Novel Stacking Classification And Prediction Algorithm Based Ambient Assisted Living For Elderly

محرك تسلسل رقمي للحمض النووي لتحليل برامج طلب الفدية باستخدام شبكة تعلم الآلات

by

JINESH PADIKKAPPARAMBIL

A thesis submitted in parial fulfilment

of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

at

The British University in Dubai

September 2022

NOVEL STACKING CLASSIFICATION AND PREDICTION ALGORITHM BASED AMBIENT ASSISTED LIVING FOR ELDERLY

محرك تسلسل رقمي للحمض النووي لتحليل برامج طلب الفدية باستخدام شبكة تعلم الآلات

by

JINESH PADIKKAPPARAMBIL

A thesis submitted to the Faculty of Computer Science

in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

at

The British University in Dubai

September 2022

Thesis Supervisor
Dr Cornelius Ncube

Approved for award:

Name Designation	Name Designation
Name Designation	Name Designation

Date: ____________

DECLARATION

I warrant that the content of this research is the direct result of myown work and that any use made in it of published or unpublished copyright material falls within the limits permitted by international copyright conventions.

I understand that a copy of my research will be deposited in the University Library for permanent retention.

Ihereby agree that the material mentioned above for which I am author and copyright holder may be copied and distributed by The British University in Dubai for the purposes of research, private study or education and that The British University in Dubai may recover from purchasers the costs incurred in such copying and distribution, where appropriate.

I understand that The British University in Dubai may make a digital copy available in the institutional repository.

I understand that I may apply to the University to retain the right to withhold or to restrict access to my thesis for a period which shall not normally exceed four calendar years from the congregation at which the degree is conferred, the length of the period to be specified in the application, together with the precise reasons for making that application.

Signature of the student

ABSTRACT

The ageing of the population in developed nations necessitates the expansion of medical services, raising the cost of both economic and human resources. In this respect, Ambient Assisted Living (AAL) is a comparatively new information and communication technology (ICT) that delivers services. It acknowledges numerous products that help elderly and disabled people to live autonomously and enrich the quality of their lives. It also aids in the cost-cutting of hospital services. Various sensors and equipment are installed in the AAL context to collect a wide variety of data. Furthermore, AAL could be the motivating technique for the most recent care models by working as an adjunct.

Research and development (R&D) projects and business activities in AAL and smart home contexts frequently emphasize the significance of technology. While ICTs promote health care, they have the potential to alleviate loneliness and social isolation among the elderly. They help to enhance, expand, and maintain the social interactions of the elderly while also increasing the individual's emotional well-being.

The emergence of smart homes will help the elderly and the disabled live better lives. Over the past five years, the acceptance of wearable fall detection technologies has increased. These techniques involve calling for help in an emergency, such as falling or being immobile for long periods. Because falling is widespread among older adults, it can have serious health consequences. Falls can result in physical traumas such as fractures, head injuries, and severe decay. Falls will have a considerable impact on some populations, necessitating the development of better fall prevention and management solutions.

Therefore, this thesis proposed a Novel Stacking Classification and Prediction (NSCP) algorithm based on AAL for the older people with Multi-strategy Combination based Feature Selection (MCFS) and Novel Clustering Aggregation (NCA) algorithms. The primary objective of this thesis is to identify the fall detection and prediction in older persons, such as 0 - no fall detected, 1 - person slipped/tripped / fall prediction, and 2 - definite fall. This study's dataset was sourced from the Kaggle machine learning repository, and it refers to data gathering from wearable IoT devices. The experimental outcomes demonstrate the proposed MCFS, NCA, and NSCP algorithms work more effectively than previous feature selection, clustering and classification algorithms, respectively, in terms of accuracy, sensitivity, specificity, precision, recall, f-measure and execution time. This thesis concludes with a discussion of future work to improve the proposed methodology and future research directions.

الملخص

تتطلب شيخوخة السكان في الدول المتقدمة التوسع في الخدمات الطبية ، مما يرفع تكلفة الموارد الاقتصادية والبشرية. في هذا الصدد ، تعتبر تقنية Ambient Assisted Living (AAL) تقنية معلومات واتصالات جديدة نسبيًا (ICT) تقدم الخدمات. وهي تعترف بالعديد من المنتجات التي تساعد كبار السن والمعاقين على العيش بشكل مستقل وإثراء نوعية حياتهم. كما أنه يساعد في خفض تكاليف خدمات المستشفيات. يتم تثبيت العديد من أجهزة الاستشعار والمعدات في سياق AAL لجمع مجموعة متنوعة من البيانات. علاوة على ذلك ، يمكن أن تكون AAL هي التقنية المحفزة لأحدث نماذج الرعاية من خلال العمل كمساعد.

كثيرًا ما تؤكد مشاريع البحث والتطوير (D&R) وأنشطة الأعمال في سياقات AAL والمنزل الذكي على أهمية التكنولوجيا. بينما تعزز تكنولوجيا المعلومات والاتصالات الرعاية الصحية ، فإن لديها القدرة على تخفيف الشعور بالوحدة والعزلة الاجتماعية بين كبار السن. إنها تساعد على تعزيز وتوسيع والحفاظ على التفاعلات الاجتماعية لكبار السن مع زيادة الرفاهية العاطفية للفرد. سيساعد ظهور البيوت الذكية كبار السن والمعاقين على عيش حياة أفضل. على مدى السنوات الخمس الماضية ، ازداد قبول تقنيات الكشف عن السقوط القابلة للارتداء. تتضمن هذه الأساليب طلب المساعدة في حالات الطوارئ ، مثل السقوط أو عدم القدرة على الحركة لفترات طويلة. لأن السقوط منتشر بين كبار السن ، يمكن أن يكون له عواقب صحية خطيرة. يمكن أن يؤدي السقوط إلى إصابات جسدية مثل الكسور وإصابات الرأس والتسوس الشديد. سيكون للشلالات تأثير كبير على بعض السكان ، مما يستلزم تطوير حلول أفضل لمنع السقوط وإدارته.

لذلك ، اقترحت هذه الأطروحة خوارزمية تصنيف وتوقع تراكم الروايات (NSCP) استنادًا إلى AAL لكبار السن باستخدام خوارزميات اختيار الميزات القائمة على مزيج متعدد الإستراتيجيات (MCFS) وتجميع تجميع الروايات (NCA). الهدف الأساسي من هذه الأطروحة هو تحديد اكتشاف السقوط والتنبؤ به لدى كبار السن ، مثل 0 - عدم اكتشاف السقوط ، 1 - توقع سقوط / تعثر / سقوط الشخص ، و 2 - سقوط مؤكد. تم الحصول على مجموعة بيانات هذه الدراسة من مستودع التعلم الآلي Kaggle ، وهي تشير إلى جمع البيانات من أجهزة إنترنت الأشياء القابلة للارتداء.

توضح النتائج التجريبية أن خوارزميات MCFS و NCA و NSCP المقترحة تعمل بشكل أكثر فاعلية من اختيار الميزات السابقة وخوارزميات التجميع والتصنيف ، على التوالي ، من حيث الدقة والحساسية والنوعية والدقة والاستدعاء والقياس ووقت التنفيذ. تختتم هذه الأطروحة بمناقشة العمل المستقبلي لتحسين المنهجية المقترحة وتوجهات البحث المستقبلية.

TABLE OF CONTENTS

TABLE OF TABLES

TABLE OF FIGURES

CHAPTER 1

Overview: Elderly Fall Detection in Ambient Assisted Living

1.1 Introduction

Recent medical advancements have allowed humans to live a long and healthy life than prior generations. According to the 2017 Revision of World Population Prospects, by 2050, the number of people aged 60 and more will have more than doubled, growing from 962 million in 2017 to 2.1 billion (Béjot and Yaffe, 2019). Due to cognitive decline, recurrent age-related disorders, and restrictions in physical activity, eyesight, and hearing, ageing presents many obstacles to older persons. As per the World Health Organization (WHO), over 646 thousand fatal falls occur globally each year, with most casualties being adults over 60 (WHO, 2018). It is the second leading cause of injury death following road traffic injuries. Falls among the elderly are a significant public health concern worldwide. Without a doubt, injuries sustained by elderly individuals as a result of falls have far-reaching consequences for their families and healthcare institutions and society at large.

