hopfield neural network to describe gene expression patterns and analyze the transcriptome data. They used discrete values to represent the signature genes in the sample and assigned corresponding neurons to them, where the network used a different gene to mark the nodes. With the network convergence, the model finally evolved to classify the genetic data. In another study, Ray and Majumder [278] used the Hopfield neural network to perform feature matching on a two-dimensional array. By comparing the features of the test scene and the object model, they used the neuron to output the probability to achieve the data classification. López et al. [279] implemented the Hopfield neural network on embedded systems such as the Arduino UNO, Tiva-C and BeagleBone development boards, and achieved fast execution times that even performed machine-learning algorithms. Furthermore, Boriskov [280] proposed Hopfield algorithm uses rate coding to expand the capabilities of neuromorphic engineering by hardware design that uses the thresholds of zero crossings of output voltages of neurons where spiking frequencies of oscillators are controlled either by supply currents or by variable resistances. However, the main disadvantages of these hardware implementations are their limited processing and storage capacity, therefore, such systems can only be used when there is scarce data [279], and request special hardware component design (such as Leaky Integrated-and-Fire (LIF) Rate Coding (RC) oscillators) [280] to achieve neuron function. It greatly limits scalability and extensibility of hardware-based neuromorphic computing implementation.

This chapter focuses on the use of neuromorphic computing algorithms on hardware circuits, so that low-power and real-time systems particularly for health care applications can be implemented. It uses hardware circuit features for algorithm calculations, to reduce power consumption due to the complexity of the computing architecture. The algorithm adopts the integrated structure of storage and computing like human brain processing. It is different from the von Neumann architecture where storage and computing are separated, and can process data more efficiently as a neuron model. The neuromorphic computing can process data in real time that does not require data interaction between multiple modules during the calculation. In the Al field, its conditions in terms of flexible architecture, modular design, and non-declining computing efficiency are rather good, and development potential is very significant [281]. Meanwhile, the neuromorphic computing of the feasibility of the design way of hardware, which to take advantage of low power consumption and real-time processing [282].

The chapter is structured as follows. Section 5.2 introduces the neuromorphic computing with data classification of the Hopfield neural network. Section 5.3 shows the details of hard-

ware architecture. In Section 5.4, fall recognition results and comparison of the application with other AI methods is presented, which discusses in detail the hardware-based neuromorphic computing and software-based machine-learning method. Section 5.5 summarizes the entire hardware implementation of the Hopfield neuromorphic computing algorithm and outlines the potential future direction.

5.2 Neuromorphic Computing Implementation

Traditional human fall recognition techniques mostly rely on the Microprogrammed Control Unit (MCU)-based embedded systems that only sense but process human activity. However, such an approach has an inherent disadvantage in terms of the speed with which any recognition can be detected. Therefore, it is difficult to support the complex calculations any machine-learning algorithm requires. Moreover, most embedded systems are limited to low power, and hence sensor data cannot be processed in real time, something which is critical in designing prediction-based systems.

In this chapter, it proposed a system that first collects data in real-time from an IMU sensor related to human activities, and then pre-processes this sensor data by posture calculation to estimate the fall feature during human movement. Finally, the data is passed to a hardware-based neural network, which replaces the software-based machine-learning method to perform the recognition calculations. The proposed hardware-based neural network has lower power consumption and real-time processing speed, which is designed by congenital advantage of lower power analog components, and no latency processing to match the sensors input. Thus, it improves application quality for human activity recognition scenarios. Here, a hardware circuit has been designed that implements a Hopfield neural network algorithm. Through our system, the human fall activity can be processed on analog circuits. In this chapter, the entire process from the beginning of the IMU sensor data collection to the final recognition result on the circuits has been built and verified in a simulation environment. The experiment completed the Hopfield neural network in the analog module by Cadence PSpice 17.4, and through the co-simulation of MATLAB Simulink 2020B. As shown in Figure 5.1, the sensor is worn on the wrist, and the real-time signal can be collected at a sampling rate of 50 Hz, which is then passed to the front-end circuit for data preprocessing. The next step involves processing by the neural network algorithm, hardware done on the system hardware. In Section 5.3, the work describes in detail the working principle of the algorithm, specifically, how neuromorphic calculations are generated based on Hopfield neural network. It further illustrates how to design and build the analog circuit which can further enable the production of useful feedback signals.

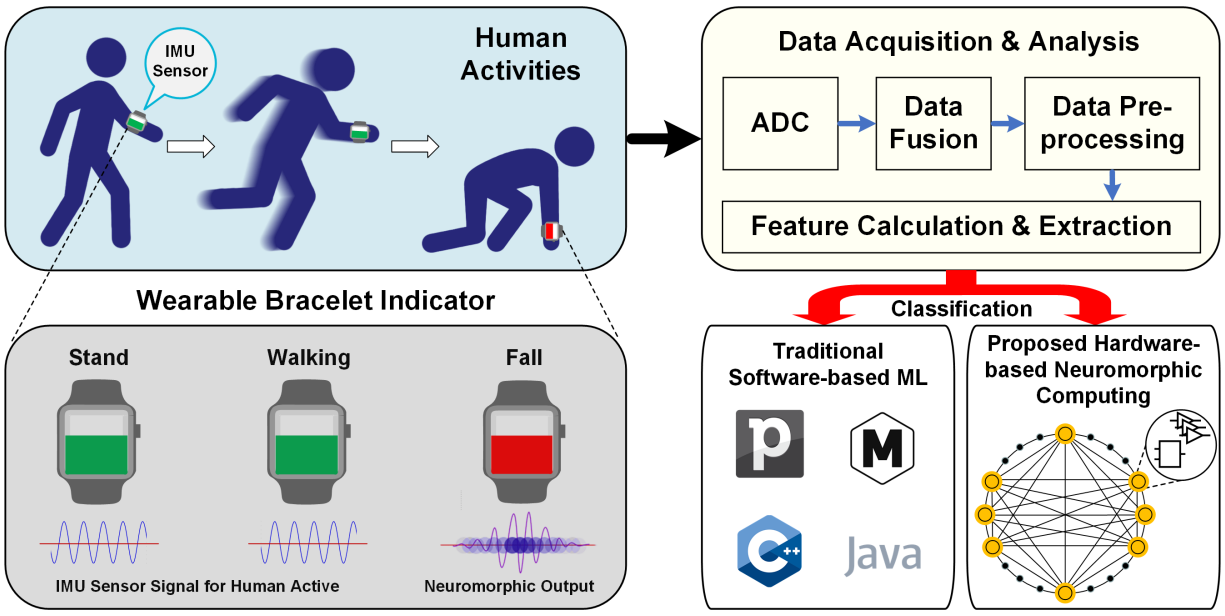


Figure 5.1: System Workflow from human activities to fall detection.

5.2.1 Hopfield Neural Network Algorithm and Training

The Hopfield neural network is a fully connected network proposed by J. Hopfield in 1982 [283] that can be used as an associative memory. It is a recurrent neural network, which has feedback connections from input to output. All neurons are the same structure and connected. Each neuron receives feedback information from other neurons through connection weights, and the signal can transfer in both positive and negative directions. Such a design allows the output of neurons to be controlled by all other neurons, so that each neuron can interact with each other.

Hopfield network can be divided into discrete Hopfield neural network (DHNN) and continuous Hopfield neural network (CHNN) [284]. Continuous Hopfield neural network is mainly used for optimization calculation, and discrete Hopfield neural network is primarily used for associative memory. Among them, the neuron variable function of discrete Hopfield network is symbolic. The node states of the network take the binarized +1 and −1. Hopfield neural network is derived from a nonlinear dynamical system, and DHNN can be described by a set of nonlinear difference equations [285], while differential equations usually describe to the CHNN [286]. Compared to other machine-learning algorithms, the Hopfield neural network is more straightforward and less dependent on the data.

Discrete Hopfield neural network (DHNN) is an essential type of Hopfield neural network, where both the input and output are binarized. The synaptic weight between neuron i and neuron j is W_{ij} [287], so for a Hopfield neural network with N neurons, the weight matrix size is $N * N$, and its unique associative memory of DHNN is through a series of iterative processes until the system is stable. Meanwhile, neurons in the network are connected symmetrically, i.e., $W_{ij} = W_{ji}$. If Wii is output to 0, then the neuron has no connection with itself. It is called the Hopfield network without self-feedback. If W_{ii} output is not 0 it means a Hopfield neural network with

self-feedback structure. However, considering the stability of the network, it should avoid using networks with self-feedback. Each neuron of DHNN only takes discrete binary values of 0 or 1. W_{ij} determines the weight between neuron i and neuron j. Neurons have current state u_i and output v_i [288]. The u_i can be continuous value in the processing, but v_i is binary value in discrete models. The relationship between neuron state and output is as follows, which is the discrete Hopfield neural network evolution Equations (5.1) [106] [107] and (5.2) [108].

$$u_i(t+1) = \sum_{j=1}^{n} W_{ij} v_j(t) + I_i \tag{5.1}$$

$$v_i(t+1) = f(u_i) = \begin{cases} 1 & if \quad u_i > 0 \\ 0 & if \quad u_i \leq 0 \end{cases} \tag{5.2}$$

where I_i is the continuous external input of neuron i, and f () is the activation function of the network. When used for associative memory applications, the weights remain stable after the network training is completed. At this point, the network only two variable parameters, which are updated state and the output of the neurons. Due to the random updating of the neurons, the model is discrete and random on the network. When the network is updated, if the weight matrix is symmetric to the non-negative diagonal, the energy function can achieve minimized value until the system converges to a stable state. When DHNN is designing the connection weight, the stable state of the system is proposed. At this point, the available weight matrix W can be obtained through the learned memory of the network. After the learning, the associative network can be achieved by the network weight calculation of output. For trained M models, the DHNN can learn depending on Hebb rules.

The association memory process of the Hopfield neural network indicates that patterns memorized in the neural network are stored on the weight matrix, so the training and learning processes of the weight matrix on the Hopfield neural network becomes particularly important. Hopfield neural networks usually use Hebb learning rules to complete the weight training of neurons. This learning method, proposed by Hebb [289] is the earliest and most famous training algorithm; it still plays an important role in various neural network models. The Hebb rule assumes that when two neurons are excited at the same time, the strength of the connection between them should be strengthened. This rule is consistent with the biological theory of conditioned reflex, which was later confirmed by the neurocyte theory. The Hebb algorithm in the Hopfield neural network can be simply described as one processing node receiving an input excitation signal from another processing node, and if both are at a high excitation level, the weight between processing nodes should be enhanced. Mathematically, the connection weight of the two nodes will be changed according to the product of the two nodes of excitation levels

and can be described as:

Hebb Learning Rule on the Hopfield Neural Network [290] as follows [291]:

$$\Delta W_{ij} = W_{ij}(n+1) - W_{ij}(n) = \eta Y_i X_j \tag{5.3}$$

where $W_{ij}(n)$ represents the connection weight from node j to node i before the $n+1$ adjustment; $W_{ij}(n+1)$ is the node n to node i after the $n+1$ adjustment of connection weight; η is the learning rate parameter; X_j is the output of node j and input to node i; Y_i is the output of node i. The objective of using a Hopfield neural network with Hebb Learning method is for a set of q different input samples $P_{nxq} = [p^1, p^2, \ldots p^q]$. It is to adjusts the weight matrix W to reach a group of input samples p^k, $k = 1, 2, 3, \ldots, q$ as the initial value of the network, the system can converge to the respective input sample vectors.

5.3 Hardware Design

A specific application based on Neuromorphic computing hardware to implement neural network model uses the physical unit simulating the neurons process, and the communication units between the neurons to simulate connections of the neural network. Among them, each neuron and each connection has corresponded to the physical design. The advantage of the hardware implementation is processing speed and easiness of execution to satisfy the real-time requirement. In addition, appropriate hardware design can further, reduce energy consumption on the system. Figure 5.2a shows that the overall fall recognition system and data stream processing workflow. It gives a schematic diagram of neuromorphic computing hardware. The work completed the Hopfield neural network algorithm based on the design of 25 neuron modules construction, and its dependence on the associative memory function to finally achieve the fall recognition result by the sensor signal.

The Hopfield neural network hardware design uses analog circuits composed of resistors, operational amplifiers and other components to describe the neurons. The objective function is converted into the energy function of the neural network, and the model of the corresponding pattern is memorized by the equilibrium point of the network energy function. Hopfield network as a recurrent neural network that includes feedback connection from output to input. The designed Hopfield neural network model is shown as in Figure 5.2b.

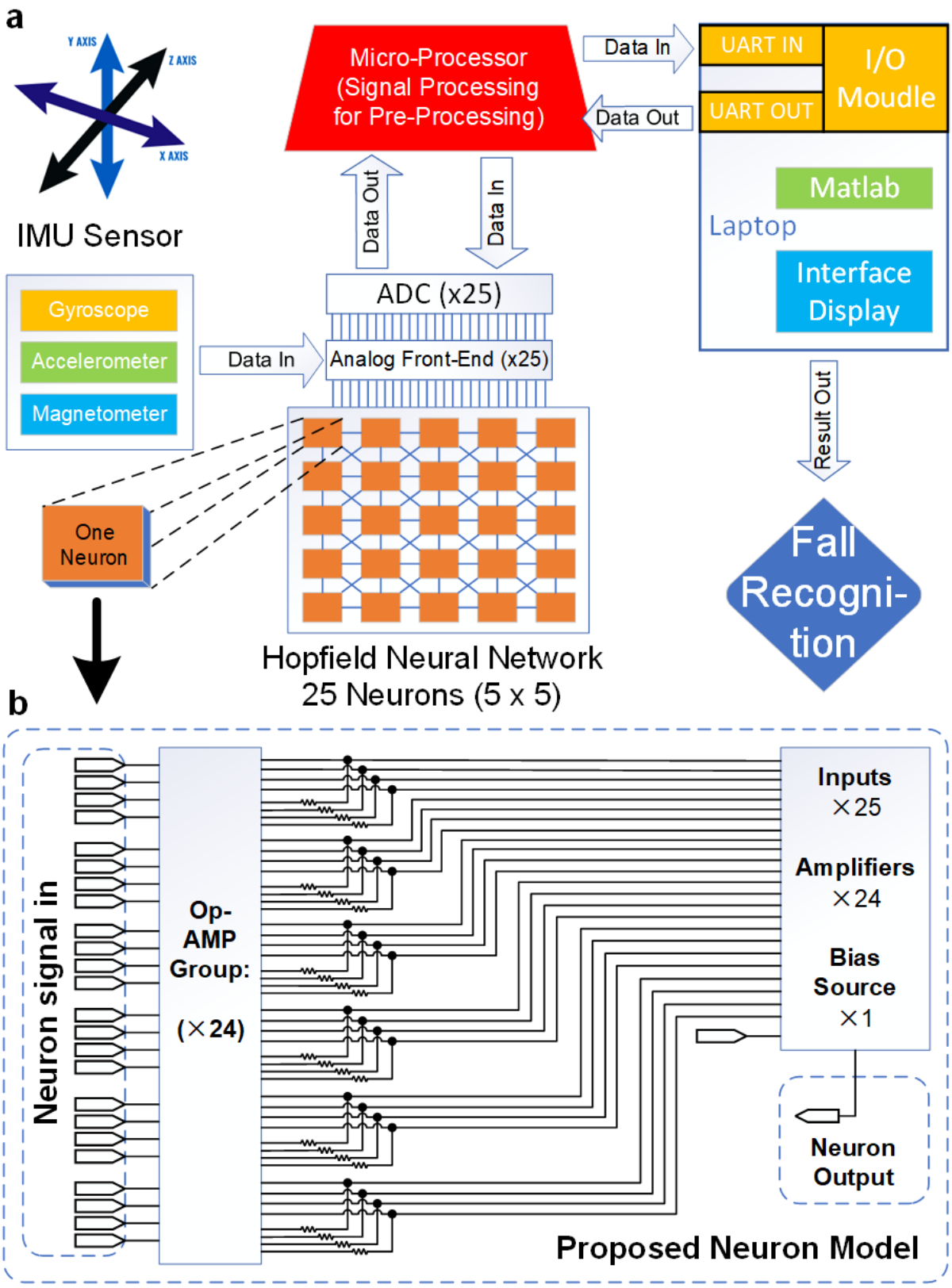


Figure 5.2: Hardware architecture for system blocks.

5.3.1 Neuron's Hardware Design

The state of neurons expresses the result of the neuromorphic processing information. Neurons are the smallest unit of information processing on the neural network. These neurons are connected by a rule to form a physiological neuron network system to the brain. Among them, the strength of the connection between each neuron changes adaptively according to the excitation degree of the external signal. Each neuron presents a state of excitation or inhibition state with the combined magnitude of the received multiple excitation signals. Excitation state refers to the change of neurons from relative rest to relative activity, while inhibition state refers to the change of neurons from relative activity to relative rest. As a result, there are two kinds of connections between the transmission of information between neurons. Positive connections stimulate each other, and negative connections inhibit each other.

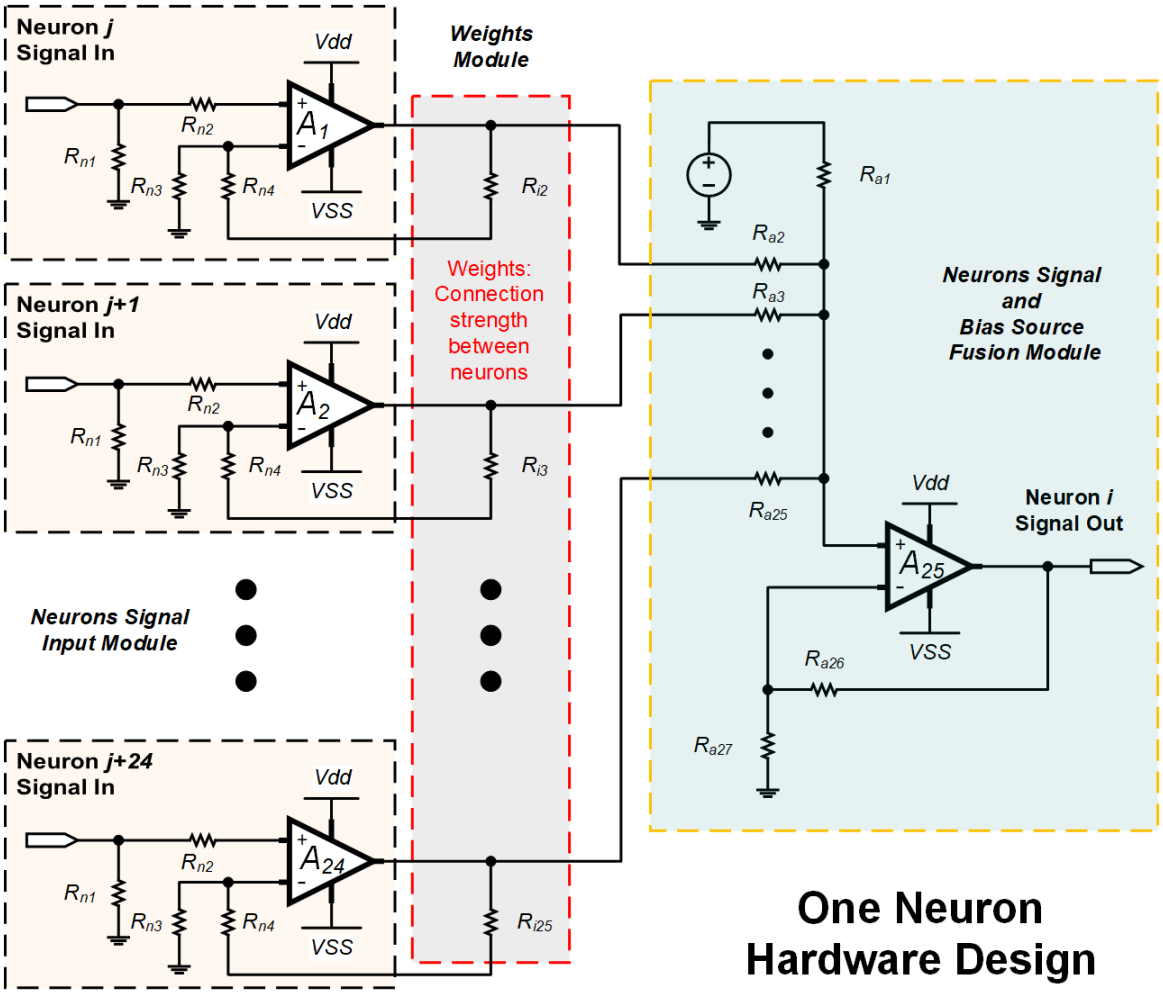


Figure 5.3: The Hopfield neural network of one Neuron circuit design.