As per Google Trends, Fall Detection is becoming more prominent in academics and business over the last several years, as seen in Figure 1 (Google Trends, 2022). Similarly, a potential fall forecast topic is also crucial, as it is linked to various services that concentrate on prevention and security.

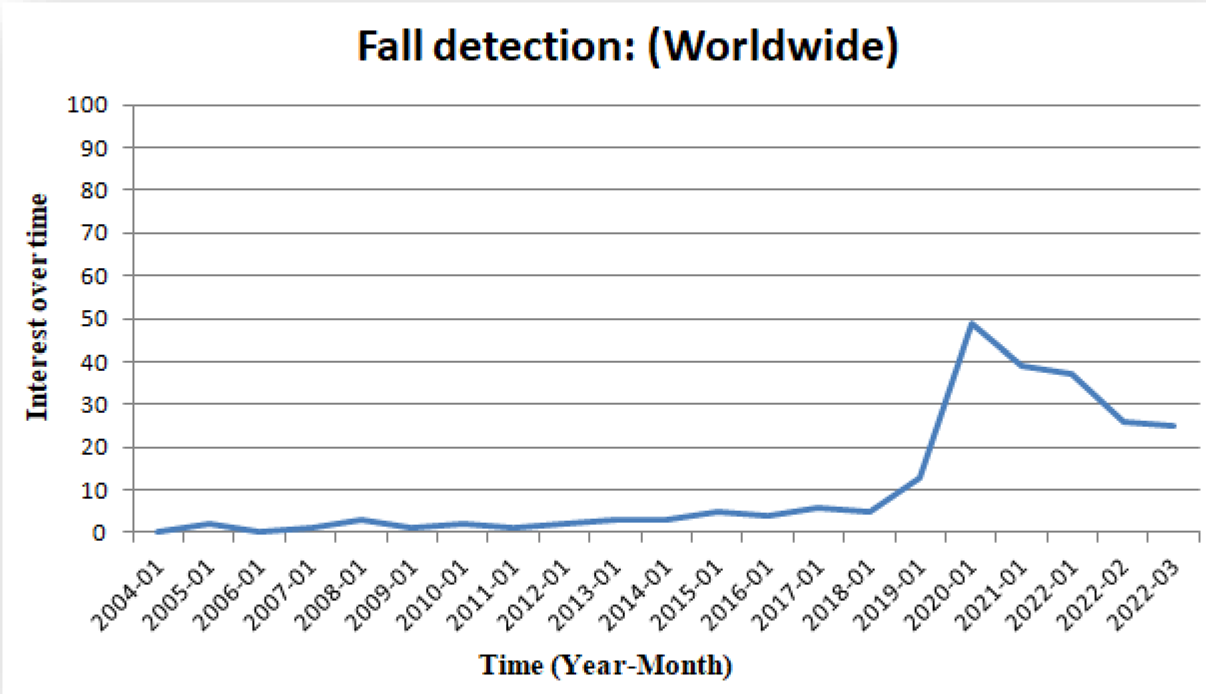

Figure 1. 1: Curiosity to detect falls over time from January 2004 to March 2022. The data was gathered using Google Trends and the search term "fall detection." The numbers are normalised using the greatest interest; therefore, the maximum interest has a number of 100

(Xu, Zhou, and Zhu, 2018) examined the opportunities and issues of detecting and predicting older people's falls. Predictive methods are aimed to predict fall occurrences during or before their incidence, allowing immediate actions like the activation of airbags. In contrast, Detection methods are concerned with realising falls after they happen and triggering an alarm to urgent caretakers. In recent decades, much effort has been made into these sectors to enhance the precision of fall detection and forecasting systems and reduce false alarms.

Given that 89% of the elderly want to be comfortable in their own homes instead of nursing home maintenance costs (Carver, Beamish, Phillips and Villeneuve, 2018). Therefore, it is essential to develop technologies that help the elderly.

In recent years, researchers have implemented several assistive devices using a novel paradigm known as "ambient intelligence". Ambient intelligence (Oguego, Augusto, Muoz, and Springett, 2018) is a novel concept in information technology that aims to empower a person's capacities through digital settings that are sensitive, flexible, and responsive to people's requirements. This vision of everyday life will allow for novel human-machine communications characterised by ubiquitous, inconspicuous, and predictive interactions.

Ambient-assisted living (AAL) tools are assisted living systems based on ambient intelligence. AAL could be utilized to prevent, treat, and improve the welfare of older people. Medication management systems and medication reminders are examples of AAL technologies that help older individuals take charge of their health (Morrissey, Casey, Glynn, Walsh, and Molloy, 2018). Mobile emergency response systems (Sandeepa, Moremada, Dissanayaka, Gamage, and Liyanage, 2020), fall detection systems (Sarabia-Jácome, Usach, Palau, and Esteve, 2020), and video surveillance systems (Ranieri, MacLeod, Dragone, Vargas and Romero, 2021) can all help the elderly feel safer. Other AAL technologies assist with everyday lives by tracking activities of daily living (ADL) and giving reminders (Kozan, Phan, Thaker, Sedarat, and Issa, 2021), and assist with the help of automation and mobility. Finally, such technology can assist the elderly engage and interacting more effectively with their peers and their families and friends (Binda, Yuan, Cope, Park, Choe and Carroll, 2018).

Recent advances in many technology areas have helped make AAL's vision changeable. These technologies include intelligent houses, assisted robotics, and mobile and wearable sensors.

A smart home is a typical home that has been enhanced with sensors and actuators of various types. Rich contextual data can be retrieved by evaluating and integrating different kinds of sensor data. Many smart homes use this information for automation and higher comfort for people and measure their physical and cognitive health.

Sensors include the global positioning system (GPS), accelerometer, proximity sensor, and gyroscope that could detect user activity and movement in today's smartphones. Latest developments in epidermal electronics and Micro Electro Mechanical Systems (MEMS) technology also point to a novel epoch in wellbeing sensing devices. Researchers have previously implemented unobtrusive sensors for monitoring health signals in patches, miniature Holter-type equipment, wearable devices, and smart clothing. For instance, wearable sensors could calculate blood glucose, blood pressure, and heart function utilizing infrared sensitivity, optical sensitivity, and oscillometric methods.

Assistive robots help older adults overcome physical restrictions by assisting them in daily chores. Assistive robots come in three varieties: those that help with ADL activities, those that help with instrumental activities of daily living (IADL), and those that help with enhanced activities of daily living (EADL). Examples of ADL duties are feeding, dressing, grooming, and other self-maintenance chores. IADL duties contain the capacity to utilize instruments in everyday life, like successfully using the phone, cooking food, etc. One of the activities of EADLs is to participate in social tasks, such as hobbies. The goal of this thesis was to employ AAL to detect elderly falls.

1.2 Fall Detection and Prevention Systems

The creation of fall detection and prevention systems has become a prominent research topic in previous years. A variety of approaches are employed in the construction of such systems. These systems could be classified into two groups: Systems that are both wearable and non-wearable.

1.2.1 Non-Wearable Systems

Non-wearable systems are made of sensors placed close to humans for data/gait monitoring. These systems use two types of sensors: vision and floor-based sensors (Viswakumar, Rajagopalan, Ray and Parimi, 2019). Laser Range Scanners (LRS), infrared sensors, and Cameras (Rizk, Yamaguchi, Youssef, and Higashino, 2020) are vision sensors that take visual observations and analyse them using the image processing. A typical system is video surveillance, which takes images and employs various algorithms to predict fall incidence. For example, Ground Reaction Force (GRF) sensors and pressure sensors monitor the force applied by a person's feet to detect a fall (Clemente, Song, Valero, Li and Liy, 2019). The quantity of sensors used in each experiment varies. The main disadvantage of non-wearable technologies is the restricted coverage they provide. These systems may be used in offices, residences, and research labs, providing they are low scalable and costly. Users' privacy is likewise compromised by non-wearable technology (Singh, Luhach, Gao, Kumar and Roy, 2020). As a result, most real-world applications do not benefit from adopting such systems.

1.2.2 Wearable Systems

Wearable systems are data-collecting sensors or devices which can be fixed on the body of a human. In wearable systems, accelerometers, gyroscopes, magnetometers, IMUs, and other sensors are used (Lou, Wang, Jiang, Wei and Shen, 2020). Table 1.1 provides a summary of wearable sensors.

Table 1. 1: Summary of Wearable Sensor

Sensor Name	Functionality
Accelerometer	Measures the rate at which an object's velocity changes (acceleration).
Gyroscope	Rotational variations with orientation are measured. As a result, the angular velocity is calculated along three axes: pitch (x-axis), roll (y-axis), and yaw (z-axis)
Magnetometer	Measures the strength, direction, and comparative variation of a magnetic field.
Inertial Measurement Unit (IMU)	An accelerometer, gyroscope, and magnetometer are included. It has 2 to 6 degrees of freedom, which relates to various object motions in three dimensions.
Surface Electromyography (sEMG)	It detects potentials by placing electrodes on the skin and utilizing an electrochemical transducer.

Collecting data outside of the laboratory setting is the real benefit of wearable systems (Niknejad, Ismail, Mardani, Liao and Ghani, 2020). As a result, these technologies can analyse fall detection or prevent falls whilst doing ADLs. In addition, these sensors are frequently built-in smartphones, allowing information to be gathered without extra devices (Shahzad and Kim, 2018). Furthermore, they offer more secrecy than non-wearable technologies. But on the other hand, Wearable gadgets have restricted processing power over time (Khan, Saboor, Kim and Park, 2021). Moreover, the data from the wearable must be processed utilizing statistical or ML techniques to make decisions.

Statistical procedures frequently result in lower classification accuracy and are ineffective when dealing with noisy data. As a result, ml algorithms are routinely employed to detect and prevent falls (Hemmatpour, Ferrero, Montrucchio and Rebaudengo, 2019). The primary fall activities have been identified as falling forwards, backwards, sides, rotating clockwise, and rotating anti-clockwise direction (Kim, Choi, Heo, Kim, Lee and Mun, 2019). In addition, ML algorithms categorize fall events from non-fall events (Salleh and Koh 2020). Likewise, these ML classifiers detect anomalies in gait and use approaches like muscle stimulation to avoid falls (Kumar, Ha, Sawicki, and Liu, 2020).

1.2.3 Overview of the System:

The four phases in the entire system for fall discovery and prevention are represented in Figure 1.2.

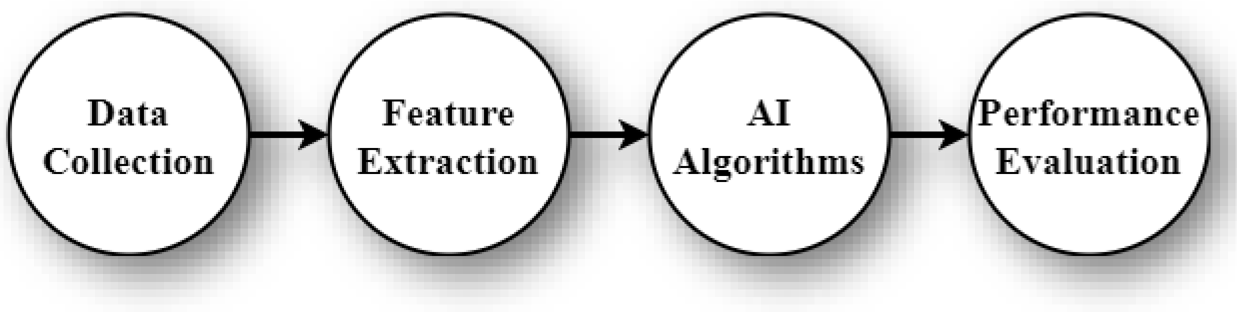

Figure 1. 2: Fall detecting and preventing process

The initial phase, relying on the application's needs, is data collection. Public datasets and controlled and realistic settings are all examples of data collection approaches. Accelerometer readings are included in publicly available datasets that could be utilised to construct such systems

(Putra, Brusey, Gaura and Vesilo, 2018). A lab or realistic setting, on the other hand, collects data via wearable (Saleh and Jeannès, 2019) or non-wearable (Dubois, Mouthon, Sivagnanaselvam, and Bresciani, 2019) technologies.

The second phase is feature extraction, which takes the collected dataset and extracts the preferred features. The features differ from one experiment to the next and from one researcher to the next. For instance, Sound length, matching filters, and relative power are desirable features in voice recognition. Similarly, in computer vision applications, edges and objects are preferred features. Fall detection and prevention applications, on the other hand, desired a feature set that includes pressure, heart rate variability, and peripheral oxygen saturation level. A slight shift in any of these parameters can aid visualise gait variations, leading to falling detection or prevention. As a result, these features' mean and standard deviation are regarded helpful information for this application. Therefore, feature selection is critical because classification accuracy is strongly dependent on the features used. Feature selection further decreases the size of the dataset and the cost of pattern recognition. Filter methods, wrapper methods, and embedding methods can all pick features. A vast number of features can cause overfitting, whilst fewer features might cause underfitting. As a result, this phase necessitates extra attention to improve the system's overall effectiveness.

Artificial Intelligence (AI) algorithms are used to classify ADL, abnormal gait, and falls in the third phase. It separates the dataset into two categories: training and testing datasets. The system design experiment determines the ratio of each data type. This pace uses Machine Learning (ML) and Deep Learning (DL) algorithms to detect fall behaviours or abnormal gaits to prevent falls.

The classifier's effectiveness is assessed utilizing test data after being trained. This phase examines the system efficiency using several metrics, including accuracy, precision, and recall of the findings obtained.

1.3 Artificial Intelligence

AI is a method that enables machines to mimic human behaviour and nature. AI makes it possible for machines to learn from their experience. By processing enormous data and identifying patterns, the machines adjust their responses based on new inputs, allowing them to do human-like tasks. AI is the broader umbrella under which ML and DL come (Sinha, Sachan and Parthasarathi, 2021). In addition, as shown in Figure 1.3, even DL is a subset of ML. As a result, AI, ML, and DL are all subsets of each other. So, let us figure out how they differ from each other.