The structure of the discrete Hopfield neural network is a single-layer full feedback network architecture, and in this work, it used a total of 25 neurons as 5 * 5 network. Each neuron through the connection weight to receives information from the output of all other neurons. The purpose

is to make the output of any neuron controlled by all other neurons, which means each neuron can restrict each other. The essence of associative memory on the discrete Hopfield neural network is that the memory sample to be stored and represented by a vector. After inputting the memory sample, the neural network weight is stable on the memory sample after evolution. When the input of the neural network is a nonlinear memory sample, the output of the network should be either stable in the memory sample; or it is stable in the nonlinear sample. Therefore, the output of the neural network should be stable regardless of the external input.

The dynamic differential system of biological neurons on the neural network can be simulated by the operational amplifier to achieve learning and associate function such as the biological brain. Therefore, in the Figure 5.2, each group of operational amplifiers and their associated resistors constitute a model for representing a neuron function. Each neuron has two sets of inputs, one is a constant bias source signal, and the other is a positive or negative feedback connection from the output of other operational amplifiers (other neurons).

Figure 5.3 shows the circuit design for one of the neurons, which is a general neuron module for the neuromorphic computing of the Hopfield neural network. Based on the neurons connect concept, it includes Neurons signal input module, Weight module and Source fusion module. There are three circuit model designs all based on sample analog components, which are amplifier and resistor. The Neurons signal input module is designed by 25 amplifiers, each amplifier as an input port. The Neurons' signal input module is to link the other neurons' output, so that the work is complete in 24 modules to connect with the other 24 neurons (This Hopfield neural network designed 25 neurons, and each neuron links to each other neurons). The Weight module is constructed by resistors, and it works for amplification factor control. The Weight module is designed for neuron connection strength representation, which depends on the resistors to control the amplifier gain. The corresponding parameters are calculated by Hebb learning when the neural network is training. For the Source fusion module, it is a summing unit designed by an amplifier that is additional computing to fuse the bias source and 24 neurons input. Meanwhile, this module is a data input interface on the Hopfield neural network hardware. The 25 neurons of the Hopfield neural network can input 5 * 5 matrix data size to processing, which is each neuron to match one of the matrix dots (Arranged dot by rows to input).

So, one neuron's design should include 24 amplifiers on the Neurons signal input module and one amplifier on the Source fusion module, meaning each neuron design has 25 amplifiers. The whole 25 neurons are 25 * 25 = 625 amplifiers completed neuron network structure. At this point, the Cadence Pspice hardware simulation chooses the OPA227 amplifier on the library. All the hardware components' hyper-parameters has shown in the Table 5.1. As the amplifier of the Gain–BandWidth(GBW) is a maximum of 8 MHz, which is easy to real-time process the IMU sensor's sampling rate for 50Hz. And the system's total power dissipation is 493.75 uw, which is lower by more than an order of magnitude for the traditional embedded system (Normally, embedded system hardware power consumption for 5 -10 watts). Following these output, the

low power and real-time neuromorphic computing hardware for HAR tasks are completed.

Table 5.1: Amplifier hyper-parameters

Hyper-Parameter	**Value**
1. Wide Supply Range	±2.5 V to ±18 V
2. GBW (Typ)	8 MHz
3. IMU sensor sampling rate	50 Hz
4. Low Noise	3nV/$\sqrt{Hz}$
5. High CMRR	138 dB
6. High Open-loop Gain	160 dB
7. Slew rate (Typ)	2.3 V/μs
8. Operating Power Current	790 uA
9. Voltage Supply	12V
10. Input Bias Voltage	0.001V
11. Input Equivalent Noise	0.2 pA/$\sqrt{Hz}$
12. System Power Dissipation	493.75 uw

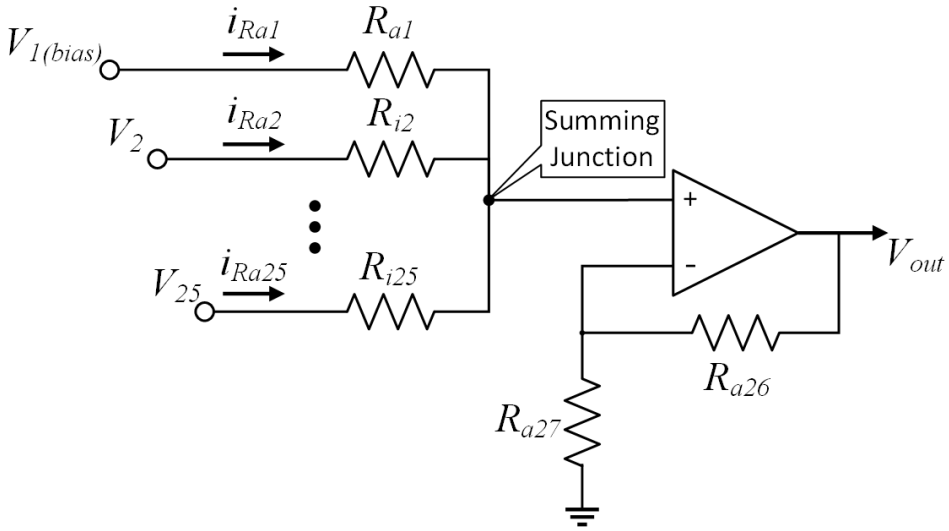


Figure 5.4: The equivalent circuit of Hopfield neural network.

In the equivalent circuit design of one neuron in Figure 5.4, the internal membrane potential of the neuron i is V_i (i = 1, 2, …, n without self-position, it is a bias source input point), the transmission resistance of the cell membrane is R_i, the output potential of the neuron is V_{out}, the external input current is V_{bias}. The resistance R_i (i = 1, 2, …, n) to simulate the synaptic

properties between the i and j neurons. Among them, f_i represents the transfer function of the neuron i, and defines $W = R_{ij}$ $(i, j = 1, 2, \ldots, n)$ as the weight coefficient matrix of the neural network. Each operational amplifier simulates the nonlinear characteristics between the input and output by the neuron through the following calculation [292]:

$$V_{out}(t) = f_i\left(\sum_{i=1}^{n} V_i(t)\right) \tag{5.4}$$

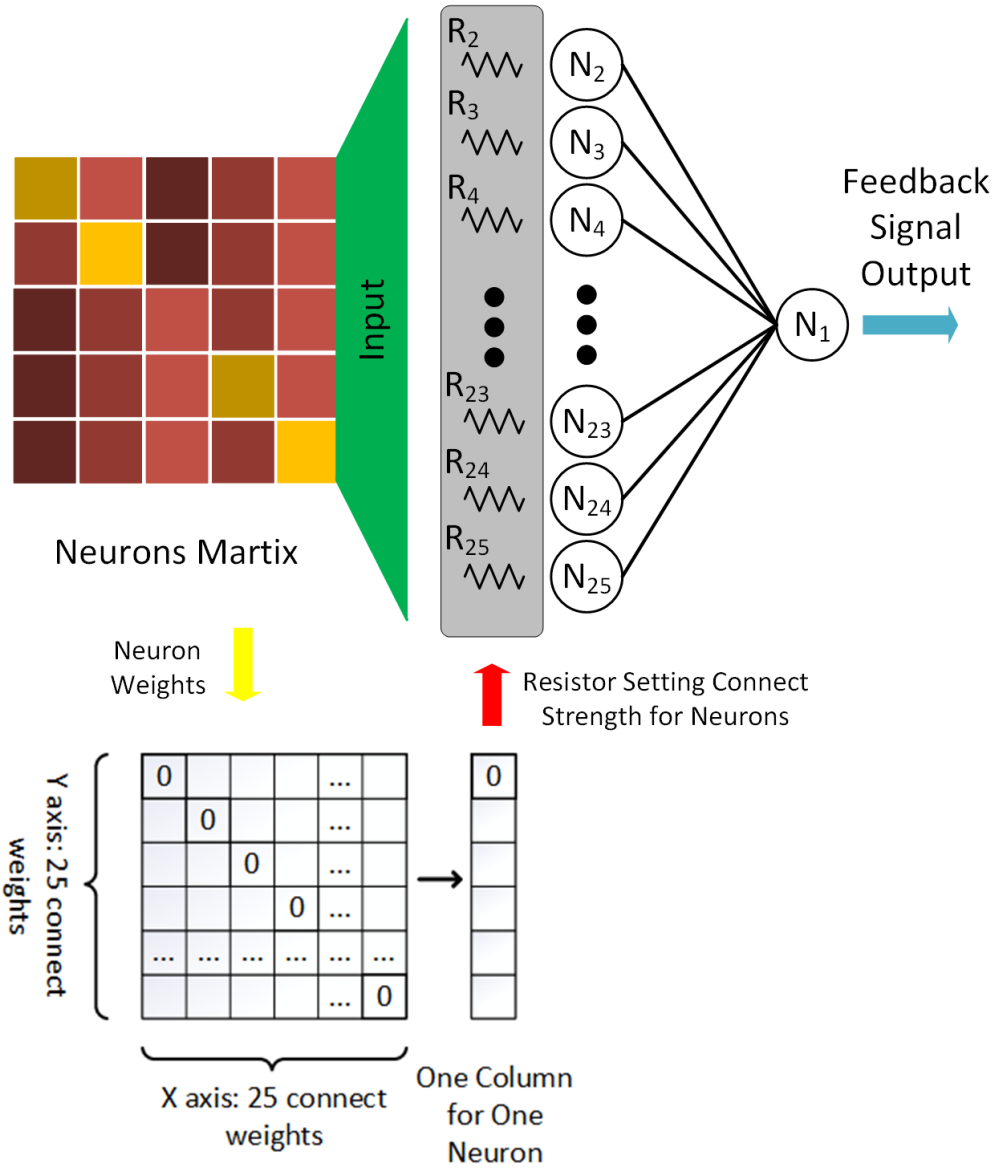


Figure 5.5: The Hopfield neural network of Neuron Matrix and Weight.

At this point, the function of the amplifier as a summing circuit is to simulate the activation function of the neural network, and the parallel resistance can adjust the connection strength between the neurons. In the equivalent circuit calculation of a neuron, which is based on Kirch-

hoff's Current Law (KCL), in a lumped circuit, all the node at any moment, it is the algebraic sum on the identity of the branch currents for all outgoing nodes is zero. The inflow current and outflow current at the input node of the amplifier is maintained balance to achieves neuron voltage signal output. The rest 24 neurons can be done in the same manner that is following by corresponding connection strength. The weight is derived from the matrix value obtained by training on the Hebb algorithm. Figure 5.5 shows the details about the relationships between the weight and matrix.

5.4 Evaluation and Results

In this study, the Cadence PSpice has been used for circuit design and hardware simulation, which is integrated with MATLAB Simulink to complete the system construction and validation. Following the hardware simulation result, which is can show the hardware performance information. It is can be used for theoretical analysis of how to complete low-power and real-time neuromorphic computing hardware. However, hardware simulation tools cannot layout the circuit design, so the size of the hardware design is unable to known, but it can achieve the size result for future work on hardware fabrication. Meanwhile, for data analysis, the IMU sensors captured 5 classes of human activities as a dataset to achieve data classification for fall detection, which includes Fall, Sit down, Stand up, Under and Walk sensor signals. The neuromorphic computing of the Hopfield neural network algorithm was used to classify the human fall activity data as a positive sample and other activities as negative samples to complete fall detection.

5.4.1 Data Preprocessing for Feature Extraction

First, the matrix sample template of human fall activity is achieved by feature extraction computing that is based on data preprocessing for IMU sensor data. There are 9 axis signals acquired by the IMU Sensors, which is sampling frequency is 50 Hz and capture 5 classes activities for 5–10 s. Figure 5.6a shows the raw IMU signal output to the Fall activity. The project uses a construction matrix to fuse 9-axis sensor data, and then depend on threshold selection to identify human fall activity. The processing results are shown in Figure 5.6b. It shows the output of the human fall signal in the data has a different value with other classes human activities. To adjust Hopfield Neurons circuits design, a sample template of human fall activity is extracted from the fused computing results. Due to the symmetry characteristics of the Hopfield neural network, it flipped the fall signal output to 5 * 5 feature template to facilitate the visualization of the weight output. The 5 * 5 Fall sample template as shown in Figure 5.6c, which is matched with 25 neurons matrix on the circuits design. For the hardware design, the Neuron Weight is calculated by the Hebb Learning algorithm. The 5 * 5 Fall activity feature matrix is applied as input to the Hebb Learning algorithm to train with Hopfield neural network architecture of 25

neurons. The trained Weight output is shown in Figure 5.6d, which is shown the neuron connection relationship on the neural network architecture. Finally, setting the weight value into the Hopfield neural network is taken as circuit parameters to complete hardware design. Algorithm 5.1 describes the main steps of the sensor data feature extraction and binary conversion, where the first step involves calculating the 9 axes of three sensors data, which is shown in Figure 5.6a. After the threshold value processing, it achieves a 5 * N binary matrix for which the results are shown in Figure 5.6 B. The suitable threshold values were obtained through iterative trial and error steps, so that the fall activity with other data can be accurately classified. Finally, a sliding 5 * 5 window box decides which 5 * 5 feature map for a fall activity is captured, and it is illustrated in Figure 5.6 C.

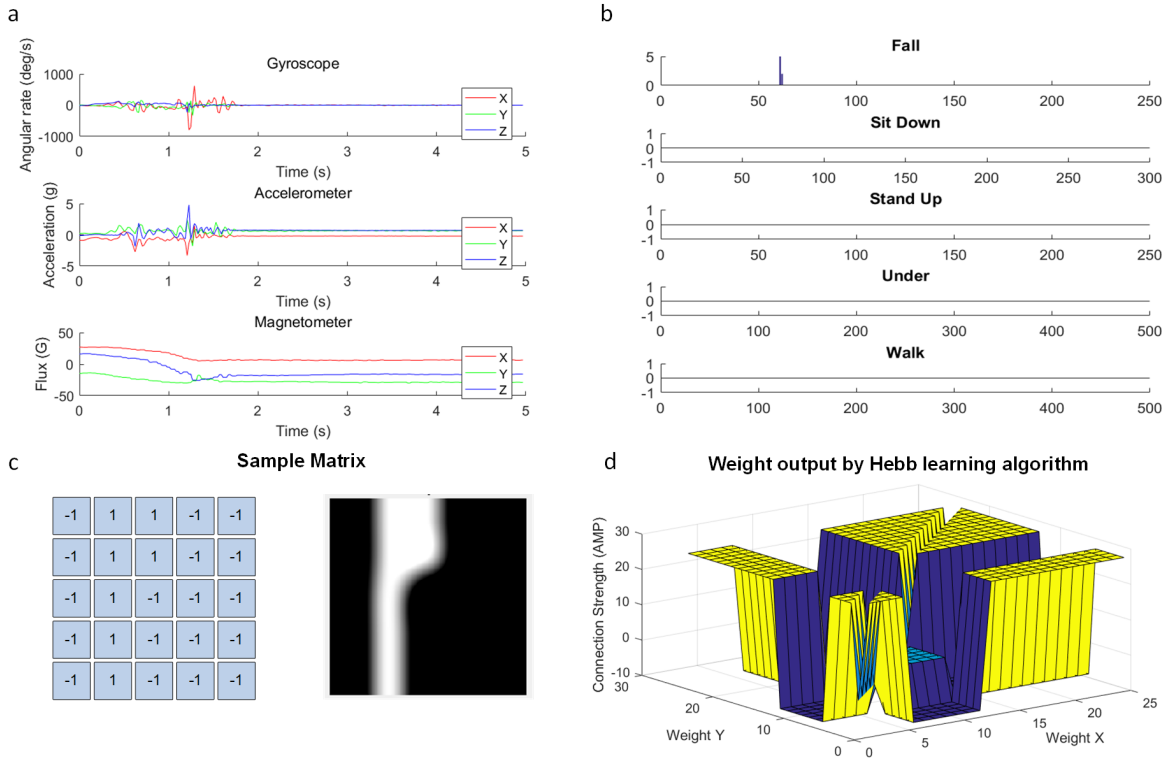


Figure 5.6: The Sensor data converted into Hopfield neural network of sample and weight, (a) Raw IMU sensor signal output, (b) Sensor data feature extraction and binary conversion, (c) Fall activity sample matrix of feature map, (d) Hebb learning algorithm output the Hopfield neural network weight.

5.4.2 Comparison with State-of-the-Art Machine-Learning Algorithm and Discussion

In the test, saving the Hopfield Neuron's output feedback signal as a binary matrix (high voltage level represents Binary value 1, and low voltage level represents Binary value -1), and then using the cosine distance method [293] the feedback matrix is computed to achieve the confidence

level and accuracy of the classification. Finally, following the accuracy calculation (True Positive (TP) + True Negative(TN))/(True Positive (TP) + False Positive(FP) + True Negative(TN) + False Negative(FN)), the fall activity classification result that has been achieved is 88.9% on the hardware-based design. Table 5.2 shows comparison results for different algorithms and their respective platform accuracy. Such as Li et al. [294] worked on the same dataset, and they used traditional machine-learning algorithms on the MATLAB software-based platform, which is a good software comparison result with neuromorphic computing for the HAR task. Meanwhile, there are many machine-learning algorithms on low-power platforms to achieve fall detection. Based on the ZYNQ hardware of the FPGA platform, Nguyen [295] implements the Gaussian Mixture Model by digital design, and Garg et al. [296] deploys the Deep Neural Network (DNN) algorithm through the ARM processor to calculate the sensor data of the fall activity. However, through comparing accuracy, our implementation is more accurate than their classification. In addition, comparing training dataset requirements, it can be found that machine learning and deep learning rely on a large number of samples to learn features. The learning strategy of neuromorphic computing is more friendly with limited sample datasets. The model template of the fall sensor signal is obtained by preprocessing the fall feature extraction. Depending on the Hebb learning method, the associative memory of the sample is realized on the Hopfield neural network. In this way, only one sample learning is requested on the training and then the fall activity recognition weight is achieved for the Hopfield neural network.

Algorithm 5.1: Sensor data feature extraction and binary conversion.