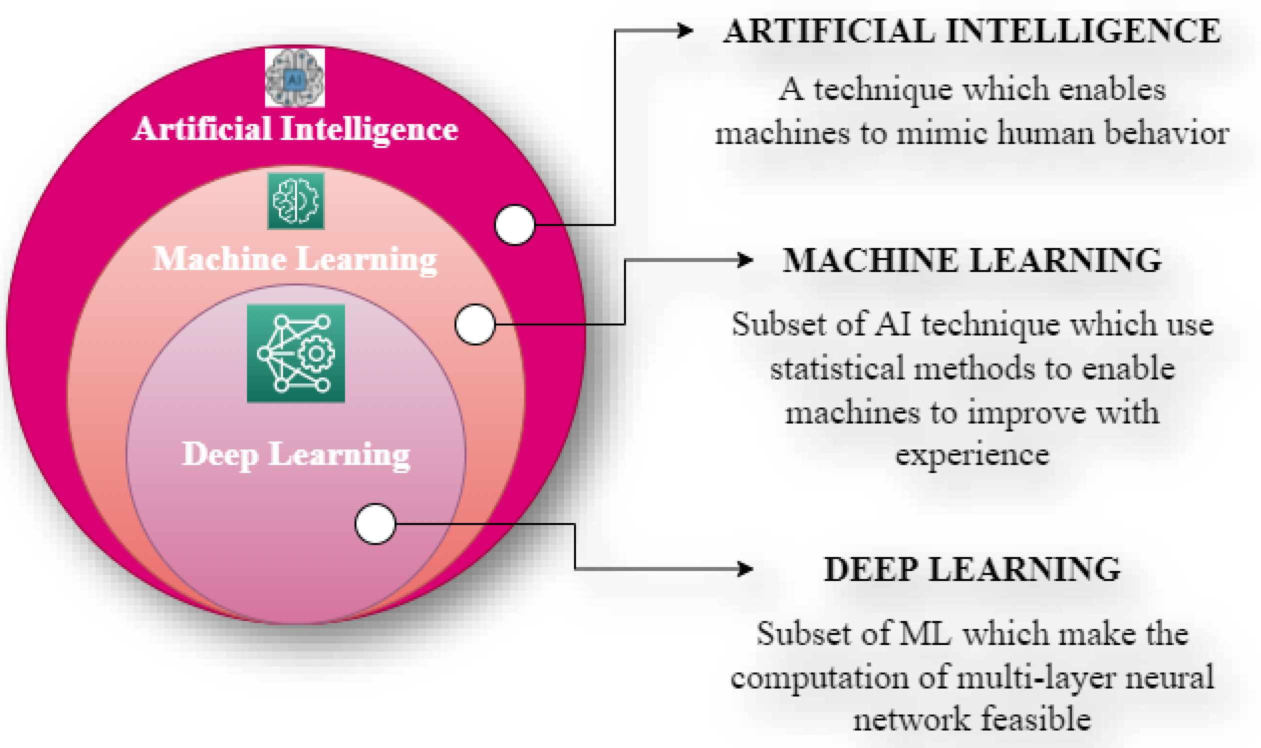

Figure 1. 3: AI vs ML vs DL

As can be seen, ML algorithms aid in identifying and preventing falls. As a result, the functionality of the main ML algorithms utilized for fall detection and prevention is discussed in the subsequent section.

1.4 Machine Learning

ML is a subset of AI. It allows machines to study from their past experiences and make predictions (data) (Alzubi, Nayyar and Kumar, 2018). In addition, ML provides the scheme with the capacity to learn depending on the dataset. Sensors offer data related to various fall parameters throughout the data collection process. Consequently, ML algorithms are utilised to analyse data to categorise or detect fall behaviours according to the application's needs. The following are the most broadly used ML methods to identify and prevent falls:

1.4.1 Support Vector Machine (SVM):

SVM is a supervised ML approach for finding a hyperplane in an n-dimensional space (here, N is the number of features that separate the data). Specifically, a hyperplane reduces an n-dimensional area to (n-1) dimensions. Although the SVM can be used for classification and regression, it is most typically used for classification (Shah, Hussain and Batool, 2020). SVM is classified into two types: linear and non-linear. The linear classifier (Yang, Si, Wang, and Guo, 2018) assumes that each data point could be separated sequentially. As a result, it distinguishes between the two classes via determining the ideal hyper plane with the highest margin. But, at the frequently utilized non-linear classifier (Zhang, Wei, Zhang, and Liu, 2019), the data is initially mapped using a kernel before a discriminant function is discovered. In the converted space, that discriminant function is connected by the hyperplane. Furthermore, the kernel is employed in various ML techniques for pattern recognition.

The data points in SVM are accessible on a plane and could be achieved from several classes using their locations. A hyperplane (Hamidzadeh and Moslemnejad, 2019) serves as a decision boundary to categorise these data points. Several hyperplanes to choose from, but the essential goal is to locate the plane with the most significant distance among the support vectors, as illustrated in Figure 1.4.

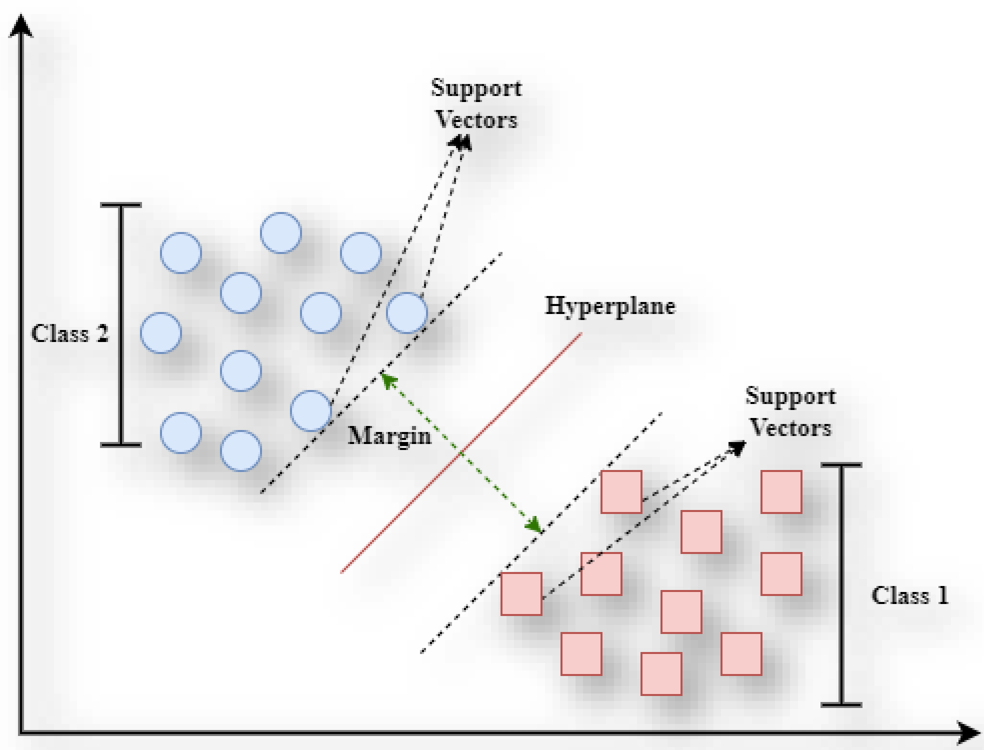

Figure 1. 4: SVM Algorithm

Support vectors are data points close to the hyperplane and perform a vital part in determining the hyperplane's spot. The SVM attempts to construct a decision boundary (Li, Zhang, Xu, Yao, Lau, and Wu, 2017) with the most significant distance between the classes. As a result, it's a standard classifier for separating fall behaviours from everyday activities. For instance, Class 1 in Figure 1.4 reflects the usual walking pattern, whereas class 2 depicts the fall feature.

1.4.2 Artificial Neural Network (ANN):

An ANN is an ML algorithm whose method is influenced by how the human brain works (Gupta and Raza, 2019). The human brain comprises billions of neurons that procedure data as electric signals. In general, neurons make decisions based on the intensity of the electric signal. Likewise, as shown in Figure 1.5, the ANN comprises numerous interlinked processing components that work together to solve a particular issue.