Initialize matrix M(s, s) to full zero matrices.
Load three Sensors data:
[Gx, Gy, Gz] = Gyroscope Sensor data Matrix [:1, 2, 3]
[Ax, Ay, Az] = Accelerometor Sensor data Matrix [:1, 2, 3]
[Mx, My, Mz] = Magnetometor Sensor data Matrix [:1, 2, 3]
Require: :
$M_1(n) = (abs(Gx) + 1) * (abs(Gy)) + 1) * (abs(Gz)) + 1);$
$M_2(n) = (abs(Ax)) + 1) * (abs(Ay)) + 1) * (abs(Az)) + 1);$
$M_3(n) = (abs(Mx)) + 1) * (abs(My)) + 1) * (abs(Mz)) + 1);$
$M_4(n) = (1/2) * (Ax * My * Gz - Gx * My * Az)$
$M_5(n) = (1/2) * (Gx * Ay + Ax * My + Mx * Gy - Gx * My - Ax * Gy - Mx * Ay)$
Binarization and Feature Extraction:
Threshold value: t = [16, 24, 32, 40, 48];
for i = 0:4 **do**
out[n][i] = (M > t[i]) ? 1 : − 1;
Update the Matrix M with the Binarization line by line.
return **Matrix M(5:N)**;
Sliding window box **do** M = M(starti:starti + 4, :)
end
return **Matrix M(5:5)**;

Table 5.2: Comparison table with different methods and hardware environments

Project	Algorithm	Environment	Training Dataset	Accuracy
Our Work (Hardware based)	Hopfield Neuron design on analog circuit	PSpice (Analog Design) + Matlab Simulink (Only Data Transit) Co-Simulation	1 Sensor sample to generate a standard pattern for each activity	88.9%
Our Work (Software based)	Hopfield Neural Network	Matlab (Code)		94.4%
Li *et al* [294]	Support Vector Machine (SVM)	Matlab (Code)	Using a 70% Sensor dataset as training data (20 volunteers * 3 repetitions * 70% = 42 training samples for each activity)	79.83%
	Artificial Neural Network (ANN)		Using a 70% Radar dataset as training data (20 volunteers * 3 repetitions * 70% = 42 training samples for each activity)	85.53%
Nguyen [295]	Gaussian Mixture Model-Hidden Markov Model (GMM-HMM)	FPGA (Digital Design)	DUT-HBU dataset is used and all video data are compressed in avi format and captured by a single camera in a small room with the changeable conditions	87.3%
Garg *et al* [296]	Deep Neural Network	ARM (Tensorflow)	A sample of data points from 2.5 seconds before and after the spike, making a total of 200 * 5 = 1000 discrete samples [296]	86.2%

Furthermore, our proposed project also compares the software-based (MATLAB) output with Hardware-based approaches (PSpice-Matlab Simulink co-simulation results). For the hard-

ware circuit design, slight differences can occur between analog signal transmission, which is different from the software environment and its respective ideal calculation. However, Hopfield neural network can ignore part of analog noise, which is based on associative memory function to restore noised distortion analog signals. It is depending on the signal associative recovery effect on neuromorphic computing. Compared with conventional machine-learning algorithms, better outcomes are obtained with neuromorphic computation after careful extraction of the function and configuration of the model models. Both software-based and hardware-based HAR task Hopfield neural network results are good than other related work. Especially, Li et al work with the same dataset, but the machine learning algorithms' accuracy are low than the neuromorphic computing. However, it prevents the cycle of internal loss and noise effects on the analog circuit, relying on the binary approximation to create the neuromorphic computing algorithm. For the complexity and precision requirements of analog and digital conversion equipment, neuromorphic computing is closely integrated with the analog circuit's performance, which enables systems to reduce the impact on the accuracy.

5.5 Summary

In this chapter, a neuromorphic computing hardware design both in the software and hardware domains with a sensor data fusion method is proposed. It is co-designed especially for health care applications. where it uses neural network hardware to achieve low-power and real-time operation for multiple sensors data in an embedded system. The neurons are stimulated by amplifiers, resistors and other components, which are used for processing inhibition and activation signals on the neurons. The neural network is then constructed by combining neurons. By designing the front-end feature extraction algorithm, the multiple sensor data fusion results are achieved, which are then passed to the hardware-based neural network to finally generate the recognition result. This proposed hardware and software co-design solution can be used for smart wearable devices in healthcare applications and can perform local real-time data processing, which does not require the additional cloud computing processes for data interaction with sensors. The low-power and real-time hardware can simulate neuromorphic computing with the help of which machine learning can be applied for data processing. Results show that through simulations, the hardware-based neuromorphic computing of the fall data reached an accuracy of 88.9% which is at least 1.6% greater than the state-of-the-art algorithms, such as SVM, ANN and DNN. The neuromorphic computing hardware design proposed in this chapter is a simple, fast and reliable solution. In the design, a straightforward circuit design is used to control the weight setting from the Hopfield neural network which is based on the resistance to vary the amplification factor, which essentially maximizes the reusability and flexibility of the circuit design. The proposed design can reduce the design time and help avoid separate debugging processes of the amplifier to easily adjust the weight setting. Following hardware simulation parameter

setting, and depending on the theoretical analysis, the low power and real-time neuromorphic computing hardware for HAR tasks are completed. I believe that the design has the potential to exploit the neuromorphic computing framework for data fusion in healthcare applications.

Chapter 6

Human Activity Recognition and Case Study of Benchmark NodeNS Sensor Datasets

This Chapter provides a new benchmark dataset for 3D point cloud classification in which the manually labelled human activity data exceeds 100 point clouds per frame and is capable of meeting the training needs for data-intensive learning approaches. In this study, a case study is considered for evaluating the benchmark using a deep LSTM neural network, which demonstrated a significant performance improvement over the state-of-the-art human activity recognition area. To date, numerous types of collection devices have been used in the recognition of human activities. However, due to the scarcity of training data, the task of 3D point cloud labelling has not yet made significant progress. To overcome this challenge, it is aimed to deduce this data requirements gap, allowing deep learning methods to reach their full potential in 3D point cloud tasks. The dataset used for this process is comprised of dense point clouds acquired with the static ground sensor by the NodeNs company supported MIMO radar (NodeNs ZERO 60 GHz IQ radar) [297]. It contains multiple types of human being data ranging from one to four individuals and encompasses a range of human action scenarios, including standing, sitting, picking up, falling, and walking. Furthermore, it also investigated sensor locations and requirements for human being data collection that is from a single subject to multiple subjects, as well as identified and analysed various sensing devices and applications that collect activity data. In this regard, a thorough study is conducted on several benchmark datasets, examining sensors, characteristics, activity categories, and other data. Finally, it compares and analyses the activity recognition methods used in several benchmark datasets based on the current study. Unlike existing devices, the new NodeNs sensor provides more accessible and straightforward point cloud data to capture human movement information. Depending on an advanced detection algorithm to process point cloud data it achieved more than 95% accuracy on the benchmark dataset.

6.1 Introduction

Human Activity Recognition (HAR) using low power embedded systems has recently made significant progress. It has attracted increasing attention in many research areas, including machine learning, pattern recognition, context awareness and human body perception. In the past few years, the HAR field has been one of the most prominent topics in much research. It aims to understand a person's daily behaviour by analysing their movements and their surrounding living environment [298]. The information is gathered from different hardware embedded in smartphones, wearables and fixed position capture devices [299].

In the process of data collection, researchers have to face many difficulties such as technical challenges, subjects' privacy issues [300], and organizational permissions [301]. Due to these difficulties, few datasets are currently available in sensor-based activity recognition. Some well-known repositories have collected sensor benchmark datasets related to human movements, which are described in detail below, including focused area and data collection devices:

- UCI Machine Learning Repository [302]: Datasets hosted by the University of California Irvine Machine Learning Repository, in CSV format. The dataset in this repository contains information about attribute types, missing values, target domains, etc. The main advantage is the flexibility of data donation and the diversity of data. It contains smart phones and wearable sensor-based activity recognition datasets. Such as the Single Chest benchmark dataset [303]. It is continuous data collected by wearing a 52Hz accelerometer on the chest, which includes climbing stairs, standing, talking, walking and working.

- Datasets based on wearable sensors: Wearable devices are designed with flexibility in mind for the customer's daily use. Lightweight, stylish and comfortable wearables with embedded sensors for activity monitoring have been used in multiple datasets. The REALDISP benchmark dataset [304] uses wearable devices including Accelerometers, Gyroscopes and Magnetic sensors to collect continuous movement signals from the human body.

- In medical activity related datasets: Medical activity identification has many applications in remote monitoring of patients, pregnant women and elderly people. Among them, the medical activity-related benchmark dataset Nursing Activity [305] uses iPod's Accelerometer for monitoring nursing activities in the hospital.

- Dataset based on smartphone sensors: Nowadays, smartphones have become the daily companions of most people. Therefore, it is easy to monitor user activity data by embedded sensor from smartphones. Due to the availability of sensors and cost-effective methods, activity recognition based on smartphone sensors has attracted the interest of many researchers. The Smart Devices benchmark dataset [306] uses the Accelerometer,

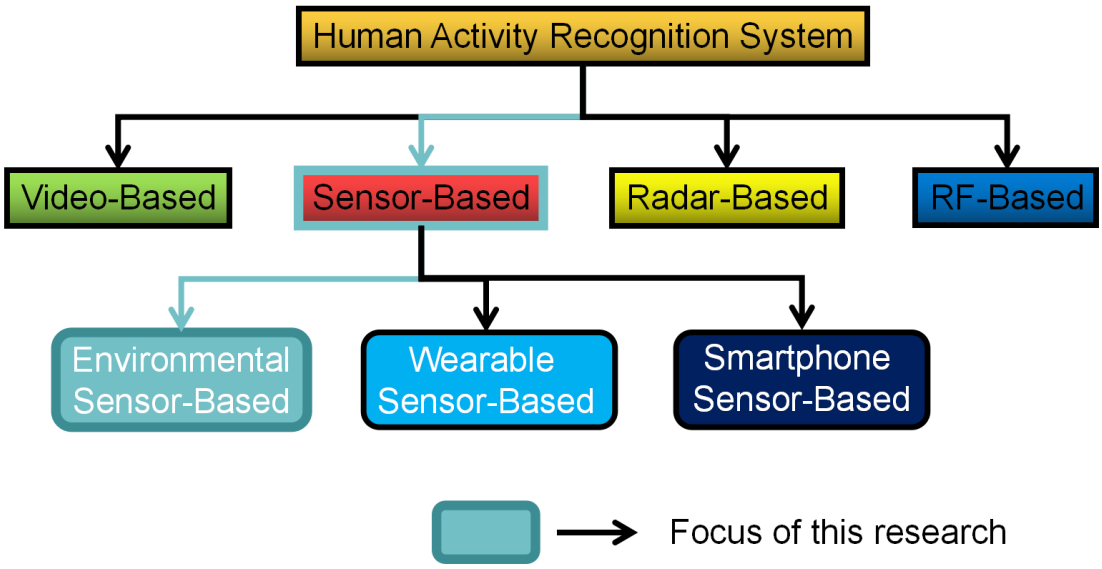


Figure 6.1: Approaches Employed for Human Activity Recognition.

Gyroscope, EOG, and Pressure sensor of Smartphone and Smartwatch to complete Human behaviour recognition.

There are currently four main types of technologies widely used to perform HAR: Wearable Sensors, Radar, Radio Frequency (RF) signals, and Cameras. In order to achieve the goal of human activity recognition, there is a need for systems with perceptual capabilities [204]. Considering cost constraints, the two most common methods used for this purpose are based on sensors and radar to collect activity data. Their principles are different, radar mainly obtains the distance, speed and angle of the object by sending electromagnetic waves and receiving echoes. It can realise object detection with higher precision through the Doppler frequency shift principle [307]. Moreover, it is not constrained by power, and the hardware performance can be better released. Sensors are based on wearable devices to collect physical quantities of human movements, such as angular velocity [308], linear velocity relative to the ground [309], acceleration [310], vibration [311] and direction based on contact [312]. It is low cost, easy to use, and less limited by usage scenarios. It is now necessary hardware for mobile portable devices to realise motion recognition. Depending on these research points, the Figure 6.1 draws the categorises of activity recognition, and points out the environmental sensor-based method is our focused research work. The proposed benchmark dataset is based on the new NodeNs ZERO 60 GHz IQ sensor [297] of MIMO radar that provides more accessible and straightforward point cloud data to capture human movement information. Point cloud data refers to a set of vectors with geometric positions in a spatial coordinates system. It is sampling points that scan objects and record them as points data, each of which contains three-dimensional coordinates, and the number of points is large and dense which means point cloud data.

The main contributions of this paper are as follows:

• We construct a large-scale point cloud benchmark dataset for HAR. It is a publicly available, high-resolution dataset that contains more classes and more subjects than the current HAR datasets.

• In contrast to previous research, we explored state-of-the-art methods in the domains of the HAR field. We have discussed the technology from traditional handcrafted features to recently developed deep learning.

• We evaluate the advanced detection algorithm to process this benchmark point cloud dataset under consistent experimental conditions. This provides the literature with extensive baseline results for future research on HAR tasks.

In this chapter, Section 6.1 has a brief introduction to human activity recognition systems. Section 6.2 describes the raw-sensor device's human activities and data collection method. Furthermore, the work has provided a case study and evaluated results in Section 6.3, which shows the benchmark challenges in activity recognition. Finally, Section 6.4 discussed the HAR benchmark datasets and compared some related works to prove our contributions, and Section 6.5 indicate directions for future research.

6.2 Materials and Methods

For daily activities of humans, the complexity of the actions may be different, and human movements are classified according to various situations, including the number of activities [313], the types of activities [314], data collection energy consumption [315], and significance [316].

Figure 6.2 shows, human activities can be divided into three basic types: static activities [317], dynamic activities [318] and activities with posture transitions [301]. Static activities generally refer to activities, such as Sitting and Standing, which basically do not involve human movement. Compared with static activities, dynamic activities have a motion trend and it represents more intense environmental interactions. For example, walking, falling and other short-lived behaviours, which are the interaction between human body posture and behaviour [298]. It is usually some high-frequency behaviours of the human being in daily life. Activities with posture transition are combinatorial behaviours that are aggregated through multiple basic activities, and multiple users may participate during the behaviour [208].

Static activities (such as Sit and Stand, etc.) are more accessible to recognise than regular activities (Walking, Picking up some things, etc.). However, highly similar posture transitions, such as Sitting and Standing, will cause great complexity in the case of separation in that the activity feature space has obvious overlap. In the static activities of Sit and Stand, which means completed action without movements, and dynamic activities or posture transitions for Sitting down and Standing up which means action in progress, it is continuous movement. In addition, dynamic activities (standing up and sitting down) with a high degree of similarity in the activity

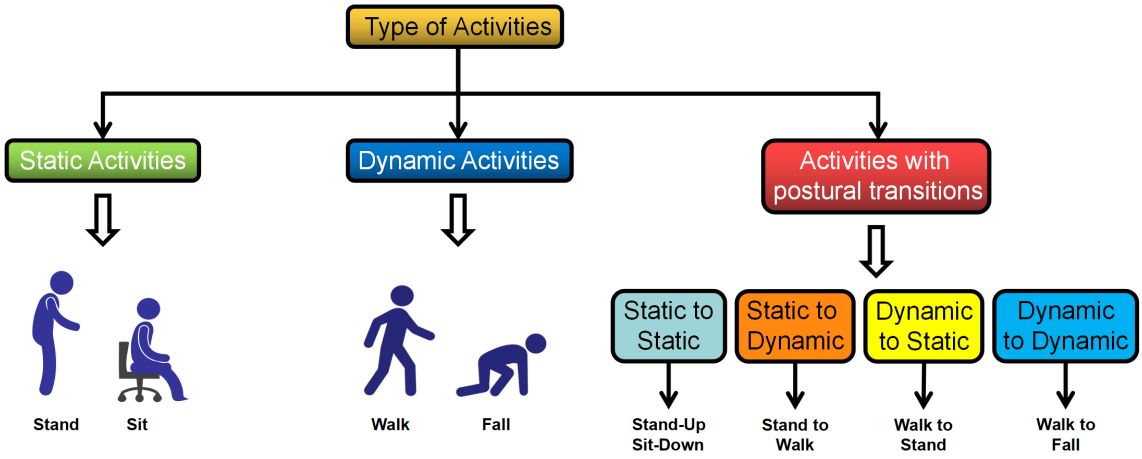


Figure 6.2: Different Types of Human Activities.

feature space are also difficult to distinguish due to related activity patterns.

In most cases, the correspondence between the activities performed is not similar during the entire activity period, which makes the HAR task more difficult. For example, sitting and standing are very similar, so it is hard to tell them differently from frame data. However, they are remarkable gaps from walking and are easy to separate from activities data at this point. Therefore, transition activities can be further divided into four types: static to static posture transition (Stand to Sit), static to dynamic (Stand to Walk), and dynamic to static (Walk to Stand) and dynamic to dynamic (Walk to Fall).

Several variations have been built that represent various difficulty levels of the human activities dataset based on the selected activity objects. This allows researchers to explore the robustness of existing classification methods on benchmark datasets in more extreme real-world scenarios.

6.2.1 Proposed Benchmark Dataset Overview

The HAR data in this work that is collected at the University of Glasgow with a group of 4 volunteers. A concept figure 6.3 represents the activities recording environment and equipment arrangement plan. Figure 6.3a shows the NodeNs sensor has been fixed setting at the table and keeps 2 meters to acquire for human beings signal. There are 15 different sequences of continuous activities that were designed, which are from a single subject to multi-subjects (maximum 4 subjects). All the participants were asked to simulate the designed daily activities in the setup zone in front of the NodeNs sensors, approximately 3m × 3m, which the experimental setup zone has been shown in figure 6.3b. The dataset was divided into 4 classes, that is, a total of 1011 data samples/files, each represents a particular number of subjects and activities. The main data folder is subdivided into 15 folders corresponding to the 15 classes. The detail about the benchmark dataset are shown in Table 6.1.

Table 6.1: Benchmark Dataset details about the Classes and Samples

Number of Subjects	Activity	Number of Subjects and Corresponding Activity	Description	Number of samples
1	Sit down, Stand up, Walking, Fall, Pick up.	• P1 (All activities)	• One subject performing each Action single	101 / Activity class * 5 = 505
2	• Sit down + Stand up, • Sit down + Pick up, • Stand up + Pick up	• P2(Sit) + P1(Stand), • P1(Sit) + P2(Stand), • P2(Sit) + P1 (Pick), • P1(Sit) + P2 (Pick)	• One subject performing the action of "Sitting" and one subject performing the action of "Standing", at the same time. • One subject performing the action of "Sitting" and one subject performing the action of "Pick up a box", at the same time. • One subject performing the action of "Standing" and one subject performing the action of "Pick up a box", at the same time	40 + 40 + 39 = 119
3	• Sit down + Stand up + Pick up, • Stand up + Pick up + Sit down, • Pick up + Sit down + Stand up.	• P3(Sit) + P2(Stand) + P1(Pick), • P1(Sit) + P3(Stand) + P2(Pick), • P2(Sit) + P1(Stand) + P3(Pick), • P1(Sit) + P3(Stand) + P2(Pick), • P3(Sit) + P2(Stand) + P1(Pick), • P2(Sit) + P1(Stand) + P3(Pick), • P3(Sit) + P2(Stand) + P1(Pick)	• One subject performing the action of "Sitting", one subject performing the action of "Standing" and one subject performing the action of "Pick up a box", at the same time. • One subject performing the action of "Standing", one subject performing the action of "Pick up a box" and one subject performing the action of "Sitting", at the same time. • One subject performing the action of "Pick up a box", one subject performing the action of "Sitting" and one subject performing the action of "Standing", at the same time	62 + 93 + 62 = 217
4	• Sit down + Stand up + Pick up +Sit down, • Sit down +Sit down + Stand up + Pick up, • Pick up + Sit down + Sit down + Stand up, • Stand up + Pick up + Sit down + Sit down.	• P3(Sit) + P2(Stand) + P4(Pick) + P1(Sit), • P1(Sit) + P3(Stand) + P2(Pick) + P4(Sit), • P4(Sit) + P1(Stand) + P3(Pick) + P2(Sit), • P4(Sit) + P1(Stand) + P2(Pick) + P3(Sit), • P3(Sit) + P2(Sit) + P4(Stand) + P1(Pick), • P3(Pick) + P2(Sit) + P4(Sit) + P1(Pick), • P3(Pick) + P2(Sit) + P4(Sit) + P1(Pick), • P3(Stand) + P2(Pick) + P4(Sit) + P1(Sit), • P1(Stand) + P2(Pick) + P3(Sit) + P4(Sit)	• One subjects performing the action of "Sitting", one subject performing the action of "Standing", one subject performing the action of "Pick up a box" and one subjects performing the action of "Sitting", at the same time. • One subjects performing the action of "Sitting", one subject performing the action of "Sitting", one subject performing the action of "Standing" and one subjects performing the action of "Pick up a box", at the same time. • One subjects performing the action of "Pick up a box", one subject performing the action of "Sitting", one subject performing the action of "Sitting" and one subjects performing the action of "Standing", at the same time. • One subjects performing the action of "Standing", one subject performing the action of "Pick up a box", one subject performing the action of "Sitting" and one subjects performing the action of "Sitting", at the same time	30 + 30 + 80 + 30 = 170

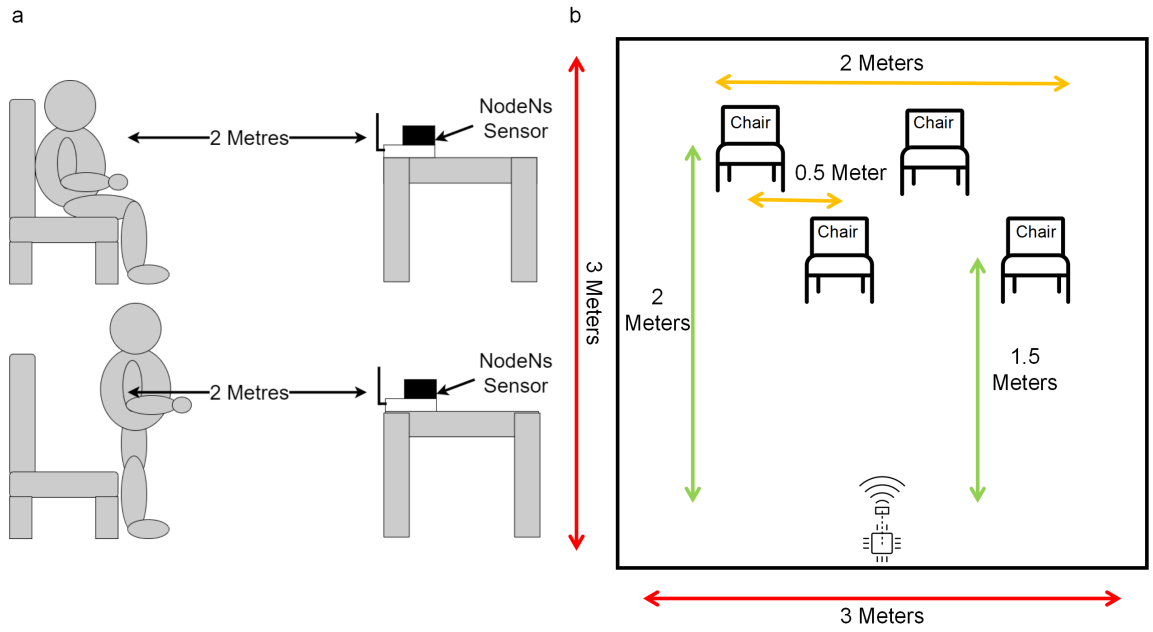


Figure 6.3: Experimental Setup Data Collection.

6.3 Case Study

For a clearer picture of the maturity of the point cloud-based HAR task to the designed benchmark dataset, this work uses representative methods to benchmark test. The goal is to identify the contributions and limitations of the current work on the HAR task. In this case study, the first is data preprocessing to the point cloud data through DBSCAN (Density-Based Spatial Clustering of Applications with Noise) for clustering computing and then training and testing the HAR task on the LSTM (Long Short-Term Memory) neural network for classification to demonstrate the functionality of the benchmark dataset. Meanwhile, to prove that training on data sets with accurate attributes can help improve classification performance. The evaluation of LSTM is based on the labelled dataset are randomly divided into two subsets: training set (80%) and test set (20%), and make sure that the training set and test set contain objects from different scenes. The workflow is shown in the figure 6.4.

6.3.1 Point Cloud Data Pre-processing

Point cloud data analysis often needs to segment the point cloud data and extract the region of interest. The methods of point cloud segmentation include model fitting, graph-based computing, deep learning, and clustering is popularly deployed. For 3D point cloud data, its feature attributes are usually used for clustering, and feature extraction or transformation is performed for each point space or local space. In this way, various attributes can be obtained, such as normal vector, density, distance, elevation, intensity, which segments point clouds with different attributes. For example, for density clustering, a cluster is defined as the most extensive set of

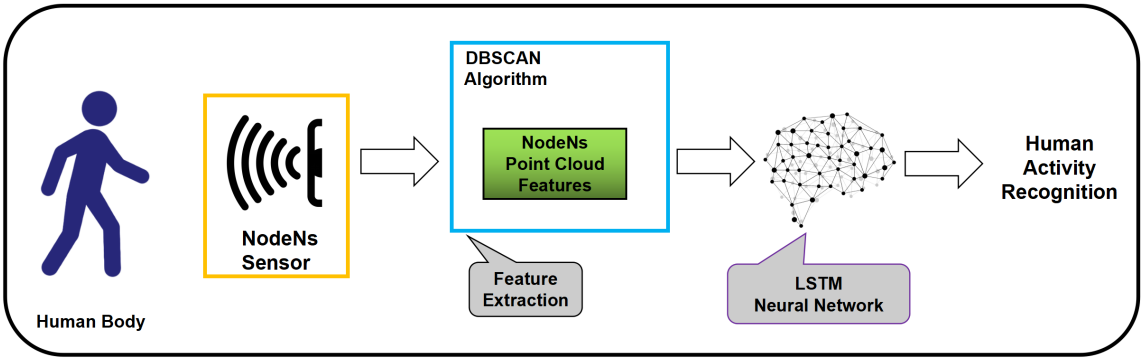


Figure 6.4: Experimental Workflow for HAR Task.

density-connected points, dividing regions with high density into clusters and finding clusters of arbitrary shape in the spatial database of noise.

At this point, DBSCAN is a density-based clustering algorithm, which generally assumes that the tightness of the sample distribution can determine the point cloud categories. For the same category samples, the points between them are closely connected. Therefore, there must be points of the same category around the sample points in this category. So, it is a density-based spatial clustering calculation. There are advantages of anti-noise, no need to specify the number of categories and clustering of arbitrary shapes in spatial data, which is suitable for clustering point cloud data.

The core of the DBSCAN algorithm is to find the most extensive set of densely connected points. It is based on two parameters to describe the tightness of the sample. One is the radius of the neighbourhood, which is used to express the neighbourhood distance threshold of the current point; The second is the number of points, which is used to represent the minimum number of data points in the neighbourhood. The DBSCAN algorithm 6.1 flow is shown in pseudo-code.

The algorithm first traverses the point cloud data for point by point. If the point is not a core point, it is considered as a noise point and ignored, and the noise point may be classified into clusters by the core point later. If it is a core point, create a new cluster, and all neighbourhood point is added to the cluster. The core point in the neighbourhood point is added to the cluster recursively. Following this way, until no point can be added to the cluster, and then start working for new points to establish a new cluster.

In general, if the data set is dense and not convex set, then using the DBSCAN algorithm will be better than other clustering algorithms, such as K-Means. The K-Means clustering algorithm can only deal with spherical clusters, which is solid mass clusters. This is a limitation of the algorithm based on calculating the average distance to cluster. If the data set is not dense, it is not recommended to use DBSCAN for clustering. Compared with the traditional K-Means algorithm, the most significant difference of DBSCAN is that there is no need to input the number of categories K. Meanwhile, DBSCAN's most significant advantage is finding clusters of arbitrary shapes rather than K-Means, which is generally only used for convex sample set

Algorithm 6.1: DBSCAN Algorithm

Input: D is a data set containing n objects, ε is a Radius parameter, M is a Neighborhood density threshold.
Output: A set of of density-based clusters
Require: :
Initial Dataframe to Zero, and then load data D as Dataframe in the algorithm
Setting ε: Radius parameter to 0.4
Setting M: Neighborhood density threshold to 1
Mark all objects as unvisited
while Objects marked unvisited do
Randomly select an unvisited object p:
Mark p as visited
if There are at least M objects in the ε-neighborhood of p **then**
Create a new cluster C, and then add p to C
Let N as p's ε-neighborhood for the set of objects.
for Each point p' in N **do**
if p' is unvisited **then**
Mark p' as unvisited;
if The ε-neighborhood of p' has at least M points **then**
Add these points to N
end if
if p' is still not a member of any cluster **then**
add p' to C:
end if
end if
end for
Output C:
else Mark p' as noise
end if
end while

clustering. Finally, the DBSCAN algorithm can also find outliers while clustering, similar to the BIRCH clustering algorithm.

6.3.2 Human Activity Recognition

As the RNN (Recurrent Neural Network) has the problem of gradient explosion and gradient disappearance in dealing with nonlinear time problems, a temporal recursive neural network LSTM is widely used in time series data. Due to the advantage of the network structure, it can process and predict highly time-dependent and strongly coupled events.

Traditional neural networks are affected by short-term memory. If the time-series data is too long, it will be difficult to transfer information from the earlier time step to the later time step. For time-series data, RNN may miss important information from the start. Because during backpropagation, RNN will face the problem of gradient disappearance. The gradient is used to

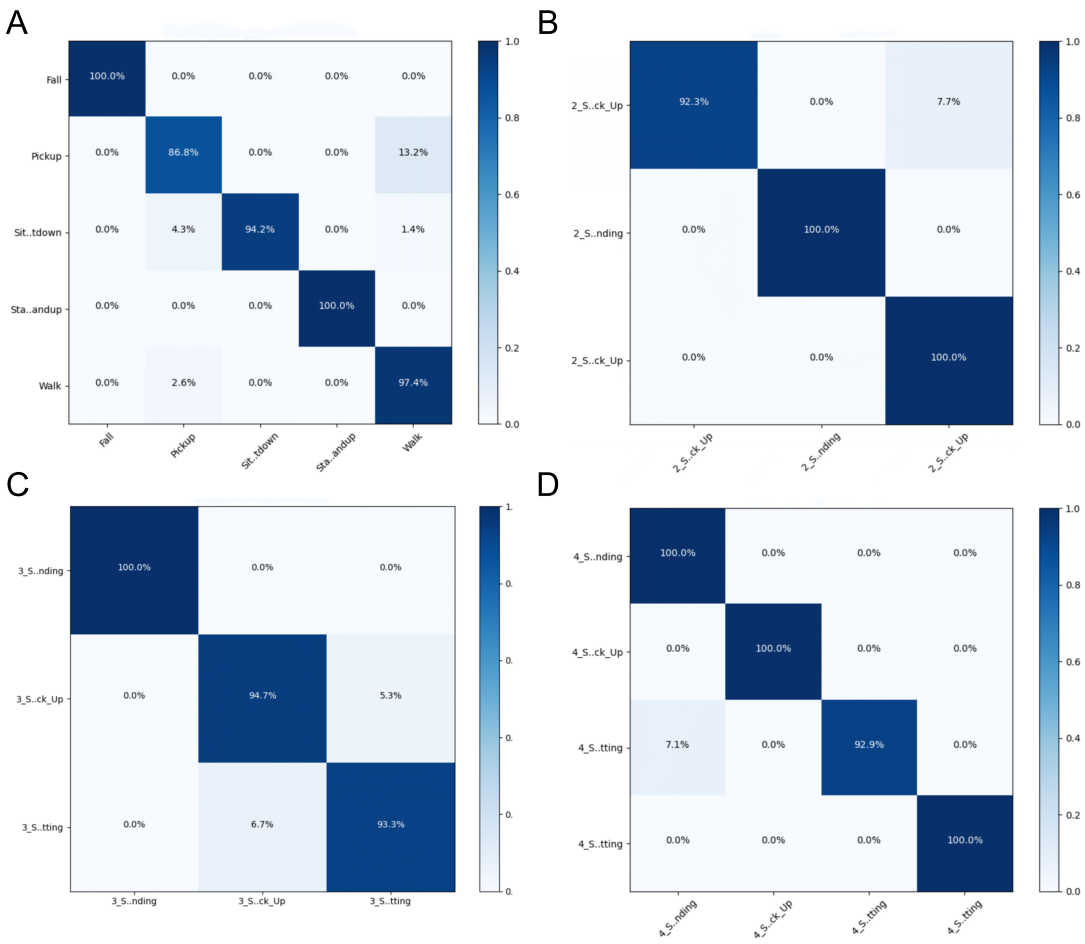


Figure 6.5: Confusion Matrices for single subject to multiple subjects, A is for single subject of 5 activities classification results, B is for two subjects of 3 activities, C is for three subjects of 3 activities and D is for four subjects of 4 activities.

update the weight value of neural networks, and the problem of gradient disappearance is that the gradient decreases as the gradient spreads over time. If the gradient becomes very small, the learning cannot be continued.

Compared with the classical method, the LSTM using the Long Short-Term memory unit can directly connect the feature data of DBSCAN's pre-processing data, input it into the neural network processing, and model the data properly. Diversified with the hidden state of the original RNN, LSTM adds a cell state. For the human being point cloud data after DBSCAN clustering, it is the time series structure, and the data is divided into t parts and fed to LSTM. Finally, the LSTM output the classified result about the HAR.

6.3.3 Evaluation and Results

Evaluate the LSTM algorithm training and testing performance after DBSCAN feature extraction on point cloud data. Figure 6.5 illustrates confusion matrices from single subject to multiple

subjects. Figure 6.5.A is for single subject of 5 activities classification results that is for Fall, Pickup, sit down, stand up and walk, and the accuracy is 95.75%. Figure 6.5.B is for two subjects of 3 activities, which including Sit down + Stand up, Sit down + Pick up and Stand up + Pick up, the accuracy is 97.59%. Figure 6.5.C is for three subjects of 3 activities, it contains Sit down + Stand up + Pick up, Stand up + Pick up + Sit down and Pick up + Sit down + Stand up, the accuracy is for 98.01%. Figure 6.5.D is for four subjects of 4 activities, there is arrangement and combination from sit down, stand up and pick up. The activities sequence are Sit down + Stand up + Pick up +Sit down, Sit down +Sit down + Stand up + Pick up, Pick up + Sit down + Sit down + Stand up and Stand up + Pick up + Sit down + Sit down, which is accuracy for 96.90%. This result reflect the performance evaluation of popular LSTM algorithms in activity recognition task.

6.4 Discussion

In this paper, the focus has been on processing of point cloud data for human movement classification. This is largely motivated by the prominence of HAR datasets, which have been greatly expanding and improving in recent years. Recent work in this area, and corresponding benchmark datasets, have been discussed. Targeting point cloud data in the HAR field, the early ideas were developed by deep learning and extension of image processing concepts. For example, multiple view images [319] or convolutional calculations on 3D voxel grids [320] for point cloud data processing. With processing improvements, convolution operations evolved from 2D convolution kernels to 3D convolution kernels on the neural network. However, running convolutional calculations on point cloud data is a complicated matter. The main difficulty stems from the cat that the point cloud data does not clearly define the point order for performing convolution calculations [321]. Qi *et al* [322] solved this problem by using symmetric functions with constant order of points, and using it to learn the global characteristics of point cloud data. Alternatively, some other methods are proposed to learn local features from convolution [323], and algorithms such as autoencoders are used to remedy the defects [324]. Further research uses data fusion methods, based on point cloud data and multi-view projection data to learn features jointly [325]. Meanwhile, Han demonstrated serializing the point cloud processing and views to analyse data [326]. Finally, some studies directly use the unsupervised learning method [327] to complete the HAR task.

Here, the work has conducted a comparative analysis of some HAR benchmark datasets shown in Table 6.2. For example, Karantonis *et al* [328] designed a benchmark dataset that includes 12 daily activities collected by volunteers' waist accelerometer data.It uses embedded intelligence and a real-time classification system on the wearable unit's board that provides the most significant signal processing. On their benchmark dataset, the HAR task achieved an overall accuracy of 90.8%, the recognition accuracy of the posture direction was 94.1%,

and the possible fall detection accuracy was 95.6%. On the other hand, in Hanai *et al*.'s work [329], they proposed a benchmark dataset based on 1D Haar-like filtering computing. It uses this feature extraction technology that only requires a lower amount of calculation. Moreover, compared with direct raw data processing, the proposed method achieved a HAR accuracy of 93.91%. For Hassan *et al* [237], the point cloud data is further processed by kernel principal component analysis and linear discriminant analysis, making them more robust on HAR feature extraction. Finally, they use the Deep Belief Network (DBN) algorithm to train the extracted features. The accuracy rate is 89.61%, which is better than typical multi-class support vector machines with 82.02% and artificial neural networks of 65.31%. On the benchmark dataset designed by Machado *et al* [330], the unsupervised machine learning method based on the K-Means clustering algorithm helps the human activities achieve the 88.57% recognise accuracy.

Table 6.2: Comparison table with different benchmark dataset and environments

Dataset	Subjects	Activities	Device	Data Format	Accuracy
Our Proposed Work	Single Subject	5	NodeNs Sensor	Point Cloud	95.75%
	Multi Subjects for Two	3			97.59%
	Multi Subjects for Three	3			98.01%
	Multi Subjects for Four	4			96.90%
Karantonis *et al* [328]	Single Subject	6	Wearable Accelerometer Sensor	Acceleration signal	90.8%
Hanai *et al* [329]	Single Subject	5	Wireless Accelerometer Sensor	1D Haar-like biaxial filtering data	93.91%
Hassan *et al* [237]	Single Subject	4(12)	Smartphone	Acceleration and Gyroscopes signal	89.61%
Machado *et al* [330]	Single Subject	7	Accelerometer Sensor	Body Acceleration and Gravitational Acceleration	88.57%

Before the work, there are some real-world human being scanned datasets released [306]. However, most of them are small in scale, which usually with thousands of parameters for data classification tasks are not suitable for training on deep learning networks. Moreover, the performance of these methods is limited when there is a mixture of static and dynamic activity in the aforementioned benchmark dataset. After comprehensive consideration, this work proposes a new point cloud benchmark dataset to help the deep learning algorithm applied by the HAR task for training and increasing recognition accuracy. All human beings are sampled by point cloud data format through accurate scanning of human movements. It is different from existing HAR datasets based on Lidar data collection. Our quantitative evaluations indicate that the proposed benchmark dataset categorising real-world human motion data is challenging. The benchmark's most advanced detection algorithm (DBSCAN+LSTM) achieved more than 95% accuracy on our dataset.