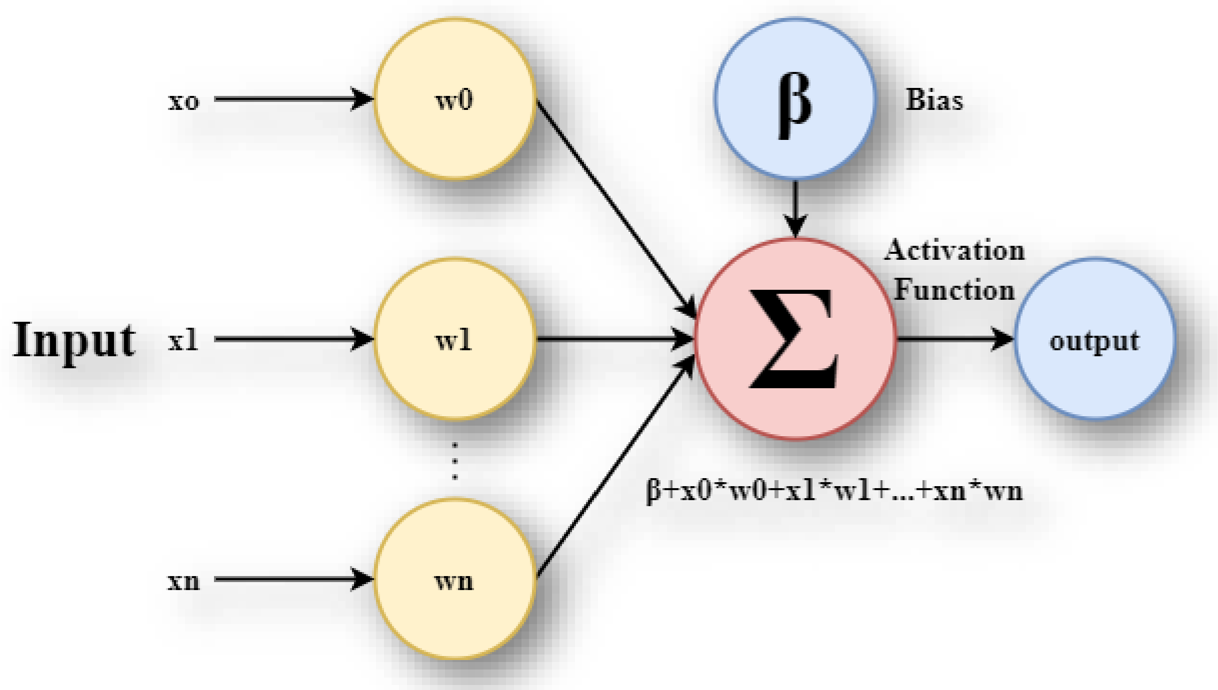

Figure 1. 5: ANN Algorithm

In this case, x variables (x0, x1, x2,..., xn) are the ANN's input, while w variables (w0, w1, w2,..., wn) are the weights of each input. Each weight represents the intensity of the input signal. By shifting the activation function up or down, the bias function (b) alters it. The activation function transforms the input function into the ANN's output signal. There may be extra hidden layers between the input and output layers to improve data classification in addition to the input and output layers. As a result, ANN is a Multi-Layer Perceptron (MLP) (Abdullah, Ismail and Ahmed, 2019). When a hidden layer exists, the ANN output signal of one layer can be used as the input signal for the next layer. The ANN needs a trainer to identify the response for each given input. As a result, more training is necessary to forecast the outcome for each input. A cost function is employed to calculate the gap between the actual and predicted values (Nowlan and Hinton, 2018). The cost function is calculated for each layer of the network, and weights are modified for the next input. This procedure continues until the minimal cost function is established, which provides the smallest difference between the actual and forecasted values. As a result, it can aid in predicting or preventing falls.

1.4.3 Random Forest (RF):

As seen in Figure 1.6, a random forest is a supervised classification algorithm that employs many decision trees.

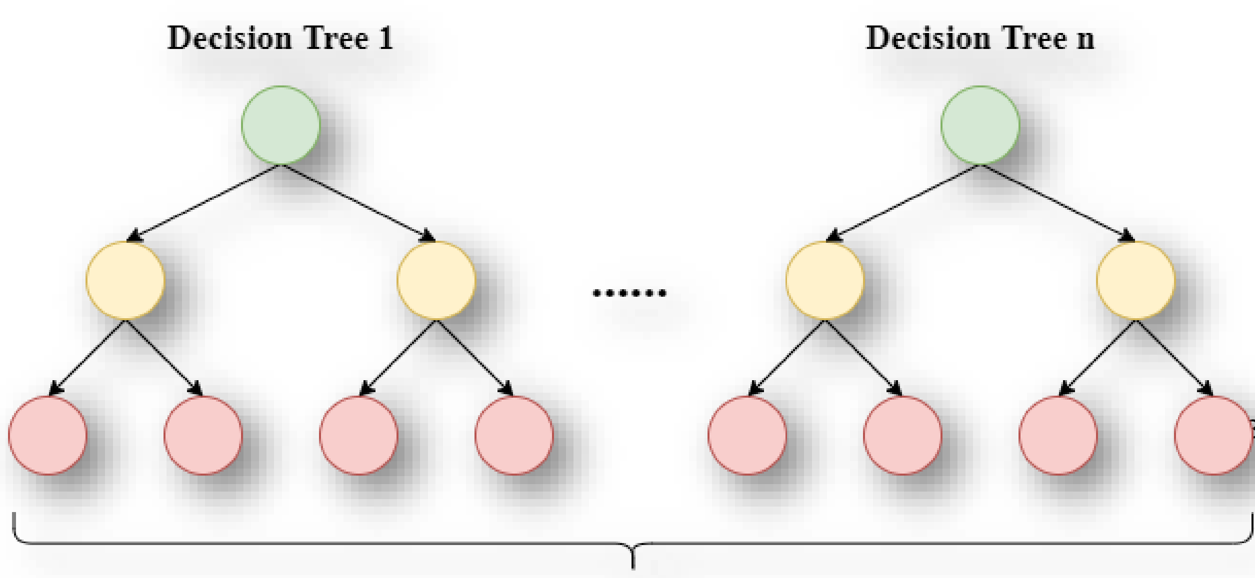

Figure 1. 6: RF Algorithm

A decision tree comprises nodes (each indicating a test) and branches that indicate the individual test result (Ayyadevara, 2018). Following that, it constructs the categorization using the routes in the tree. On the other hand, the decision tree is responsive to training data. Slight shifts in training data greatly affect the final tree structure. As a result, bagging is employed in the RF. Each tree generates numerous decision trees in bagging by randomly selecting a random sample from the dataset. Each tree in the RF belongs to a class, and voting is used to determine the model's finest prediction class.

RF adheres to the rule of the wisdom of crowds. To forecast the outcome, each tree uses the decision tree model. The predicted class has the most votes, which is forecasted by several trees in the RF). To circumvent the overfitting issue, an RF could be utilized to model categorical data. As a result, it is useful for distinguishing ADL from fall behaviours.

1.4.4 k-Nearest Neighbors (kNN):

kNN is a supervised learning technique that can tackle classification and regression problems. kNN implies that comparable objects are near together (Wibowo and Oesman, 2020). As a result, it attempts to compute the Euclidean distance among the data points. First, a k value is chosen to determine the number of neighbours for a data point. The k value is in charge of delineating the boundaries of classes or clusters. kNN produces a sorted list of the distances between each point and the other. The initial k elements are ultimately chosen for classification and regression. kNN is a very basic and low-cost algorithm. As a result, it is commonly employed in recommender systems (Saha, Chowdhury and Biswas, 2020). As a result, kNN can aid in discriminating between fall and non-fall activities.

1.4.5 k-Means:

K-means is a straightforward and widely used unsupervised ML technique. K-means cluster related data points to find hidden patterns (Hossain, Akhtar, Ahmad, Rahman, 2019). It finds the different clusters (k) in a dataset for categorization. A cluster comprises comparable data points related to certain actions, for example, the gait of a fall. It specifies the number of centroids required in the dataset by specifying k. Each cluster's centroid is located in the middle. All data point is assigned to the closest cluster while centroids are kept as minimal as feasible. This method is widely employed in data cluster analytics and feature learning. However, the k-mean algorithm's performance can change since a small alter in information can outcome in a large variation. As a result, it is infrequently employed in fall detection or prevention techniques.

1.4.6 Linear Discriminant Analysis (LDA):

LDA is a supervised classification algorithm that reduces dimensionality (Kumar, Pirogova and Fang, 2018). It is mostly employed as a pre-processing phase. The primary purpose of LDA is to convert high-dimensional data to low-dimensional data that aids in price and resource reduction. Typically, gait analytics uses wearables to collect huge amounts of data with comparable patterns. As a result, LDA aids in decreasing the data dimensionality, particularly when processing is performed on low-power devices.

1.4.7 Naive Bayes:

Naive Bayes is another supervised learning method that operates on the Bayes Theorem premise (Yang, 2018). It is one of the most basic and extensively utilized classification techniques to produce quick prediction results. In essence, it uses the Bayes theorem to build classes using probability. Then, abnormal gait and falls could be easily identified depending on the classes.

The methods mentioned above are mostly employed in Systems for detecting and preventing falls. Other techniques for comparable applications include Logistic Regression, Dynamic Time Wrapping (DTW), and decision trees. Another method for detecting falls to consider them an anomaly discovery issue. Autoencoders are employed in such systems to detect falls (Elshwemy, Elbasiony, and Saidahmed, 2020). By training the ADL models, the autoencoder learns features. As a result of the reconstruction error, fall activities are detected as an anomaly. It comprises an encoder, a decoder, and a code layer. An encoder is a programme that learns and reduces the relevant attributes of the input. The code layer serves as the intermediate layer, including compressed and pertinent data details. On the other hand, the decoder converts the information back into its unique form. This approach could assist in reducing the dimensionality of data, obtaining necessary gait attributes, and detecting previously unnoticed falls (Nogas, Khan and Mihailidis, 2018).