6.5 Summary

With the growing development and technological innovation in the sensor industry, human activity identification has become a popular application in many developing computing fields. The work, on the other hand, has revisited a state-of-the-art point cloud dataset of human movements, thanks to the rise of the point cloud data format. In most situations, however, researchers are unable to obtain an appropriate benchmark dataset to finish the algorithm training. As a result, it advocated using the new NodeNs sensor's human point cloud benchmark dataset to complete the HAR task using a comprehensive analysis of the previous activity recognition dataset. Depending on an advanced detection algorithm, which is based on DBSCAN for clustering NodeNs's point cloud data and LSTM neural network for activity classification. Greater than 95% accuracy is achieved when classifying from a single subject to multiple subjects based on an advanced detection algorithm. The results shows that it is more suitable for deep learning algorithm training than traditional sensor data format and can gather point cloud data more efficiently than Lidar devices. In comparison to the current benchmark dataset, the suggested dataset is significant for realistic issues and has higher detection performance accuracy. Meanwhile, the case studies in the offered benchmark dataset can also provide a better comparison and insight into the benefits and challenges of various classification approaches for HAR processing point clouds. In future research work, hardware resource consumption can be considered, including Memory, CPU, number of sensors, and battery usage. In addition, the most prevalent trade-offs between recognition accuracy, precision, and resource usage should be investigated further.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The main contribution of this thesis is to propose neuromorphic computing for the data fusion framework of multi-sensing signal, and then achieve the algorithm hardware design to complete HAR tasks. It is a software and hardware co-design to achieve health care applications to suitable for a different types of human movement data acquisition devices. It can use neural network hardware to low power and real-time process multiple sensors data in an embedded system. The neurons are stimulated by amplifiers, resistors and other components, which to processing inhibition and activation signals on the neurons. And then, the neural network is constructed by combining neurons. The multiple sensor data fusion results are achieved by designing the front-end feature extraction algorithm, and passing it into the hardware-based neural network to output the recognition result to complete the health care application finally. This proposed hardware and software co-design solution can be used for smart wearable devices to perform local real-time data processing, which does not require additional cloud computing processes for data interaction with sensors. The significance of this thesis is to design a low-power and real-time hardware to simulate neuromorphic computing, which to helping machine learning of data processing to go beyond relying on the software environment. This neuromorphic computing hardware design is a simple, fast and reliable solution. It can build the data fusion neuromorphic computing system flexibly to health care applications. The proposed neuromorphic computing data fusion framework has been successfully validated by the human activities classification results of different human being datasets collected by the CSI laboratory at the University of Glasgow.

This thesis provides a comprehensive overview to review the design process and hardware implementation ideas of HNN neuromorphic computing, and discusses the characteristics of different algorithms (neuromorphic computing, machine learning and neural network) in the field of data processing. As well as the realization of the neuromorphic computing recognition function of case studies and research for related fields of work that gives feasible workflow.

The thesis explores the robustness and computational process of neuromorphic computing for classifying human activities and movements data. Compared with HAR tasks based on traditional machine learning and deep neural network, the use of HNN neuromorphic computing has significant advantages in limited datasets. Because traditional algorithms have a huge demand for training samples, there are high-precision results often require many learning costs. HNN algorithm can use the associative memory function to complete the learning of the samples, and only need one-shot learning to obtain the memory weights of the samples. As a result, it can provide higher precision performance than deep learning on limited datasets.

This thesis starts with a single human activity acquisition device, and researches the wearable IMU sensor to measure the acceleration, angular velocity and magnetic field strength of an individual body part (wrist) to obtain the recognition of human movement by neuromorphic computing. At this point, it only works for a single acquisition device, but the IMU device included has three 3-axis sensors. There are sensor data preprocessed by euler angle and quaternion calculation. Then, the three sensor data are fused by the SVD algorithm to complete the preliminary neuromorphic computing of data fusion for the HAR task. This method is verified on the human motion dataset captured by the IMU sensor, and the recognition accuracy of 96.3% is obtained.

The neuromorphic computing data fusion framework is improved and optimized by further adding human movement acquisition devices, including USRP based on RF sensor and Radar with fixed signal transmitting position. They are different from dynamic physical data of the human body represented by wearable sensors. Radar and USRP data can be represented environment information in 3D space, including relatively static physical data of the object's physical distance, time, and object radial velocity based on the Doppler effect. At this time, the thesis begins to think about how to signal process and make trade-offs with multiple devices. Furthermore, it completes the attention mechanism of neural networks on neuromorphic computing to achieve more efficient learning effects. Finally, a flexible neuromorphic computing framework of multi-device signal data fusion is achieved, and achieve more accurate recognition results than a single device. With the increase of data types and acquisition devices, the algorithm's accuracy has also been improved. Compared with the previous single IMU sensor data, the accuracy of the multi-hardware data fusion framework has reached 98.98%.

In addition to combining IMU, Radar and RF sensors in the framework as the data fusion method, this thesis also introduces the implementation of HNN neuromorphic computing in hardware design in detail. Some simple hardware components such as amplifiers and summers are used to flexible build neurons, and then structure the neural network architecture of HNN. According to the preprocessed feature matrix, the switching of recognition samples can be efficiently completed, which is a step forward compared to the previous inflexible hardware design research of neuromorphic computing. Under the test of hardware simulation, the training and testing of the HNN algorithm through electrical signals on the input and output, it still achieved

an accuracy of 88.9%. Although the results are not as good as the software-only environment, it helps neuromorphic computing obtain hardware acceleration effects. Due to the resource utilization of neuromorphic computing being almost 80% in the software environment, which is difficult to real-time processing the HAR tasks.

Moreover, this thesis also introduces how to work on the benchmark dataset of human activities. Through the study of human activities data, the definition of different types of human movements is completed, and various difficulties of human activities are evaluated. It collected and sorted the point cloud human being data by the new NodeNS sensor, and then packaged it into a standard benchmark HAR dataset.

7.2 Future Work

Although the proposed data fusion framework based on HNN neuromorphic computing shows a significant improvement in classification accuracy compared with traditional machine learning. It is still possible to further improve the system performance. The most critical work in the future is further to advance the completion of the HNN hardware design, and ultimately to achieve the verification and tapeout as the neuromorphic hardware chip.

From the design concept, neuromorphic computing has a hardware-friendly advantage. However, software simulation works with neuromorphic computing at this stage, which takes up most of the hardware resources utilization. In relative terms, traditional artificial intelligence algorithms are more suitable on the software simulation end. The hardware of neuromorphic computing is based on ASIC, and a part of worked on the FPGA; only the hardware as above-mentioned can reflect its advantages. A more comprehensive range of applications by traditional artificial intelligence algorithms that it can work on general-purpose computing chips. Therefore, once the hardware design of neuromorphic computing is completed, its advantages suddenly manifest. There are no requirements for Memory, Cache and BUS. The memory-computing integrated architecture makes the hardware design less resource-intensive, and PPA (Performance, Power, Area) is better. At this point, the modular design of neurons is more flexible to structure the network architecture. The hardware imitates the working principle of the human brain, using neurons and synapses to replace the traditional von-Neumann architecture, enabling the chip to perform asynchronous, parallel, low-power and distributed processing information. It has the ability to perceive, recognize and learn independently.

For neuromorphic computing, HNN as an associative memory algorithm is hardware-friendly and implemented with only simple circuits by amplifier and summer. The algorithm can be updated to the latest Spiking Neural Network(SNN) algorithm in the future. It is recognized as the third generation of neural networks developed after the current MLP-based second-generation neural networks (ANN). However, its hardware design is more complicated and mainly involves neurons' synapses. The signal processing can simulate the changes in synaptic connections be-

tween neurons, and it is a neural model that focuses more on biological interpretation. Other points need to be further optimized for the storage-to-capacity ratio in neuromorphic computing. For example, the current HNN neuromorphic computing adopts the traditional Hebb learning method. As a result, only the memory capacity is approximately 0.14N (N is Neurons) can be achieved. Nevertheless, a higher storage-to-capacity ratio can be achieved by optimizing the learning methods, such as using pseudo inverse matrix, non-monotonic continuous neurons and polynomial interaction function. These allow neuromorphic computing neurons to perform more associative memory functions.

For the data fusion part, the current framework has adopted TopK as an attention mechanism for feature selection. In the case of large-scale feature sets, the TopK calculation can be optimized to speed up feature sorting and help the framework to obtain the required feature matrix more quickly. At this point, the method will be validated in a broader range of acquisition hardware, including more different data types (such as with Doppler data).

Reference

[1] G. V. Angelov, D. P. Nikolakov, I. N. Ruskova, E. E. Gieva, and M. L. Spasova, "Healthcare sensing and monitoring," in *Enhanced Living Environments*. Springer, 2019, pp. 226–262.

[2] J. Gao, Y. Yang, P. Lin, and D. S. Park, "Computer vision in healthcare applications," 2018.

[3] B. Goertzel, "Artificial general intelligence: concept, state of the art, and future prospects," *Journal of Artificial General Intelligence*, vol. 5, no. 1, p. 1, 2014.

[4] B. Rajendran, A. Sebastian, M. Schmuker, N. Srinivasa, and E. Eleftheriou, "Low-power neuromorphic hardware for signal processing applications: A review of architectural and system-level design approaches," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 97–110, 2019.

[5] C. Ahlstrom and K. Kircher, "A generalized method to extract visual time-sharing sequences from naturalistic driving data," *IEEE transactions on intelligent transportation systems*, vol. 18, no. 11, pp. 2929–2938, 2017.

[6] A. Goel, C. Tung, Y.-H. Lu, and G. K. Thiruvathukal, "A survey of methods for low-power deep learning and computer vision," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*. IEEE, 2020, pp. 1–6.

[7] C. Chen, P. Zhang, H. Zhang, J. Dai, Y. Yi, H. Zhang, and Y. Zhang, "Deep learning on computational-resource-limited platforms: A survey," *Mobile Information Systems*, vol. 2020, 2020.

[8] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, "Deep learning vs. traditional computer vision," in *Science and Information Conference*. Springer, 2019, pp. 128–144.

[9] Z. Huang, J. Fan, S. Cheng, S. Yi, X. Wang, and H. Li, "Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion," *IEEE Transactions on Image Processing*, vol. 29, pp. 3429–3441, 2019.

[10] Z. Cao, Z. Li, X. Guo, and G. Wang, "Towards cross-environment human activity recognition based on radar without source data," *IEEE Transactions on Vehicular Technology*, 2021.

[11] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[12] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[13] D. Efnusheva, A. Cholakoska, and A. Tentov, "A survey of different approaches for overcoming the processor-memory bottleneck," *International Journal of Computer Science and Information Technology*, vol. 9, no. 2, pp. 151–163, 2017.

[14] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti, "A high reliability wearable device for elderly fall detection," *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4544–4553, 2015.

[15] Y. Lee, H. Yeh, K.-H. Kim, and O. Choi, "A real-time fall detection system based on the acceleration sensor of smartphone," *International Journal of Engineering Business Management*, vol. 10, p. 1847979017750669, 2018.

[16] Y. Kim and T. Moon, "Human detection and activity classification based on micro-doppler signatures using deep convolutional neural networks," *IEEE geoscience and remote sensing letters*, vol. 13, no. 1, pp. 8–12, 2015.

[17] L. Shi, J. Pei, N. Deng, D. Wang, L. Deng, Y. Wang, Y. Zhang, F. Chen, M. Zhao, S. Song *et al.*, "Development of a neuromorphic computing system," in *2015 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2015, pp. 4–3.

[18] M. Fatahi, "Toward neuromorphic agent."

[19] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, and J. S. Plank, "A survey of neuromorphic computing and neural networks in hardware," *arXiv preprint arXiv:1705.06963*, 2017.

[20] I. K. Schuller, R. Stevens, R. Pino, and M. Pechan, *Neuromorphic Computing–From Materials Research to Systems Architecture Roundtable*, 2015.

[21] D. Monroe, *Neuromorphic computing gets ready for the (really) big time*, 2014.

[22] P. Hasler and L. Akers, "Vlsi neural systems and circuits," in *Ninth Annual International Phoenix Conference on Computers and Communications. 1990 Conference Proceedings*. IEEE, 1990, pp. 31–37.

[23] J.-C. Lee and B. J. Sheu, "Parallel digital image restoration using adaptive vlsi neural chips," in *Proceedings., 1990 IEEE International Conference on Computer Design: VLSI in Computers and Processors*. IEEE, 1990, pp. 126–129.

[24] L. Tarassenko, M. Brownlow, G. Marshall, J. Tombs, and A. Murray, "Real-time autonomous robot navigation using vlsi neural networks," in *Advances in neural information processing systems*, 1991, pp. 422–428.

[25] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[26] S.-C. Liu, T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas, *Event-based neuromorphic systems*. John Wiley & Sons, 2014.

[27] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps)," *IEEE transactions on biomedical circuits and systems*, vol. 12, no. 1, pp. 106–122, 2017.

[28] G. Indiveri and S.-C. Liu, "Memory and information processing in neuromorphic systems," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1379–1397, 2015.

[29] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of physiology*, vol. 117, no. 4, pp. 500–544, 1952.

[30] Q. Ma, M. R. Haider, V. L. Shrestha, and Y. Massoud, "Bursting hodgkin–huxley model-based ultra-low-power neuromimetic silicon neuron," *Analog Integrated Circuits and Signal Processing*, vol. 73, no. 1, pp. 329–337, 2012.

[31] A. Borisyuk, "Morris-lecar model." pp. 1758–1764, 2015.

[32] M. Graupner and N. Brunel, "Calcium-based plasticity model explains sensitivity of synaptic changes to spike pattern, rate, and dendritic location," *Proceedings of the National Academy of Sciences*, vol. 109, no. 10, pp. 3991–3996, 2012.

[33] A. Galves and E. Löcherbach, "Infinite systems of interacting chains with memory of variable length—a stochastic model for biological neural nets," *Journal of Statistical Physics*, vol. 151, no. 5, pp. 896–921, 2013.

[34] F. Grassia, T. Levi, T. Kohno, and S. Saïghi, "Silicon neuron: digital hardware implementation of the quartic model," *Artificial Life and Robotics*, vol. 19, no. 3, pp. 215–219, 2014.

[35] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE transactions on neural networks*, vol. 15, no. 5, pp. 1063–1070, 2004.

[36] M. Hayati, M. Nouri, D. Abbott, and S. Haghiri, "Digital multiplierless realization of two-coupled biological hindmarsh–rose neuron model," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 5, pp. 463–467, 2015.

[37] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Networks*, vol. 111, pp. 47–63, 2019.

[38] J. H. Kotaleski and K. T. Blackwell, "Modelling the molecular mechanisms of synaptic plasticity using systems biology approaches," *Nature Reviews Neuroscience*, vol. 11, no. 4, pp. 239–251, 2010.

[39] E. Neftci, "Stochastic neuromorphic learning machines for weakly labeled data," in *2016 IEEE 34th International Conference on Computer Design (ICCD)*. IEEE, 2016, pp. 670–673.

[40] E. Donati, F. Corradi, C. Stefanini, and G. Indiveri, "A spiking implementation of the lamprey's central pattern generator in neuromorphic vlsi," in *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*. IEEE, 2014, pp. 512–515.

[41] L. O. Chua and L. Yang, "Cellular neural networks: Applications," *IEEE Transactions on circuits and systems*, vol. 35, no. 10, pp. 1273–1290, 1988.

[42] Z. Wang, Y. Ma, F. Cheng, and L. Yang, "Review of pulse-coupled neural networks," *Image and Vision Computing*, vol. 28, no. 1, pp. 5–13, 2010.

[43] J. Secco, M. Farina, D. Demarchi, and F. Corinto, "Memristor cellular automata through belief propagation inspired algorithm," in *2015 International SoC Design Conference (ISOCC)*. IEEE, 2015, pp. 211–212.

[44] Y.-H. Kuo, C.-I. Kao, and J.-J. Chen, "A fuzzy neural network model and its hardware implementation," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 3, pp. 171–183, 1993.

[45] Z. Uykan, "Continuous-time hopfield neural network-based optimized solution to 2-channel allocation problem," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 23, no. 2, pp. 480–490, 2015.

[46] R. T. Cabeza, E. B. Viciedo, A. Prieto-Moreno, and V. M. Vega, "Fault diagnosis with missing data based on hopfield neural networks," in *Mathematical Modeling and Computational Intelligence in Engineering Applications*. Springer, 2016, pp. 37–46.

[47] X. Hu and T. Wang, "Training the hopfield neural network for classification using a stdp-like rule," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 737–744.

[48] G. Indiveri, *Computation in neuromorphic analog VLSI systems*, ser. Neural Nets WIRN Vietri-01. Springer, 2002, pp. 3–20.

[49] F. Blayo and P. Hurat, *A VLSI systolic array dedicated to Hopfield neural network*, ser. VLSI for Artificial Intelligence. Springer, 1989, pp. 255–264.

[50] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–1636, 1990.

[51] J. V. Arthur and K. Boahen, "Learning in silicon: Timing is everything," in *Advances in neural information processing systems*, 2006, pp. 75–82.

[52] I. Bayraktaroğlu, A. S. Öğrenci, G. Dündar, S. Balkır, and E. Alpaydın, "Annsys: an analog neural network synthesis system," *Neural Networks*, vol. 12, no. 2, pp. 325–338, 1999.

[53] D. Brüderle, M. A. Petrovici, B. Vogginger, M. Ehrlich, T. Pfeil, S. Millner, A. Grübl, K. Wendt, E. Müller, and M.-O. Schwartz, "A comprehensive workflow for general-purpose neural modeling with highly configurable neuromorphic hardware systems," *Biological cybernetics*, vol. 104, no. 4-5, pp. 263–296, 2011.

[54] P. Krzysteczko, J. Münchenberger, M. Schäfers, G. Reiss, and A. Thomas, "The memristive magnetic tunnel junction as a nanoscopic synapse-neuron system," *Advanced Materials*, vol. 24, no. 6, pp. 762–766, 2012.

[55] Y. Li, Y. Zhong, L. Xu, J. Zhang, X. Xu, H. Sun, and X. Miao, "Ultrafast synaptic events in a chalcogenide memristor," *Scientific reports*, vol. 3, p. 1619, 2013.

[56] Y. Li, Y. Zhong, J. Zhang, L. Xu, Q. Wang, H. Sun, H. Tong, X. Cheng, and X. Miao, "Activity-dependent synaptic plasticity of a chalcogenide electronic synapse for neuromorphic systems," *Scientific reports*, vol. 4, p. 4906, 2014.

[57] J. Tranchant, E. Janod, B. Corraze, P. Stoliar, M. Rozenberg, M. Besland, and L. Cario, "Control of resistive switching in am4q8 narrow gap mott insulators: A first step towards neuromorphic applications," *physica status solidi (a)*, vol. 212, no. 2, pp. 239–244, 2015.

[58] Y. Chen, G. Liu, C. Wang, W. Zhang, R.-W. Li, and L. Wang, "Polymer memristor for information storage and neuromorphic applications," *Materials Horizons*, vol. 1, no. 5, pp. 489–506, 2014.

[59] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 79, no. 8, pp. 2554–2558, Apr 1982, lR: 20190501; JID: 7505876; 1982/04/01 00:00 [pubmed]; 1982/04/01 00:01 [medline]; 1982/04/01 00:00 [entrez]; ppublish.

[60] X.-S. Zhang, *Neural networks in optimization*. Springer Science & Business Media, 2013, vol. 46.

[61] S. Hu, Y. Liu, Z. Liu, T. Chen, J. Wang, Q. Yu, L. Deng, Y. Yin, and S. Hosaka, "Associative memory realized by a reconfigurable memristive hopfield neural network," *Nature communications*, vol. 6, no. 1, pp. 1–8, 2015.

[62] J. Schmidhuber, "Habilitation thesis: System modeling and optimization," *Page 150 ff demonstrates credit assignment across the equivalent of 1,200 layers in an unfolded RNN*, 1993.

[63] R. Ma, Y. Xie, S. Zhang, and W. Liu, "Convergence of discrete delayed hopfield neural networks," *Computers & Mathematics with Applications*, vol. 57, no. 11-12, pp. 1869–1876, 2009.

[64] C. Ramya, G. Kavitha, and D. K. Shreedhara, "Recalling of images using hopfield neural network model," *arXiv preprint arXiv:1105.0332*, 2011.

[65] A. Adly and S. Abd-El-Hafiz, "Efficient vector hysteresis modeling using rotationally coupled step functions," *Physica B: Condensed Matter*, vol. 407, no. 9, pp. 1350–1353, 2012.