1.5 Deep Learning

DL is a specific type of ML that learns to describe the world as a deep hierarchy of ideas or abstraction, giving it a lot of power and flexibility (Shinde and Shah, 2018). DL models can be compared to a rocket engine; with the massive volume of data that we provide, these algorithms serve as the fuel. DL isn't a brand-new notion. However, its popularity has recently risen, and DL is receiving more attention. ANNs are a type of ML inspired by the operation of human brain cells.

It modifies the data connections between all artificial neurons to match the data pattern. If the amount of data is enormous, more neurons are required. It incorporates learning at various levels of abstraction, enabling a system to learn complicated function mapping without relying on any particular algorithm. For instance, in image processing, the lesser layers can recognize the edges, whilst the upper layers can discover human-related concepts, such as numbers, letters, or faces. A deep neural network (DNN) is one of the important architecture of the DL.

An ANN with several layers among the input and output layers is a DNN (Li, Jiang, Chen, Zhang and Du, 2018). Neurons, synapses, weights, biases, and functions are all basic components of neural networks in various shapes and sizes. These components function similarly to human brains and can be trained like an ML algorithm.

A DNN trained to detect dog breeds, for example, would examine a given image and determine the possibility that the dog in the image is of a specific breed. The user may examine the findings and choose which probabilities the network must reveal (those greater than a specific threshold, and so on) before returning the recommended label. Each mathematical operation is called a layer, and sophisticated DNNs contain numerous layers, thus the term "deep" networks.

DNNs are capable of modelling non-linear relationships that are complicated. Compositional models are generated using DNN architectures, in which the object is depicted as a basic layered composition. The extra layers allow the compilation of lower-level attributes, possibly modelling detailed information with lower units than an external network of equivalent performance (Kim, Han, Wang and Kim, 2022). For example, it has been demonstrated that DNNs are exponentially easier to estimate than external networks when dealing with sparse multivariate polynomials.

Many variations of a few fundamental methods make up deep architectures. In various fields, each architecture is successful. However, unless various designs have been assessed on similar data sets, comparing their effectiveness is not viable.

DNNs allow data to flow from the input to the output layers without looping back. The DNN starts by creating a map of virtual neurons and assigning random integer values, or "weights," to their connections. Next, the inputs and weights are multiplied, resulting in a value between 0 and 1. Of course, an algorithm would alter the weights if the network didn't detect a pattern correctly. As a result, the algorithm might increase the influence of specific parameters until it finds the optimal arithmetical manipulation to analyse the input completely.

Recurrent neural networks (RNNs) are employed in language modelling applications because input may flow in either direction (Diao, Ding and Tarokh, 2019). So it is when long short-term memory comes in handy (Chung and Shin, 2018). Therefore, convolutional deep neural networks (CNNs) are employed (Khan, Rahmani, Shah and Bennamoun, 2018). Acoustic modelling for automated speech recognition (ASR) has also used CNNs (Waris and Aggarwal, 2018).

1.6 Motivation and Problem Statement

Adults aged 60 and up are prone to falling. In India, one-third of the older people living in houses and half of the population living in nursing homes fall once a year. Various factors increase the chance of falling into old age. Movement issues, balance abnormalities, chronic diseases, and vision impairment are among them. Several falls result in a minimum of some injuries. Mild abrasions to shattered bones, and head trauma, are all possibilities. Fall risk prediction is used to evaluate whether the elderly are at a low, moderate, or high risk of falling. If older people are discovered to be at greater risk, their physicians and caregiver may recommend ways to help them avoid falls and injuries. Falls are the primary cause of death for older people, resulting in severe or even fatal injuries. Therefore, it is critical to creating reliable fall prediction systems to overcome this issue.

In this case, machine learning (ML) and deep learning (DL) algorithms are employed to forecast the likelihood of a fall. Most ML and DL algorithms are built for specific data collection or tasks. However, integrating ML and DL techniques to help compose, generalise, or unknown tasks can significantly enhance overall outcomes. As a result, a combined ML and DL technique is required to accurately forecast fall risk in the elderly.

This thesis proposes a Novel Stacking Classification and Prediction (NSCP) algorithm based on AAL with Multi-strategy Combination based Feature Selection (MCFS) and Novel Clustering Aggregation (NCA) algorithms to solve this problem.

1.7 Thesis Scope

The research work covers the following objectives:

- The thesis presents an effective feature selection algorithm for selecting the best features. Using selected features, the dataset's dimensionality can be reduced.
- In addition, the thesis presents a capable clustering algorithm to enhance classification efficiency and accuracy. An efficient and accurate classification and prediction algorithm is required to recognise the risk of older persons falling. A clustering algorithm is required before classification to improve the classification's performance.

- The thesis also provides an effective classification and prediction algorithm used for fall prediction. An open-source dataset is used to check the performance and effectiveness of the proposed work.
- A set of performance measures was utilised to assess the proposed work, compared with some previous algorithms.

The validation and experiments show that the proposed technique effectively predicted the fall compared with other methods. The algorithm, however, can yet be improved. The following are some suggestions for enhancing fall prediction performance.

- Maximise the dataset's size.
- Examine the possibilities of using alternative innovative and efficient feature selection and clustering algorithms.
- Explore the possibility to use other novel and classification algorithms for prediction.

1.8 Thesis objectives and hypotheses

The major goal of this thesis is to create a novel approach for predicting elderly falls and the processes, strategies, and algorithms that go along with it. In addition, the thesis presents feature selection and clustering algorithms to enhance prediction accuracy. The hypotheses that structure the research are:

H1 It is feasible to predict elderly falls by executing a Novel Stacking Classification and Prediction (NSCP) algorithm.

H2 It is possible to use the NSCP algorithm will utilize the training dataset to figure out how to best map incoming data instances to specified class labels.

H3 Multi-strategy Combination based Feature Selection (MCFS) algorithm was once utilized to lower the dataset's dimensionality.

H4 Novel Clustering Aggregation (NCA) algorithm can improve the efficiency and performance of NSCP.

H5 This method's guidance can increase the accuracy of the prediction of falls.

1.9 Research contributions

This thesis has implications for classifying and predicting elderly falls:

- A novel algorithm, namely NSCP using the Repeated Incremental Pruning to Produce Error Reduction (RIPPER) classifier, Multinomial Logistic Regression (MLR) classifier and Dl4jMlpClassifier are used as base learning techniques, and the Naïve Bayes classifier is utilized as a meta-classifier for elderly fall prediction;
- Optimal features from the dataset using the MCFS algorithm have been selected, which improves the data quality; MCFS uses the combination of 5 strategies. They are Information Gain, Fisher Score, Min-Max Normalization, Correlation Coefficient and Mean Absolute Deviation.
- Using the chosen features, the dimensionality of the dataset is reduced. The dimensionality reduced dataset is clustered using the NCA Algorithm, which combines three clustering algorithms, namely K-Means, Expectation-Maximization (EM) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN).
- Furthermore, the NCA algorithm adopts Majority Voting Technique for clustering.
- Based on these clusters, the accuracy of the NSCP algorithm could be increased;

- The NSCP algorithm has been used to create a prototype that incorporates all previous contributions and correctly classifies and predicts elderly falls.

1.10 Thesis Outline

The subsequent six chapters detail how the research's objectives were met. Chapter 2 provides a literature review related to the thesis. The chapter contains topics of AAL, elderly fall discovery using AAL, elderly fall detection using AAL and AI, security issues, trustworthiness and other benefits, and Issues around AI technology acceptance in AAL. This literary review has provided basic ideas on different topics and insights into various viewpoints presented by various specialists. The chapter concludes by stating that there are no appropriate mechanisms for the NSCP in fall prediction.

The proposed feature selection algorithm, MCFS, is described in depth in Chapter 3 for increasing data quality and picking the best features from the dataset. The proposed feature selection algorithm for older people's fall prediction is explained in this chapter, and all of the advantages of the proposed algorithm.

Chapter 4 discusses the NCA clustering algorithm by a variety of measurements. This chapter further provides the Majority voting technique for NCA clustering. Furthermore, the chapter uses the dimensionality reduced dataset with optimal features. The chapter concludes with creating novel clustered datasets utilizing the NCA algorithm.

Chapter 5 explains the elderly fall prediction algorithm utilizing the classification concept. The proposed NSCP algorithm is discussed in this chapter. ML and DL techniques for elderly fall prediction and classification details are further described. This chapter has attained the procedure of capably classifying the monitored parameters as (0- no fall detected, 1- person slipped/tripped/prediction of fall and 2- definite fall).

The proposed algorithm's performance is evaluated in Chapter 6 through experimental findings. The experiments were carried out using a set of sample data, and the input/output using these data was discussed. In this chapter, the findings are discussed, and their conclusions are documented.

Chapter 7 concludes and summarizes the thesis, highlighting important contributions and making recommendations for future work related to this research. The chapter further argues how the NSCP algorithm could also be extended to predict other functions using the research performed. Figure 1.7 demonstrates the thesis structure.

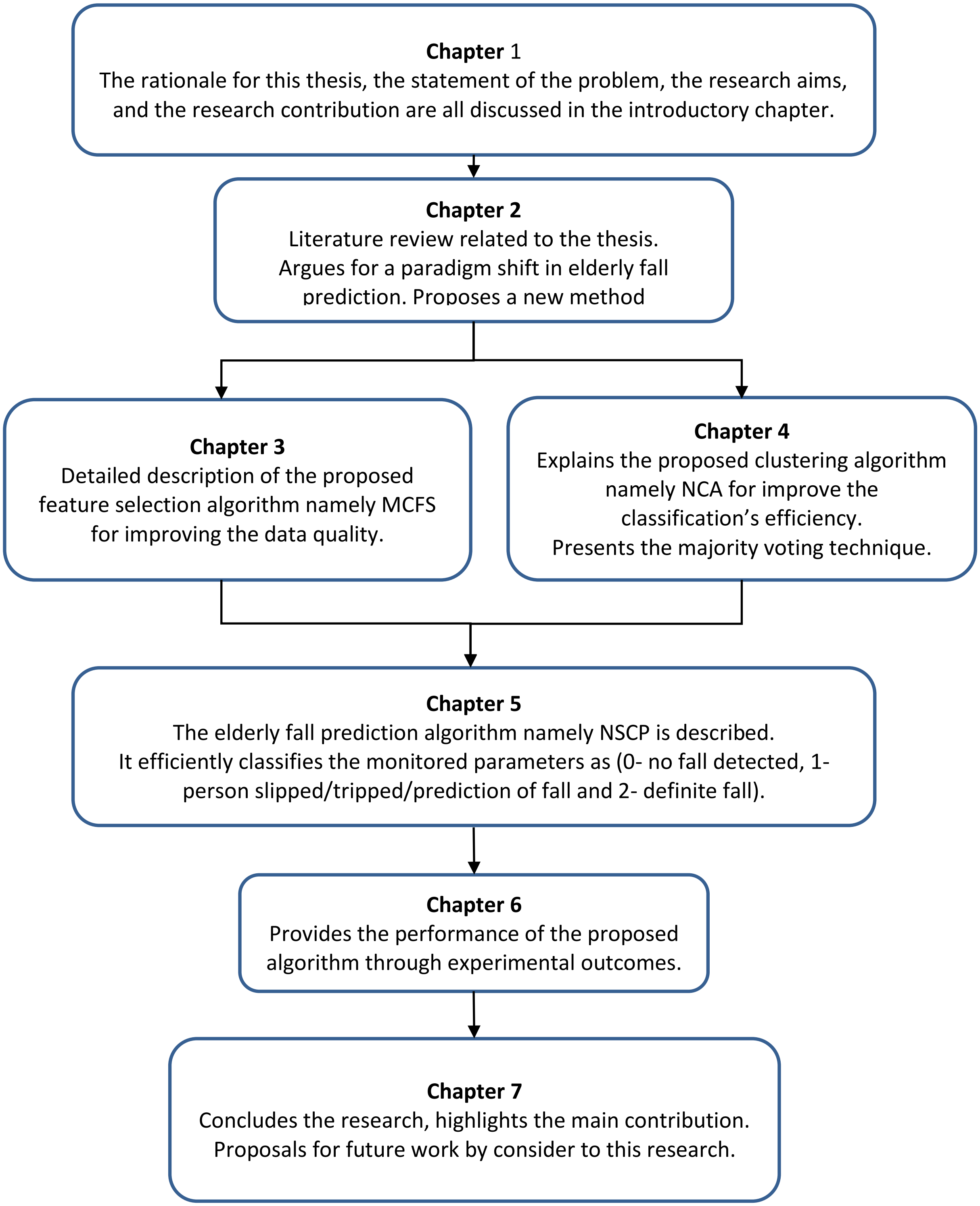

Figure 1. 7: Structure and relationships between thesis chapters

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Falls are a major cause of injury among the elderly, accounting for 40% of overall injury-related deaths (WHO, 2022). Falls are a major source of anxiety, loss of confidence and independence among the elderly. Every year, one in every three older people falls. As the time between the fall and rescue rises, the risk of mortality increases. Therefore, following the identification of a fall, a quick response by instantly reporting to caregivers is critical. As a result, a reliable fall detection method with fall alert approaches that include time and location in the notifications is quite beneficial.

This chapter examines a survey of several fall detection algorithms using various sensors. It comprises sensors such as an accelerometer, which measures a device's tilting motion and direction; a gyroscope, which adds dimension to the data presented through the accelerometer through monitoring rotation or twisting; a camera; a contrast vision sensor, and so on. The main benefit of fall detectors is that they can decrease the amount of time that the elderly spend lying on the floor after falling. It is one of the most important criteria in determining the seriousness of a fall.

Context-aware systems and wearable devices are the two basic categories of fall detectors. Context-aware systems detect falls by placing sensors in the context. Cameras, floor sensors, pressure sensors, and infrared sensors are the most often utilised in context-aware systems. The real benefit of context-aware systems is that no particular gadget is required to detect a fall. However, their functions are restricted to areas where sensors have already been installed (Batista, Moncusi, López-Aguilar, Martnez-Ballesté, and Solanas, 2021).

Wearable devices are little electronic sensor-based devices worn beneath, under, or on top of clothing by the user (Henderson, Condell, Connolly, Kelly and Curran, 2021). The majority of wearable fall detection systems are accelerometer devices. Several of them use a mix of sensors, like gyroscopes, to gather data on the patient's spot to build a precise fall detection method. Inexpensive built-in sensors in smartphones are the key benefit of utilizing a smartphone like a wearable device.

This chapter surveys the related work of elderly fall detection. It summarizes the topics of AAL, elderly fall detection using AAL, elderly fall detection using AAL and AI, privacy concerns, and other challenges and Issues around fall detection are presented. It describes the comprehensive study of various fall detection algorithms and is provided.

2.2 Ambient Assisted Living (AAL)

One of the great successes of the twentieth century was the increase in life expectancy. It is a continuing trend, with life expectancy in industrialised regions expected to reach 83 and 75 years in developing regions by 2045-2050. (Naja, Makhlouf and Chehab, 2017). Furthermore, fertility in many developed nations has been declining for some time. By that means, natural population growth rates have slowed or stopped altogether. Population ageing is caused by an increase in the average age and a reduction in the birth rate, which poses several challenges:

(i) Human capital declines when the working-age population declines, potentially lowering productivity.

(ii) Social Insurance Pension Schemes are susceptible to becoming overburdened.

(iii) An increasing population will require long-Term Care for the elderly. If present use rates are maintained, the number of persons who use such services will double by 2040 (Miranda, Mendes, and Silva, 2016), resulting in increased public investment.