[66] U.-P. Wen, K.-M. Lan, and H.-S. Shih, "A review of hopfield neural networks for solving mathematical programming problems," *European Journal of Operational Research*, vol. 198, no. 3, pp. 675–687, 2009.

[67] S. Kumar and M. P. Singh, "Pattern recall analysis of the hopfield neural network with a genetic algorithm," *Computers & Mathematics with Applications*, vol. 60, no. 4, pp. 1049–1057, 2010.

[68] J. D. Keeler, "Comparison between kanerva's sdm and hopfield-type neural networks," *Cognitive Science*, vol. 12, no. 3, pp. 299–329, 1988.

[69] M. A. Akra, *On the analysis of the Hopfield network: a geometric approach*, 1988.

[70] B. Müller, J. Reinhardt, and M. T. Strickland, *Neural networks: an introduction*. Springer Science & Business Media, 2012.

[71] J. A. Anderson, *An introduction to neural networks*. MIT press, 1995.

[72] H. K. Sulehria and Y. Zhang, "Study on the capacity of hopfield neural networks," *information technology journal*, vol. 7, no. 4, pp. 684–688, 2008.

[73] A. Tankimanova and A. P. James, "Neural network-based analog-to-digital converters," *Memristor and Memristive Neural Networks*, p. 297, 2018.

[74] A. H. Marblestone, G. Wayne, and K. P. Kording, "Toward an integration of deep learning and neuroscience," *Frontiers in computational neuroscience*, vol. 10, p. 94, 2016.

[75] G. A. Elnashar, "Dynamical nonlinear neural networks with perturbations modeling and global robust stability analysis," *International Journal of Computer Applications*, vol. 85, no. 15, 2014.

[76] S. Cocco, R. Monasson, L. Posani, S. Rosay, and J. Tubiana, "Statistical physics and representations in real and artificial neural networks," *Physica A: Statistical Mechanics and its Applications*, vol. 504, pp. 45–76, 2018.

[77] J. J. Hopfield and D. W. Tank, "Computing with neural circuits: a model," *Science (New York, N.Y.)*, vol. 233, no. 4764, pp. 625–633, Aug 8 1986, lR: 20190618; JID: 0404511; 1986/08/08 00:00 [pubmed]; 1986/08/08 00:01 [medline]; 1986/08/08 00:00 [entrez]; ppublish.

[78] D. Tank and J. Hopfield, "Simple'neural'optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit," *IEEE transactions on circuits and systems*, vol. 33, no. 5, pp. 533–541, 1986.

[79] J. J. Hopfield and D. W. Tank, ""neural" computation of decisions in optimization problems," *Biological cybernetics*, vol. 52, no. 3, pp. 141–152, 1985.

[80] A. Tankimanova, A. K. Maan, and A. P. James, "Level-shifted neural encoded analog-to-digital converter," in *2017 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, 2017, pp. 377–380.

[81] M. S. Badie and O. K. Ersoy, "A multistage approach to the hopfield model for bi-level image restoration," *ECE Technical Reports*, p. 143, 1995.

[82] S. Draghici, "Neural networks in analog hardware—design and implementation issues," *International journal of neural systems*, vol. 10, no. 01, pp. 19–42, 2000.

[83] Y. Zhu and Z. Yan, "Computerized tumor boundary detection using a hopfield neural network," *IEEE Transactions on Medical Imaging*, vol. 16, no. 1, pp. 55–67, 1997.

[84] D.-Q. Wang and Z.-J. Li, "Pattern recogition based on hopfield neural network [j]," *Journal of Wuhan Yejin University of Science and Technology*, vol. 4, 2005.

[85] G. Palm, "Neural associative memories and sparse coding," *Neural Networks*, vol. 37, pp. 165–171, 2013.

[86] J. K. Paik and A. K. Katsaggelos, "Image restoration using a modified hopfield network," *IEEE Transactions on Image Processing*, vol. 1, no. 1, pp. 49–63, 1992.

[87] B. Bavarian, "Introduction to neural networks for intelligent control," *IEEE Control Systems Magazine*, vol. 8, no. 2, pp. 3–7, 1988.

[88] S. Matsuda, ""' optimal" hopfield network for combinatorial optimization with linear cost function," *IEEE Transactions on Neural Networks*, vol. 9, no. 6, pp. 1319–1330, 1998.

[89] J.-S. Jang, S.-W. Jung, S.-Y. Lee, and S.-Y. Shin, "Optical implementation of the hopfield model for two-dimensional associative memory," *Optics Letters*, vol. 13, no. 3, pp. 248–250, 1988.

[90] W. Mansour, R. Ayoubi, H. Ziade, W. E. Falou, and R. Velazco, "A fault-tolerant implementation on fpga of a hopfield neural network."

[91] S. S. Yau and H. Fung, "Associative processor architecture—a survey," *ACM Computing Surveys (CSUR)*, vol. 9, no. 1, pp. 3–27, 1977.

[92] M. S. Riazi, M. Samragh, and F. Koushanfar, "Camsure: Secure content-addressable memory for approximate search," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 5s, pp. 1–20, 2017.

[93] L. D. Pyeatt, *Modern assembly language programming with the ARM processor*. Newnes, 2016.

[94] M. Gupta, L. Jin, and N. Homma, *Static and dynamic neural networks: from fundamentals to advanced theory*. John Wiley & Sons, 2004.

[95] Y. P. Singh, A. Khare, and A. Gupta, "Analysis of hopfield autoassociative memory in the character recognition," *International Journal on Computer Science and Engineering*, vol. 2, no. 3, 2010.

[96] X. Zhuang, Y. Huang, and S.-S. Chen, "Better learning for bidirectional associative memory," *Neural Networks*, vol. 6, no. 8, pp. 1131–1146, 1993.

[97] H. Jaeger, "Using conceptors to manage neural long-term memories for temporal patterns," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 387–429, 2017.

[98] B. Ripley, "Linear discriminant analysis," *Pattern recognition and neural networks.Cambridge University Press, Cambridge*, pp. 91–120, 2007.

[99] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[100] F. A. Unal, *Temporal Pattern Matching Using an Artificial Neural Network*, ser. Neural Networks and Pattern Recognition. Elsevier, 1998, pp. 77–104.

[101] A. Meyer-Baese and V. J. Schmid, *Pattern recognition and signal analysis in medical imaging*. Elsevier, 2014.

[102] F. Mastrogiuseppe and S. Ostojic, "A geometrical analysis of global stability in trained feedback networks," *Neural computation*, vol. 31, no. 6, pp. 1139–1182, 2019.

[103] J. F. Pagel and P. Kirshtein, *Machine dreaming and consciousness*. Academic Press, 2017.

[104] G. Serpen, "Hopfield network as static optimizer: Learning the weights and eliminating the guesswork," *Neural Processing Letters*, vol. 27, no. 1, pp. 1–15, 2008.

[105] L. Rong and Q. Junfei, "A new water quality evaluation model based on simplified hopfield neural network," in *2015 34th Chinese Control Conference (CCC)*. IEEE, 2015, pp. 3530–3535.

[106] H. Kappen, "An introduction to stochastic neural networks," in *Handbook of Biological Physics*. Elsevier, 2001, vol. 4, pp. 517–552.

[107] S. P. Muscinelli, W. Gerstner, and J. Brea, "Exponentially long orbits in hopfield neural networks," *Neural computation*, vol. 29, no. 2, pp. 458–484, 2017.

[108] P. Floréen, P. Orponen *et al.*, "On the computational complexity of analyzing hopfield nets," *Complex Systems*, 1989.

[109] H. Yang and Z. Liu, "An optimization routing protocol for fanets," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1–8, 2019.

[110] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 81, no. 10, pp. 3088–3092, May 1984, lR: 20190501; JID: 7505876; 1984/05/01 00:00 [pubmed]; 1984/05/01 00:01 [medline]; 1984/05/01 00:00 [entrez]; ppublish.

[111] C. V. Negoita, *Cybernetics and applied systems*. CRC Press, 1992.

[112] M. S. Ansari, *Voltage-mode neural network for the solution of linear equations*, ser. Non-Linear Feedback Neural Networks. Springer, 2014, pp. 55–104.

[113] R. Rojas, "The hopfield model," in *Neural Networks*. Springer, 1996, pp. 335–369.

[114] S. Xiao, J. Yan, M. Farajtabar, L. Song, X. Yang, and H. Zha, "Joint modeling of event sequence and time series with attentional twin recurrent neural networks," *arXiv preprint arXiv:1703.08524*, 2017.

[115] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.

[116] S. Roa and F. Nino, "Classification of natural language sentences using neural networks." in *Flairs conference*, 2003, pp. 444–449.

[117] P. Zheng, J. Zhang, and W. Tang, "Analysis and design of asymmetric hopfield networks with discrete-time dynamics," *Biological cybernetics*, vol. 103, no. 1, pp. 79–85, 2010.

[118] J. Torres, Ed., *First contact with Deep Learning: Practical introduction with Keras*, watch this space collection - barcelona (book 5) ed. Independently published (July 13, 2018), July 13, 2018 2018, no. Independently published.

[119] R. Reed and R. J. MarksII, *Neural smithing: supervised learning in feedforward artificial neural networks*. Mit Press, 1999.

[120] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," *arXiv preprint arXiv:1710.02913*, 2017.

[121] D.-C. Park, M. A. El-Sharkawi, and R. J. Marks, "An adaptively trained neural network," *IEEE Transactions on Neural Networks*, vol. 2, no. 3, pp. 334–345, 1991.

[122] X. Wu, J. Ghaboussi, and J. G. Jr, "Use of neural networks in detection of structural damage," *Computers & Structures*, vol. 42, no. 4, pp. 649–659, 1992.

[123] M. Riedmiller, "Advanced supervised learning in multi-layer perceptrons—from backpropagation to adaptive learning algorithms," *Computer Standards & Interfaces*, vol. 16, no. 3, pp. 265–278, 1994.

[124] S. Marinai and H. Fujisawa, *Machine learning in document analysis and recognition*. Springer, 2007, vol. 90.

[125] G. ZHANG, *STAR IDENTIFICATION: Methods, Techniques and Algorithms*. SPRINGER, 2019.

[126] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks*, vol. 2, no. 6, pp. 459–473, 1989.

[127] A. Yaman, D. C. Mocanu, G. Iacca, M. Coler, G. Fletcher, and M. Pechenizkiy, "Evolving plasticity for autonomous learning under changing environmental conditions," *arXiv preprint arXiv:1904.01709*, 2019.

[128] J. L. McClelland, "How far can you go with hebbian learning, and when does it lead you astray," *Processes of change in brain and cognitive development: Attention and performance xxi*, vol. 21, pp. 33–69, 2006.

[129] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[130] T. Szandała, "Comparison of different learning algorithms for pattern recognition with hopfield's neural network," *Procedia Computer Science*, vol. 71, pp. 68–75, 2015.

[131] J. Holmes, "Knowledge discovery in biomedical data: theory and methods," *Methods in Biomedical Informatics*, pp. 179–240, 2014.

[132] K. J. Jeffery and I. C. Reid, "Modifiable neuronal connections: an overview for psychiatrists," *The American Journal of Psychiatry*, vol. 154, no. 2, pp. 156–164, Feb 1997, lR: 20191210; JID: 0370512; RF: 35; 1997/02/01 00:00 [pubmed]; 1997/02/01 00:01 [medline]; 1997/02/01 00:00 [entrez]; ppublish.

[133] E. S. Bromberg-Martin, M. Matsumoto, and O. Hikosaka, "Dopamine in motivational control: rewarding, aversive, and alerting," *Neuron*, vol. 68, no. 5, pp. 815–834, 2010.

[134] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[135] C. Grosan and A. Abraham, *Intelligent systems*. Springer, 2011.

[136] C. Fyfe, "Artificial neural networks and information theory," *Department of Computing and Information System, The university of Paisley*, 2000.

[137] R. Rojas, *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.

[138] H. A. Hama, "Hebb rule method in neural network for pattern association," Ph.D. dissertation, Eastern Mediterranean University (EMU)-Doğu Akdeniz Üniversitesi (DAÜ), 2014.

[139] Y. L. Karpov, L. Karpov, and Y. G. Smetanin, "Some aspects of associative memory construction based on a hopfield network," *Programming and Computer Software*, vol. 46, no. 5, pp. 305–311, 2020.

[140] M. Zhang, Z. Zhou, L. Li, Z. Liu, M. Yang, and Y. Feng, "Hpgan: Sequence search with generative adversarial networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[141] Y. Lou, Z. Ren, Y. Zhao, and Y. Song, "Using auto-associative neural networks for signal recognition technology on sky screen," in *Proceedings of International Conference on Soft Computing Techniques and Engineering Application*. Springer, 2014, pp. 71–79.

[142] H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, L. Gruber, M. Holzleitner, M. Pavlović, G. K. Sandve, V. Greiff *et al.*, "Hopfield networks is all you need," *arXiv preprint arXiv:2008.02217*, 2020.

[143] R. McEliece, E. Posner, E. Rodemich, and S. Venkatesh, "The capacity of the hopfield associative memory," *IEEE Transactions on Information Theory*, vol. 33, no. 4, pp. 461–482, 1987.

[144] V. Folli, M. Leonetti, and G. Ruocco, "On the maximum storage capacity of the hopfield model," *Frontiers in computational neuroscience*, vol. 10, p. 144, 2017.

[145] D. J. Amit, H. Gutfreund, and H. Sompolinsky, "Storing infinite numbers of patterns in a spin-glass model of neural networks," *Physical Review Letters*, vol. 55, no. 14, p. 1530, 1985.

[146] S. Seung, "The hopfield model," *Introduction to Computational Neuroscience*, pp. 1–6, 2004.

[147] S. Bartunov, J. W. Rae, S. Osindero, and T. P. Lillicrap, "Meta-learning deep energy-based memory models," *arXiv preprint arXiv:1910.02720*, 2019.

[148] A. Storkey, "Increasing the capacity of a hopfield network without sacrificing functionality," in *International Conference on Artificial Neural Networks*. Springer, 1997, pp. 451–456.

[149] S. N. Srihari, "Recognition of handwritten and machine-printed text for postal address interpretation," *Pattern Recognition Letters*, vol. 14, no. 4, pp. 291–302, 1993.

[150] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate-recognition algorithm for intelligent transportation system applications," *IEEE Transactions on Intelligent transportation systems*, vol. 7, no. 3, pp. 377–392, 2006.

[151] A. Asif, S. A. Hannan, Y. Perwej, and M. A. Vithalrao, "An overview and applications of optical character recognition," *Int.J.Adv.Res.Sci.Eng*, vol. 3, no. 7, 2014.

[152] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen, and P. Shi, "An algorithm for license plate recognition applied to intelligent transportation system," *IEEE Transactions on intelligent transportation systems*, vol. 12, no. 3, pp. 830–845, 2011.

[153] J. Chang, Y. Zhou, and Z. Liu, "Limited top-down influence from recognition to same-different matching of chinese characters," *PloS one*, vol. 11, no. 6, p. e0156517, Jun 3 2016, lR: 20181113; JID: 101285081; 2015/12/23 00:00 [received]; 2016/05/16 00:00 [accepted]; 2016/06/04 06:00 [entrez]; 2016/06/04 06:00 [pubmed]; 2017/08/19 06:00 [medline]; epublish.

[154] R. Parisi, E. D. D. Claudio, G. Lucarelli, and G. Orlandi, "Car plate recognition by neural networks and image processing," in *ISCAS'98. Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (Cat. No. 98CH36187)*, vol. 3. IEEE, 1998, pp. 195–198.

[155] M. Padmanaban and E. Yfantis, "Handwritten character recognition using conditional probabilities."

[156] J. Nijhuis, M. T. Brugge, K. Helmholt, J. Pluim, L. Spaanenburg, R. Venema, and M. Westenberg, "Car license plate recognition with neural networks and fuzzy logic," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 5. IEEE, 1995, pp. 2232–2236.

[157] P. Wittek, *Quantum machine learning: what quantum computing means to data mining*. Academic Press, 2014.

[158] H. Chu, W. Lu, and L. Zhang, "Application of artificial neural network in environmental water quality assessment," 2013.

[159] X. Yang, L. Zhao, G. M. Megson, and D. J. Evans, "A system-level fault diagnosis algorithm based on preprocessing and parallel hopfield neural network," in *4th IEEE Workshop on RTL and High Level Testing*, 2003, pp. 189–196.

[160] G. Di Modica, S. Gulino, and O. Tomarchio, "Iot fault management in cloud/fog environments," in *Proceedings of the 9th International Conference on the Internet of Things*, 2019, pp. 1–4.

[161] H. Yang, B. Wang, Q. Yao, A. Yu, and J. Zhang, "Efficient hybrid multi-faults location based on hopfield neural network in 5g coexisting radio and optical wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1218–1228, 2019.

[162] B. Wang, H. Yang, Q. Yao, A. Yu, T. Hong, J. Zhang, M. Kadoch, and M. Cheriet, "Hopfield neural network-based fault location in wireless and optical networks for smart city iot," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2019, pp. 1696–1701.

[163] Y. Watanabe, N. Mizuguchi, and Y. Fujii, "Solving optimization problems by using a hopfield neural network and genetic algorithm combination," *Systems and Computers in Japan*, vol. 29, no. 10, pp. 68–74, 1998.

[164] L. Cantini and M. Caselle, "Hope4genes: a hopfield-like class prediction algorithm for transcriptomic data," *Scientific reports*, vol. 9, no. 1, pp. 1–9, 2019.

[165] T. Washizawa, "Application of hopfield network to saccades," *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 995–997, 1993.

[166] N. Soni, E. K. Sharma, and A. Kapoor, "Application of hopfield neural network for facial image recognition."

[167] N. Soni, N. Singh, A. Kapoor, and E. K. Sharma, *Low-Resolution Image Recognition Using Cloud Hopfield Neural Network*, ser. Progress in Advanced Computing and Intelligent Engineering. Springer, 2018, pp. 39–46.

[168] K. S. Ray and D. D. Majumder, "Application of hopfield neural networks and canonical perspectives to recognize and locate partially occluded 3-d objects," *Pattern Recognition Letters*, vol. 15, no. 8, pp. 815–824, 1994.

[169] D. O. Albanez, S. F. da Silva, M. A. Batista, and C. A. Z. Barcelos, "Images segmentation using a modified hopfield artificial neural network," *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, vol. 6, no. 1, 2018.

[170] F. Sabahi, M. O. Ahmad, and M. Swamy, "Hopfield network-based image retrieval using re-ranking and voting," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2017, pp. 1–4.

[171] R. A. Silver, "Neuronal arithmetic," *Nature Reviews Neuroscience*, vol. 11, no. 7, pp. 474–489, 2010.

[172] E. Blem, J. Menon, and K. Sankaralingam, "Power struggles: Revisiting the risc vs. cisc debate on contemporary arm and x86 architectures," in *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2013, pp. 1–12.

[173] D. Thain, *Introduction to Compilers and Language Design*. Lulu. com, 2019.

[174] L. Wang, S.-R. Lu, and J. Wen, "Recent advances on neuromorphic systems using phase-change materials," *Nanoscale research letters*, vol. 12, no. 1, p. 347, 2017.

[175] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent advances in physical reservoir computing: A review," *Neural Networks*, 2019.

[176] V. Saxena, X. Wu, I. Srivastava, and K. Zhu, "Towards neuromorphic learning machines using emerging memory devices with brain-like energy efficiency," *Journal of Low Power Electronics and Applications*, vol. 8, no. 4, p. 34, 2018.

[177] P. Date, C. D. Carothers, J. A. Hendler, and M. Magdon-Ismail, "Efficient classification of supercomputer failures using neuromorphic computing," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 242–249.

[178] D. Ielmini and S. Ambrogio, "Emerging neuromorphic devices," *Nanotechnology*, vol. 31, no. 9, p. 092001, 2019.

[179] S. Furber, "Large-scale neuromorphic computing systems," *Journal of neural engineering*, vol. 13, no. 5, p. 051001, 2016.

[180] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, 2019.

[181] X. Yan, Q. Zhao, A. P. Chen, J. Zhao, Z. Zhou, J. Wang, H. Wang, L. Zhang, X. Li, and Z. Xiao, "Vacancy-induced synaptic behavior in 2d ws2 nanosheet–based memristor for low-power neuromorphic computing," *Small*, vol. 15, no. 24, p. 1901423, 2019.

[182] M. Grossi, "A sensor-centric survey on the development of smartphone measurement and sensing systems," *Measurement*, vol. 135, pp. 572–592, 2019.

[183] J. J. Guiry, P. van de Ven, J. Nelson, L. Warmerdam, and H. Riper, "Activity recognition with smartphone support," *Medical engineering & physics*, vol. 36, no. 6, pp. 670–675, 2014.

[184] L. Liu, S. A. Shah, G. Zhao, and X. Yang, "Respiration symptoms monitoring in body area networks," *Applied Sciences*, vol. 8, no. 4, p. 568, 2018.

[185] X. Yang, S. A. Shah, A. Ren, N. Zhao, J. Zhao, F. Hu, Z. Zhang, W. Zhao, M. U. Rehman, and A. Alomainy, "Monitoring of patients suffering from rem sleep behavior disorder," *IEEE Journal of Electromagnetics, RF and Microwaves in Medicine and Biology*, vol. 2, no. 2, pp. 138–143, 2018.

[186] S. Zhang, Z. Wei, J. Nie, L. Huang, S. Wang, and Z. Li, "A review on human activity recognition using vision-based method," *Journal of healthcare engineering*, vol. 2017, 2017.

[187] R. Schmucker, C. Zhou, and M. Veloso, "Multimodal movement activity recognition using a robot's proprioceptive sensors," in *Robot World Cup*. Springer, 2018, pp. 299–310.

[188] H. Banaee, M. U. Ahmed, and A. Loutfi, "Data mining for wearable sensors in health monitoring systems: a review of recent trends and challenges," *Sensors*, vol. 13, no. 12, pp. 17 472–17 500, 2013.

[189] Z. Hussain, M. Sheng, and W. E. Zhang, "Different approaches for human activity recognition: A survey," *arXiv preprint arXiv:1906.05074*, 2019.

[190] O. Ojetola, "Detection of human falls using wearable sensors," Ph.D. dissertation, Coventry University, 2013.

[191] P. Kumar, J. Verma, and S. Prasad, "Hand data glove: a wearable real-time device for human-computer interaction," *International Journal of Advanced Science and Technology*, vol. 43, 2012.

[192] O. Lambercy, S. Maggioni, L. Lünenburger, R. Gassert, and M. Bolliger, "Robotic and wearable sensor technologies for measurements/clinical assessments," in *Neurorehabilitation technology*. Springer, 2016, pp. 183–207.

[193] S. Zhu, J. Xu, H. Guo, Q. Liu, S. Wu, and H. Wang, "Indoor human activity recognition based on ambient radar with signal processing and machine learning," in *2018 IEEE international conference on communications (ICC)*. IEEE, 2018, pp. 1–6.

[194] S.-R. Ke, H. L. U. Thuc, Y.-J. Lee, J.-N. Hwang, J.-H. Yoo, and K.-H. Choi, "A review on video-based human activity recognition," *Computers*, vol. 2, no. 2, pp. 88–131, 2013.

[195] S. A. Shah, N. Zhao, A. Ren, Z. Zhang, X. Yang, J. Yang, and W. Zhao, "Posture recognition to prevent bedsores for multiple patients using leaking coaxial cable," *IEEE Access*, vol. 4, pp. 8065–8072, 2016.

[196] T. Suzuki, H. Tanaka, S. Minami, H. Yamada, and T. Miyata, "Wearable wireless vital monitoring technology for smart health care," in *2013 7th International Symposium on Medical Information and Communication Technology (ISMICT)*. IEEE, 2013, pp. 1–4.

[197] S. Shelke and B. Aksanli, "Static and dynamic activity detection with ambient sensors in smart spaces," *Sensors*, vol. 19, no. 4, p. 804, 2019.

[198] M. Munoz-Organero, "Outlier detection in wearable sensor data for human activity recognition (har) based on drnns," *IEEE Access*, vol. 7, pp. 74 422–74 436, 2019.

[199] A. Ayman, O. Attalah, and H. Shaban, "An efficient human activity recognition framework based on wearable imu wrist sensors," in *2019 IEEE International Conference on Imaging Systems and Techniques (IST)*. IEEE, 2019, pp. 1–5.

[200] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. d. R. Millán, and D. Roggen, "The opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 2033–2042, 2013.

[201] T. T. Um, V. Babakeshizadeh, and D. Kulić, "Exercise motion classification from large-scale wearable sensor data using convolutional neural networks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2385–2390.

[202] I. Sucholutsky and M. Schonlau, "'less than one'-shot learning: Learning n classes from m< n samples," *arXiv preprint arXiv:2009.08449*, 2020.

[203] Z. Yu, A. M. Abdulghani, A. Zahid, H. Heidari, M. A. Imran, and Q. H. Abbasi, "An overview of neuromorphic computing for artificial intelligence enabled hardware-based hopfield neural network," *IEEE Access*, vol. 8, pp. 67 085–67 099, 2020.

[204] Z. Yu, A. Zahid, S. Ansari, H. Abbas, A. M. Abdulghani, H. Heidari, M. A. Imran, and Q. H. Abbasi, "Hardware-based hopfield neuromorphic computing for fall detection," *Sensors*, vol. 20, no. 24, p. 7226, 2020.

[205] H. Li, A. Shrestha, H. Heidari, J. Le Kernec, and F. Fioranelli, "Magnetic and radar sensing for multimodal remote health monitoring," *IEEE Sensors Journal*, vol. 19, no. 20, pp. 8979–8989, 2018.

[206] ——, "A multisensory approach for remote health monitoring of older people," *IEEE Journal of Electromagnetics, RF and Microwaves in Medicine and Biology*, vol. 2, no. 2, pp. 102–108, 2018.

[207] H. Li, A. Shrestha, F. Fioranelli, J. Le Kernec, H. Heidari, M. Pepa, E. Cippitelli, E. Gambi, and S. Spinsante, "Multisensor data fusion for human activities classification and fall detection," in *2017 IEEE SENSORS*. IEEE, 2017, pp. 1–3.

[208] J. L. R. Ortiz, *Smartphone-based human activity recognition*. Springer, 2015.

[209] J. B. Kuipers, *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*. Princeton university press, 1999.

[210] S. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," *Report x-io and University of Bristol (UK)*, vol. 25, pp. 113–118, 2010.

[211] Y.-B. Jia, "Quaternions," *Course Com S*, vol. 477, p. 577, 2019.

[212] M. Ben-Ari, "A tutorial on euler angles and quaternions," *Weizmann Institute of Science, Israel*, 2014.

[213] A. Janota, V. Šimák, D. Nemec, and J. Hrbček, "Improving the precision and speed of euler angles computation from low-cost rotation sensor data," *Sensors*, vol. 15, no. 3, pp. 7016–7039, 2015.

[214] V. Klema and A. Laub, "The singular value decomposition: Its computation and some applications," *IEEE Transactions on automatic control*, vol. 25, no. 2, pp. 164–176, 1980.

[215] I. Thakur and H. Saini, "A review: Analysis of svd based image fusion methods," *International Journal of Engineering Research and Technology, 3 (5)*, pp. 802–804, 2014.

[216] Z. Drmac, "Svd of hankel matrices in vandermonde-cauchy product form," *Electron. Trans. Numer. Anal*, vol. 44, pp. 593–623, 2015.

[217] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Asian conference on computer vision*. Springer, 2010, pp. 709–720.

[218] P. Dangeti, *Statistics for machine learning*. Packt Publishing Ltd, 2017.

[219] G. Spiros, V, S. K. Tasoulis, G. I. Mallis, A. G. Vrahatis, V. P. Plagianakos, and I. G. Maglogiannis, "Change detection and convolution neural networks for fall recognition," *Neural Computing and Applications*, vol. 32, no. 23, pp. 17 245–17 258, 2020.

[220] S. Ashry, T. Ogawa, and W. Gomaa, "Charm-deep: Continuous human activity recognition model based on deep neural network using imu sensors of smartwatch," *IEEE Sensors Journal*, vol. 20, no. 15, pp. 8757–8770, 2020.

[221] W. Taylor, S. A. Shah, K. Dashtipour, A. Zahid, Q. H. Abbasi, and M. A. Imran, "An intelligent non-invasive real-time human activity recognition system for next-generation healthcare," *Sensors*, vol. 20, no. 9, p. 2653, 2020.

[222] G. Fortino, S. Galzarano, R. Gravina, and W. Li, "A framework for collaborative computing and multi-sensor data fusion in body sensor networks," *Information Fusion*, vol. 22, pp. 50–70, 2015.

[223] X. Zhao, Q. Luo, and B. Han, "Survey on robot multi-sensor information fusion technology," in *2008 7th World Congress on Intelligent Control and Automation*. IEEE, 2008, pp. 5019–5023.

[224] Y. Dongyong and Y. Yuzo, "Multi-sensor data fusion and its application to industrial control," in *SICE 2000. Proceedings of the 39th SICE Annual Conference. International Session Papers (IEEE Cat. No. 00TH8545)*. IEEE, 2000, pp. 215–220.

[225] C. Qian, X. Li, J. Zhu, T. Liu, R. Li, B. Li, M. Hu, Y. Xin, and Y. Xu, "A bionic manipulator based on multi-sensor data fusion," *Integrated Ferroelectrics*, vol. 192, no. 1, pp. 10–15, 2018.

[226] J. Kim, D. S. Han, and B. Senouci, "Radar and vision sensor fusion for object detection in autonomous vehicle surroundings," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2018, pp. 76–78.

[227] Q. Liu, "Intelligent environmental monitoring system based on multi-sensor data technology," *International Journal of Ambient Computing and Intelligence (IJACI)*, vol. 11, no. 4, pp. 57–71, 2020.

[228] B. Gao, G. Hu, S. Gao, Y. Zhong, and C. Gu, "Multi-sensor optimal data fusion for ins/gnss/cns integration based on unscented kalman filter," *International Journal of Control, Automation, and Systems*, vol. 16, no. 1, pp. 129–140, 2018.

[229] M. Maimaitijiang, A. Ghulam, P. Sidike, S. Hartling, M. Maimaitiyiming, K. Peterson, E. Shavers, J. Fishman, J. Peterson, S. Kadam *et al.*, "Unmanned aerial system (uas)-based phenotyping of soybean using multi-sensor data fusion and extreme learning machine," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 134, pp. 43–58, 2017.

[230] W. Jiao, L. Wang, and M. F. McCabe, "Multi-sensor remote sensing for drought characterization: current status, opportunities and a roadmap for the future," *Remote Sensing of Environment*, vol. 256, p. 112313, 2021.

[231] L. Kong, X. Peng, Y. Chen, P. Wang, and M. Xu, "Multi-sensor measurement and data fusion technology for manufacturing process monitoring: a literature review," *International Journal of Extreme Manufacturing*, vol. 2, no. 2, p. 022001, 2020.

[232] F. Kong, Y. Zhou, and G. Chen, "Multimedia data fusion method based on wireless sensor network in intelligent transportation system," *Multimedia Tools and Applications*, vol. 79, no. 47, pp. 35 195–35 207, 2020.

[233] P. Zhang, T. Li, G. Wang, C. Luo, H. Chen, J. Zhang, D. Wang, and Z. Yu, "Multi-source information fusion based on rough set theory: A review," *Information Fusion*, 2020.

[234] Y. Sun, C. Li, G. Li, G. Jiang, D. Jiang, H. Liu, Z. Zheng, and W. Shu, "Gesture recognition based on kinect and semg signal fusion," *Mobile Networks and Applications*, vol. 23, no. 4, pp. 797–805, 2018.

[235] M. Muzammal, R. Talat, A. H. Sodhro, and S. Pirbhulal, "A multi-sensor data fusion enabled ensemble approach for medical data from body sensor networks," *Information Fusion*, vol. 53, pp. 155–164, 2020.

[236] Z. Yu, P. Machado, A. Zahid, A. M. Abdulghani, K. Dashtipour, H. Heidari, M. A. Imran, and Q. H. Abbasi, "Energy and performance trade-off optimization in heterogeneous computing via reinforcement learning," *Electronics*, vol. 9, no. 11, p. 1812, 2020.

[237] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," *Future Generation Computer Systems*, vol. 81, pp. 307–313, 2018.

[238] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020.

[239] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of big data*, vol. 2, no. 1, pp. 1–21, 2015.

[240] M. Z. Alom, T. Josue, M. N. Rahman, W. Mitchell, C. Yakopcic, and T. M. Taha, "Deep versus wide convolutional neural networks for object recognition on neuromorphic system," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[241] B. M. Eskofier, S. I. Lee, J.-F. Daneault, F. N. Golabchi, G. Ferreira-Carvalho, G. Vergara-Diaz, S. Sapienza, G. Costante, J. Klucken, T. Kautz *et al.*, "Recent machine learning advancements in sensor-based mobility analysis: Deep learning for parkinson's disease assessment," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 655–658.

[242] R. Tasooji, N. Buckingham, D. Gračanin, and R. B. Knapp, "An approach to analysis of physiological responses to stimulus," in *International Conference on Human-Computer Interaction*. Springer, 2019, pp. 492–509.

[243] O. Holland, H. Bogucka, and A. Medeisis, *Opportunistic spectrum sharing and white space access: The practical reality*. John Wiley & Sons, 2015.

[244] G. De Leonardis, S. Rosati, G. Balestra, V. Agostini, E. Panero, L. Gastaldi, and M. Knaflitz, "Human activity recognition by wearable sensors: Comparison of different classifiers for real-time applications," in *2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. IEEE, 2018, pp. 1–6.

[245] H. Guo, N. Zhang, S. Wu, and Q. Yang, "Deep learning driven wireless real-time human activity recognition," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[246] A. Merwaday, N. Rupasinghe, I. Güvenç, W. Saad, and M. Yuksel, "Usrp-based indoor channel sounding for d2d and multi-hop communications," in *WAMICON 2014*. IEEE, 2014, pp. 1–6.

[247] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[248] Z. Noshad, N. Javaid, T. Saba, Z. Wadud, M. Q. Saleem, M. E. Alzahrani, and O. E. Sheta, "Fault detection in wireless sensor networks through the random forest classifier," *Sensors*, vol. 19, no. 7, p. 1568, 2019.

[249] J. Chen, Y. Sun, and S. Sun, "Improving human activity recognition performance by data fusion and feature engineering," *Sensors*, vol. 21, no. 3, p. 692, 2021.

[250] S. Chung, J. Lim, K. J. Noh, G. Kim, and H. Jeong, "Sensor data acquisition and multimodal sensor fusion for human activity recognition using deep learning," *Sensors*, vol. 19, no. 7, p. 1716, 2019.

[251] A. F. Calvo, G. A. Holguin, and H. Medeiros, "Human activity recognition using multimodal data fusion," in *Iberoamerican Congress on Pattern Recognition*. Springer, 2018, pp. 946–953.

[252] H. Zou, J. Yang, H. Prasanna Das, H. Liu, Y. Zhou, and C. J. Spanos, "Wifi and vision multimodal learning for accurate and robust device-free human activity recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[253] A. Meyer-Baese, A. Meyer-Baese, and V. J. Schmid, *Pattern Recognition and Signal Analysis in Medical Imaging*. Academic Press, 2004.

[254] R. Kavitha and E. Kannan, "An efficient framework for heart disease classification using feature extraction and feature selection technique in data mining," in *2016 international conference on emerging trends in engineering, technology and science (icetets)*. IEEE, 2016, pp. 1–5.

[255] C.-U. Shina and J.-W. Chab, "Top-k attention mechanism for complex dialogue system," 2019.

[256] X. Zhang, M. Fan, D. Wang, P. Zhou, and D. Tao, "Top-k feature selection framework using robust 0-1 integer programming," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[257] L. Jia, "A hybrid feature selection method for software defect prediction," in *IOP Conference Series: Materials Science and Engineering*, vol. 394, no. 3. IOP Publishing, 2018, p. 032035.

[258] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[259] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[260] L. Wang, Y. Zhang, and J. Feng, "On the euclidean distance of images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1334–1339, 2005.

[261] S. S. Bangaru, C. Wang, S. A. Busam, and F. Aghazadeh, "Ann-based automated scaffold builder activity recognition through wearable emg and imu sensors," *Automation in Construction*, vol. 126, p. 103653, 2021.

[262] G. V. Angelov, D. P. Nikolakov, I. N. Ruskova, E. E. Gieva, and M. L. Spasova, *Healthcare Sensing and Monitoring*, ser. Enhanced Living Environments. Springer, 2019, pp. 226–262.

[263] I. Ganchev, N. M. Garcia, C. Dobre, C. X. Mavromoustakis, and R. Goleva, *Enhanced Living Environments: Algorithms, Architectures, Platforms, and Systems*. Springer, 2019, vol. 11369.

[264] C. Chatzaki, M. Pediaditis, G. Vavoulas, and M. Tsiknakis, "Human daily activity and fall recognition using a smartphone's acceleration sensor," in *International Conference on Information and Communication Technologies for Ageing Well and e-Health*. Springer, 2016, pp. 100–118.

[265] T. Haak, H. Hanaire, R. Ajjan, N. Hermanns, J.-P. Riveline, and G. Rayman, "Use of flash glucose-sensing technology for 12 months as a replacement for blood glucose monitoring in insulin-treated type 2 diabetes," *Diabetes Therapy*, vol. 8, no. 3, pp. 573–586, 2017.

[266] D. Oletic and V. Bilas, "Energy-efficient respiratory sounds sensing for personal mobile asthma monitoring," *Ieee sensors journal*, vol. 16, no. 23, pp. 8295–8303, 2016.

[267] J. Gao, Y. Yang, P. Lin, and D. S. Park, "Computer vision in healthcare applications," *Journal of healthcare engineering*, vol. 2018, 2018.

[268] P. Gembaczka, B. Heidemann, B. Bennertz, W. Groeting, T. Norgall, and K. Seidl, "Combination of sensor-embedded and secure server-distributed artificial intelligence for healthcare applications," *Current Directions in Biomedical Engineering*, vol. 5, no. 1, pp. 29–32, 2019.

[269] K. W. Ching and M. M. Singh, "Wearable technology devices security and privacy vulnerability analysis," *International Journal of Network Security & Its Applications*, vol. 8, no. 3, pp. 19–30, 2016.

[270] T. Kameda, M. Kimura, and Y. Nakashima, "Neuromorphic hardware using simplified elements and thin-film semiconductor devices as synapse elements-simulation of hopfield and cellular neural network," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 769–776.

[271] B. Yan, A. M. Mahmoud, J. J. Yang, Q. Wu, Y. Chen, and H. H. Li, "A neuromorphic asic design using one-selector-one-memristor crossbar," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2016, pp. 1390–1393.

[272] G. Indiveri and S.-C. Liu, "Memory and information processing in neuromorphic systems," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1379–1397, 2015.

[273] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri, "Neuromorphic electronic circuits for building autonomous cognitive systems," *Proceedings of the IEEE*, vol. 102, no. 9, pp. 1367–1388, 2014.

[274] J. Bueno, S. Maktoobi, M. Jacquot, I. Fischer, L. Lager, S. Reitzenstein, and D. Brunner, "Towards photonic networks of micropillar lasers for neuromorphic computing," 2018.

[275] C. Mayr, J. Partzsch, M. Noack, S. Hänzsche, S. Scholze, S. Höppner, G. Ellguth, and R. Schüffny, "A biological-realtime neuromorphic system in 28 nm cmos using low-leakage switched capacitor circuits," *IEEE transactions on biomedical circuits and systems*, vol. 10, no. 1, pp. 243–254, 2015.

[276] L. Rong and Q. Junfei, "A new water quality evaluation model based on simplified hopfield neural network," in *2015 34th Chinese Control Conference (CCC)*. IEEE, 2015, pp. 3530–3535.

[277] L. Cantini and M. Caselle, "Hope4genes: a hopfield-like class prediction algorithm for transcriptomic data," *Scientific reports*, vol. 9, no. 1, pp. 1–9, 2019.

[278] K. S. Ray and D. D. Majumder, "Application of hopfield neural networks and canonical perspectives to recognize and locate partially occluded 3-d objects," *Pattern Recognition Letters*, vol. 15, no. 8, pp. 815–824, 1994.

[279] A. J. M. López, J. R. P. Suarez, and D. T. Varela, "Execution and analysis of classic neural network algorithms when they are implemented in embedded systems," in *MATEC Web of Conferences*, vol. 292. EDP Sciences, 2019, p. 01012.

[280] P. Boriskov, "Iot-oriented design of an associative memory based on impulsive hopfield neural network with rate coding of lif oscillators," *Electronics*, vol. 9, no. 9, p. 1468, 2020.

[281] L. Shi, J. Pei, N. Deng, D. Wang, L. Deng, Y. Wang, Y. Zhang, F. Chen, M. Zhao, and S. Song, "Development of a neuromorphic computing system," in *2015 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2015, pp. 4.3. 1–4.3. 4.

[282] M. Fatahi, "Toward neuromorphic agent."

[283] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 79, no. 8, pp. 2554–2558, Apr 1982, lR: 20190501; JID: 7505876; 1982/04/01 00:00 [pubmed]; 1982/04/01 00:01 [medline]; 1982/04/01 00:00 [entrez]; ppublish.

[284] R. Ma, Y. Xie, S. Zhang, and W. Liu, "Convergence of discrete delayed hopfield neural networks," *Computers & Mathematics with Applications*, vol. 57, no. 11-12, pp. 1869–1876, 2009.

[285] G. A. Elnashar, "Dynamical nonlinear neural networks with perturbations modeling and global robust stability analysis," *International Journal of Computer Applications*, vol. 85, no. 15, 2014.

[286] P. M. Talaván and J. Yáñez, "A continuous hopfield network equilibrium points algorithm," *Computers & Operations Research*, vol. 32, no. 8, pp. 2179–2196, 2005.

[287] S. Kumar and M. P. Singh, "Pattern recall analysis of the hopfield neural network with a genetic algorithm," *Computers & Mathematics with Applications*, vol. 60, no. 4, pp. 1049–1057, 2010.

[288] M. G. Resende and J. P. de Sousa, *Metaheuristics: computer decision-making*. Springer Science & Business Media, 2013, vol. 86.

[289] D. O. Hebb, *The organization of behavior*. na, 1949.

[290] S. Lucci and D. Kopec, *Artificial intelligence in the 21st century*. Stylus Publishing, LLC, 2015.

[291] J. Liu, M. Gong, and Q. Miao, "Modeling hebb learning rule for unsupervised learning." in *IJCAI*, 2017, pp. 2315–2321.

[292] F. Sarwar, S. Iqbal, and M. W. Hussain, "Linear and nonlinear electrical models of neurons for hopfield neural network," *Zeitschrift für Naturforschung A*, vol. 71, no. 11, pp. 995–1002, 2016.

[293] A. J. Rahman, R. Guns, L. Leydesdorff, and T. C. Engels, "Measuring the match between evaluators and evaluees: cognitive distances between panel members and research groups at the journal level," *Scientometrics*, vol. 109, no. 3, pp. 1639–1663, 2016.

[294] H. Li, A. Shrestha, H. Heidari, J. L. Kernec, and F. Fioranelli, "Magnetic and radar sensing for multimodal remote health monitoring," *IEEE Sensors Journal*, vol. 19, no. 20, pp. 8979–8989, 2018.

[295] T. K. H. Nguyen, "Low power architecture for fall detection system," Ph.D. dissertation, 2015.

[296] S. Garg, B. K. Panigrahi, and D. Joshi, "An accelerometer based fall detection system using deep neural network," in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*. IEEE, 2019, pp. 1–7.

[297] K. Z. Rajab, B. Wu, P. Alizadeh, and A. Alomainy, "Multi-target tracking and activity classification with millimeter-wave radar," *Applied Physics Letters*, vol. 119, no. 3, p. 034101, 2021.

[298] Z. Yu, A. Zahid, S. S. Ansari, H. T. Abbas, H. Heidari, M. A. Imran, and Q. H. Abbasi, "Imu sensing-based hopfield neuromorphic computing for human activity recognition," *Frontiers in Communications and Networks*, 2021.

[299] Z. Yu, A. Zahid, W. Taylor, H. Heidari, M. A. Imran, and Q. H. Abbasi, "Multi-sensing data fusion for human activity recognition based on neuromorphic computing," in *2021 IEEE USNC-URSI Radio Science Meeting (Joint with AP-S Symposium)*. IEEE, 2021, pp. 64–65.

[300] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[301] A. D. Antar, M. Ahmed, and M. A. R. Ahad, "Challenges in sensor-based human activity recognition and a comparative analysis of benchmark datasets: a review," in *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*. IEEE, 2019, pp. 134–139.

[302] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.

[303] P. Casale, O. Pujol, and P. Radeva, "Personalization and user verification in wearable systems using biometric walking patterns," *Personal and Ubiquitous Computing*, vol. 16, no. 5, pp. 563–580, 2012.

[304] O. Baños, M. Damas, H. Pomares, I. Rojas, M. A. Tóth, and O. Amft, "A benchmark dataset to evaluate sensor displacement in activity recognition," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 1026–1035.

[305] S. Inoue, N. Ueda, Y. Nohara, and N. Nakashima, "Recognizing and understanding nursing activities for a whole day with a big dataset," *Journal of Information Processing*, vol. 24, no. 6, pp. 853–866, 2016.

[306] S. Faye, N. Louveton, S. Jafarnejad, R. Kryvchenko, and T. Engel, "An open dataset for human activity analysis using smart devices," 2017.

[307] J. A. Paredes, F. J. Álvarez, M. Hansard, and K. Z. Rajab, "A gaussian process model for uav localization using millimetre wave radar," *Expert Systems with Applications*, vol. 185, p. 115563, 2021.

[308] T. Aoki, J. F.-S. Lin, D. Kulić, and G. Venture, "Segmentation of human upper body movement using multiple imu sensors," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 3163–3166.

[309] M.-D. Hua, N. Manerikar, T. Hamel, and C. Samson, "Attitude, linear velocity and depth estimation of a camera observing a planar target using continuous homography and inertial data," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1429–1435.

[310] H. Ahmed and M. Tahir, "Improving the accuracy of human body orientation estimation with wearable imu sensors," *IEEE Transactions on instrumentation and measurement*, vol. 66, no. 3, pp. 535–542, 2017.

[311] J. Spörri, J. Kröll, B. Fasel, K. Aminian, and E. Müller, "The use of body worn sensors for detecting the vibrations acting on the lower back in alpine ski racing," *Frontiers in physiology*, vol. 8, p. 522, 2017.

[312] G. Garofalo, E. Argones Rúa, D. Preuveneers, W. Joosen *et al.*, "A systematic comparison of age and gender prediction on imu sensor-based gait traces," *Sensors*, vol. 19, no. 13, p. 2945, 2019.

[313] C. Granata, A. Ibanez, and P. Bidaud, "Human activity-understanding: A multilayer approach combining body movements and contextual descriptors analysis," *International Journal of Advanced Robotic Systems*, vol. 12, no. 7, p. 89, 2015.

[314] M. Vrigkas, C. Nikou, and I. A. Kakadiaris, "A review of human activity recognition methods," *Frontiers in Robotics and AI*, vol. 2, p. 28, 2015.

[315] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[316] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–33, 2014.

[317] S. Ankalaki and M. Thippeswamy, "Static and dynamic human activity detection using multi cnn-elm approach," in *Emerging Research in Computing, Information, Communication and Applications*. Springer, 2022, pp. 207–218.

[318] H. Bi, M. Perello-Nieto, R. Santos-Rodriguez, and P. Flach, "Human activity recognition based on dynamic active learning," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 4, pp. 922–934, 2020.

[319] A. Kanezaki, Y. Matsushita, and Y. Nishida, "Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5010–5019.

[320] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 922–928.

[321] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola, "Deep sets," *arXiv preprint arXiv:1703.06114*, 2017.

[322] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[323] P. Hermosilla, T. Ritschel, P.-P. Vázquez, À. Vinacua, and T. Ropinski, "Monte carlo convolution for learning on non-uniformly sampled point clouds," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–12, 2018.

[324] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 206–215.

[325] H. You, Y. Feng, R. Ji, and Y. Gao, "Pvnet: A joint convolutional network of point cloud and multi-view for 3d shape recognition," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1310–1318.

[326] Z. Han, H. Lu, Z. Liu, C.-M. Vong, Y.-S. Liu, M. Zwicker, J. Han, and C. P. Chen, "3d2seqviews: Aggregating sequential views for 3d global feature learning by cnn with hierarchical attention aggregation," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3986–3999, 2019.

[327] Z. Han, M. Shang, Y.-S. Liu, and M. Zwicker, "View inter-prediction gan: Unsupervised representation learning for 3d shapes by learning global shape memories to support local view predictions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8376–8384.

[328] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE transactions on information technology in biomedicine*, vol. 10, no. 1, pp. 156–167, 2006.

[329] Y. Hanai, J. Nishimura, and T. Kuroda, "Haar-like filtering for human activity recognition using 3d accelerometer," in *2009 ieee 13th digital signal processing workshop and 5th ieee signal processing education workshop*. IEEE, 2009, pp. 675–678.

[330] I. P. Machado, A. L. Gomes, H. Gamboa, V. Paixão, and R. M. Costa, "Human activity data discovery from triaxial accelerometer sensor: Non-supervised learning sensitivity to feature extraction parametrization," *Information Processing & Management*, vol. 51, no. 2, pp. 204–214, 2015.

Appendix

I, Zheqi Yu, confirm that this thesis and the work for human activities data collection of ethics is approved. The Ethical Approval file has shown on the following.

University of Glasgow | College of Science & Engineering

ETHICS APPLICATION FORM

This is a fast track application (an update to the previously approved application ( )) and nothing has changed compared to the already approved version except funding info and time frame.

1. DESCRIBE THE BASIC PURPOSES OF THE PROPOSED RESEARCH.

This research will investigate software defined networks (SDNs) based on universal software radio peripherals (USRPs) will help vulnerable individuals (older people and people with cognitive or physical impairments, or those with multi-morbidity conditions) preserve their independence and quality of life, secure the internet of things (IoT)-based data obtained, and provide caregivers and health professionals with individualised information on each patient.

In addition, the aim is to introduce a novel IoT computing framework for secure and smart healthcare services that can be connected with the 5G mainstream network. A distributed approach for clustering-based techniques will be developed for the proposed IoT framework with the scalability and flexibility to aggregate and analyse the large-scale and heterogeneous patient's data in the distributed IoT devices independently before it is sent to the cloud. The proposed framework will be demonstrated by evaluating a case study for the activity and bio-signal data.

In practical terms, this system will monitor activity levels over longer periods of time to detect early signs of cognitive and functional decline, providing not only prompt detection of critical events (e.g. falls, strokes, neurological disorders), but also predicting these events from indicators in the data that will enable individualized prompt treatment and intervention from health professionals while keeping the data secured.

2. INDICATE WHO IS FUNDING THE RESEARCH (IF COMMERCIALLY FUNDED, ENSURE THAT PARTICIPANTS ARE INFORMED).

This study gets funding from following 4 projects,

1) Project Title: *COG-MHEAR: Towards cognitively-inspired 5G-IoT enabled, multi-modal Hearing Aids*
   Project No. EP/T021063/1
   Funding source: EPSRC. (PI, Qammer Abbasi)
2) Project Title: *Quantum Imaging for Monitoring of Wellbeing & Disease in Communities*
   Project No. EP/T021020/1
   Funding source: EPSRC, (PI, Prof. Jon Cooper)
3) Project Title: *An intelligent non-invasive real-time elderly activity recognition system for next generation healthcare*
   Project No. 314038
   Funding source: EPSRC IAA (PI, Qammer Abbasi)
4) Project Title: *5G New Thinking (PI, Prof Muhammad Ali Imran)*
   Funding source: *https://www.5gnewthinking.co.uk*

3. DESCRIBE THE DESIGN OF YOUR EXPERIMENT (E.G. CONDITIONS, NUMBER OF PARTICIPANTS, PROCEDURE AND EQUIPMENT WHERE APPROPRIATE).

The proposed research aims to mainly detect human activity recognition in an indoor environment that could be necessary in healthcare applications.

Equipment is flexible and adoptable universal software defines radio USRPs X300/X310 from national instrument NI. Each USRP is equipped with extended-bandwidth daughterboard slots covering DC – 6 GHz with up to 120 MHz of baseband bandwidth used for human activity recognition. Omni directional antennas will be used for transmitting and receiving process, furthermore 1 GB cable used to the USRP to the host computer, both of the devices placed in line of side LOS and involved distance around 5 meters, to be able to communicate wirelessly without interference.

- In the first part of data collection, the USRPs will be placed in line of sight of each other with distance around 5 meters, then the participants will be asked to perform the activities. The data collection will be started once the participants are briefed

about the complete process and their consent is formally taken for the voluntary participation. Large scale daily life body movement related activities for the data collection include walking for a distance of up to 5 meter, sitting on and standing from chair, picking object from the floor, waving hand etc.

- In the second method of data collection, small scale activities shall be monitored like breathing, heartbeat, lip movement, blinking of eyes etc. The distance between USRP and human will be kept around 0.5 meters in order to obtain maximum accuracy.
- Initially we shall collect data from one participant. After analysis of that data we intend to collect data from up to 10 individuals. All the participation shall be voluntary. Individuals from the same research group shall be contacted in person and via email and requested for their voluntary participation. Each session of data collection will last for 30 minutes at max. One session of data collection shall be conducted for each individual.

4. DESCRIBE HOW THE PROCEDURES AFFECT THE PARTICIPANTS.
Main requirement from the participant is to do physical activities in indoor settings. Data will be collected from human adults without any negative effect on their health. The signals transmitted and reflected from the human body will provide information about the actions.

5. STATE WHAT IN YOUR OPINION ARE THE ETHICAL ISSUES INVOLVED IN THE PROPOSAL.
Possible issues may be related to time such that experiment may need to be repeated or may take longer time due to some error in some cases. To avoid such circumstances, we shall try our best that equipment will be fully functional at the time of experiment. To ensure it we shall pre-test the equipment before the experiment time.

6. SPECIFY THE NATURE OF THE PARTICIPANTS. INDICATE IF THE RESEARCH INVOLVES CHILDREN OR THOSE WITH MENTAL DISABILITIES OR HANDICAP. IF SO, EXPLAIN THE STEPS TAKEN TO OBTAIN PERMISSION FROM L.E.A.s, HEADTEACHERS, PARENTS, ETC. GIVE A BRIEF DESCRIPTION HERE AND FILL IN THE CHILDREN RESEARCH ETHICAL PLAN. THE FORM MUST BE UPLOADED TOGETHER WITH THE CONSENT AND INFORMATION FORMS.

It will be ensured that participants in the research are adult. Data collection from the children or people with some disability is not required as part of this research.

COLLEGE OF SCIENCE & ENGINEERING ETHICS APPLICATION FORM

7. STATE IF PAYMENT WILL BE MADE TO SUBJECT. It will be an unpaid volunteer work.
8. DESCRIBE THE PROCEDURES FOR ADVERTISING, FOR RECRUITING PARTICIPANTS, AND FOR OBTAINING CONSENT FROM PARTICIPANTS. Primary participants will be recruited by circulating an email in the PhD research group for volunteer participation. Beside that help from network of friends and friends of friends shall be taken for volunteer recruitment. Further to it university staff like the academics involved may also take part in the research. The experiment will be explained in the email and will also be explained before the data collection. Experiment will be conducted on the formal agreement of the participant by signing the consent form. The consent form to be used is attached to this application. Consent forms will be kept in paper form by the PI of this study.
9. STATE WHETHER THE PROPOSAL IS IN ACCORD WITH THE BPS CODE OF CONDUCT OR THE ESRC FRAMEWORK OF RESEARCH ETHICS. We believe that this simple proof of concept study complies with the necessary ethical standards requested by the University of Glasgow. It harms no individual physically, medically or emotionally.
10. DESCRIBE HOW THE PARTICIPANTS' ANONYMITY AND CONFIDENTIALITY WILL BE MAINTAINED. Participants' anonymity shall be ensured in data processing and results publication. The participants' names will only appear on the consent forms, which will be with the PI at the University of Glasgow premises. Participants' age and gender will be recorded for only research purposes. None of the information on the consent form shall be disclosed to anyone that may lead to identification of the individual. For that electric record generated will be assigned an anonymous identifier just to discriminate different participants while the specific identity of the participants will not be discoverable from the electric records. Apart from the activity data collected using USRP, only personal information to be taken includes the name, age, and gender of the participants. Name shall also be encoded with anonymous identifier before storage in databases. Research to be published shall be completely anonymous and mostly in the form of statistical summary e.g., percentage of participants with dehydration or sugar level identified correctly by the model.
11. DATE ON WHICH PROJECT WILL BEGIN AND END. This application is an update to our previously approved application (Application No. 300190109). Total project duration is 5 years. We will submit extension every year.
12. LOCATION AT WHICH THE PROJECT WILL BE CARRIED OUT. The project will be carried out at this stage only at the James Watt School of Engineering, University of Glasgow, Glasgow, UK or its UESTC College in Chengdu China.
13. DESCRIBE HOW PARTICIPANTS WILL BE DEBRIEFED AT THE END OF THE EXPERIMENT (THIS MUST INCLUDE THE OPPORTUNITY TO CONTACT THE EXPERIMENTER - OR SUPERVISOR - FOR FEEDBACK ON THE GENERAL OUTCOME OF THE EXPERIMENT). The participants will be given contact of the PhD students and PI of the study and be clearly told that they can at any point request information on their data and on the status of the study. As the participants are mostly part of the same social network so it would be easy to keep them updated.

University of Glasgow | College of Science & Engineering

Dr. Christoph Scheepers
Senior Lecturer

School of Psychology
University of Glasgow
62 Hillhead Street
Glasgow G12 8QB
Tel.: +44 141 330 3606
Christoph.Scheepers@glasgow.ac.uk

Glasgow, February 10, 2020

Ethical approval for:

Application Number:

Project Title: Software Defined Radios (SDRs) Based Testbed for Large Scale-Small Scale Body Movements

Lead Researcher: Dr Qammer H Abbasi

This is to confirm that the above application has been reviewed by the College of Science and Engineering Ethics Committee and **approved.** Please refer to the collated reviews on the system for additional comments and suggestions, if any. Good luck with your research.

Sincerely,

Dr Christoph Scheepers
Ethics Officer
College of Science and Engineering
University of Glasgow