(iv) Even with financial stability, the population in demand of care facilities will grow far faster than the working-age population, making it impossible to provide the necessary services.

From a technology standpoint, the impending shortage of caretakers and the great wish of the vast number of older individuals to stay in their own homes and communities has sparked a resurgence of interest in Ambient Assisted Living (AAL) (ElHady, and Provost, 2018). AAL refers to technical solutions that assist individuals in their everyday activities in maintaining their independence and safety for as long as possible. In addition, AAL solutions frequently emphasize the demands of particular interest populations other than the elderly, such as those with impairments or who require temporarily support (Offermann-van Heek and Ziefle, 2018).

The utilization of Ambient Intelligence (AmI) technology to empower individuals with particular needs has been characterised as the fundamental purpose of AAL (Dunne, Morris, and Harper, 2021). This section intends to provide an outline of the advanced features of AAL systems and a description of how current AAL systems work. In addition, this section gives an outline of a few of the AAL systems that are present.

2.3 Anatomy of AAL systems

The sense-act/interact loop provided in Figure 2.1 is commonly used in AAL systems.

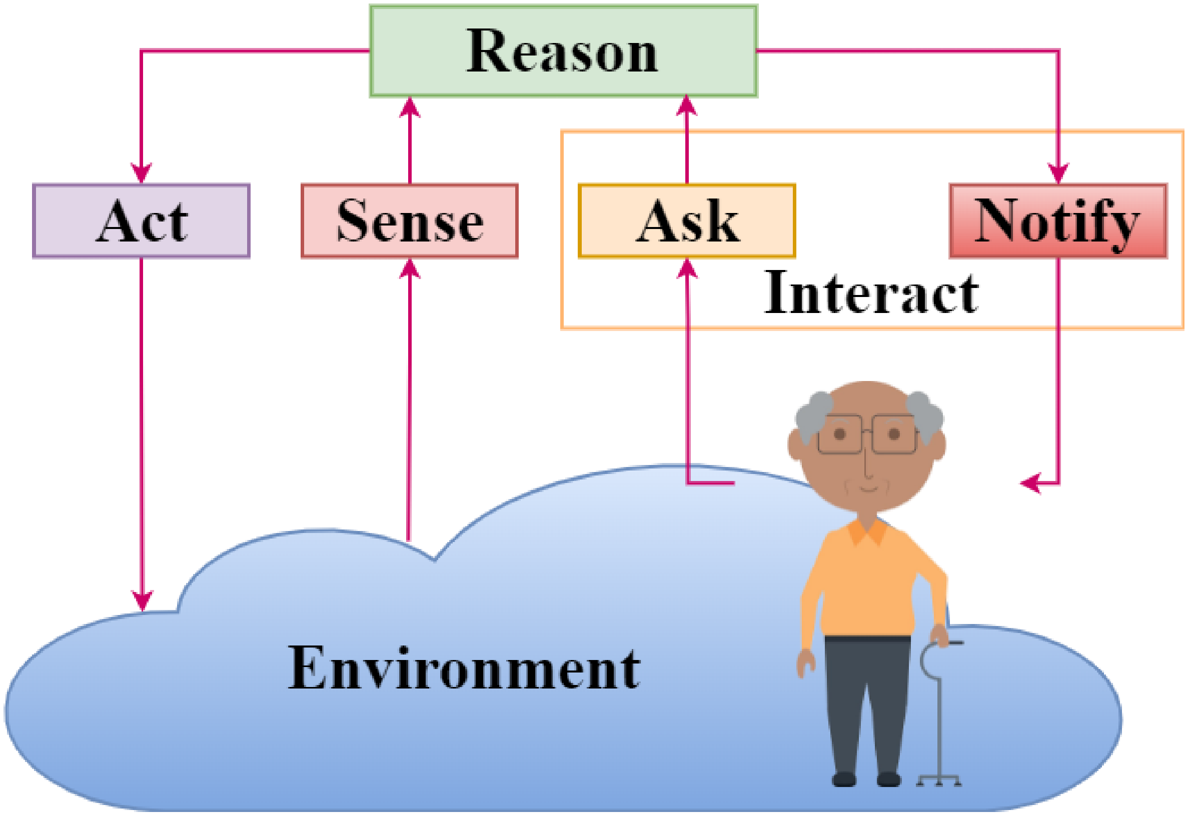

Figure 2. 1: Capabilities in AAL systems

The operations of Sensing and Asking acquire data from the context and what the consumers want. Through the Acting and Notifying features, the reasoning is responsible for understanding acquired data and acting on the context and the user. The user could be considered an integral component of the context, with knowledge about him collected via Q&A or observation abilities. Lastly, each action depends on communicating techniques, illustrated as pink arrows in Figure 2.1, to interact.

2.3.1 Sensing

An AAL system's primary ability is sensing. Sensors collect data on the settings and the people that live in them. Processing and transmission abilities are routinely added to sensors. Smart sensors are a subset of smart objects, self-contained cyber-physical objects with sensing, processing, recording, and networking abilities (Berger, Häckel, and Häfner, 2021). Wearable and environmental sensors are the two main sensors used in AAL systems.

Wearable Sensors. Wearable sensors are linked to the human body either implicitly or explicitly. They normally keep track of a person's physiological condition and their position and motions. A broad range of parameters about a person's physical condition could be gathered via various sensors shown in Table 2.1, for instance:

Table 2. 1: Various Sensors with Measurements

Sensors	For Measure
Thermistors	Tympanic, cutaneous, oral, and rectum temperatures
Sphygmomanometer cuff	Blood pressure
Capnograph	Carbon dioxide
Pulse oximetry devices	Oxygen saturation
Electrocardiography	Electrical activity of the heart
Chemical sensors	Blood chemistry

A person's position and motions are generally used to identify and classify ADLs and fall diagnoses (Kerdjidj, Ramzan, Ghanem, Amira, and Chouireb, 2020). The very general monitored parameters are:

(i) The resection procedure, which uses distances calculated from satellites, is commonly used to obtain an outdoor position using GPS (Global Positioning System) devices.

(ii) Radio Frequency Identification (RFID) is the most common method for detecting and identifying a person.

(iii) Magnetometers, tri-axial accelerometers, and angular rate sensors commonly determine body position and movement.

Environmental Sensors. The environment has environmental sensors implanted in it. They are primarily used to monitor ecological conditions or communications between users and the surroundings. In this domain, research is typically separated into video-based and non-video-based solutions.

AAL Solutions is based on the video. Due to the tremendous adaptability of cameras, vision-based solutions for AAL uses (VAAL) are an important topic. Activity recognition in healthcare and rehabilitation and fall detection are two the highly researched fields (Wang, Cang, and Yu, 2019). Utilizing video technologies to identify and analyze physiologic data is a powerful creative method. The lack of privacy is the real worry with the use of VAAL (AL-OBAIDI, 2020). Furthermore, possible users and their families must approve those solutions, which may raise issues even in applications that seek to protect privacy (Ravi, Climent-Pérez, and Florez-Revuelta, 2021).

AAL Solutions based on Non-Video. Sensors in this group often monitor merely some parameters, so they're frequently grouped jointly. The following are some instances of sensed parameters:

(i) Photodiode-based sensors are commonly used to assess ambient light.

(ii) Thermistors are used to convert room temperature to body temperature.

(iii) A Relative Humidity (RH) sensor is used to detect humidity.

(iv) Passive Infrared Sensors (PIR) are used to detect movement and presence.

(v) A magnetic proximity switch using reed components is typically used to open or close a door, window, or cabinet.

(vi) Force sensing resistors could be simply connected to smooth surfaces; for example, chairs are used to measure pressure, defined as the force exerted on a surface.

(vii) Microscopic sensors detect environmental sounds. Electret microphones, a capacitor microphone that does not require a continual supply of electrical charge to work, are the most extensively used. Microphones could be utilized as a presence sensor (similar to PIRs) or to locate an acoustic source localization (Guo, Zhu, and Dang, 2020). The ability to pinpoint the sound source can be useful for indoor location or fall detection (Alam, Faulkner, and Parr, 2020).

(viii) Odours convey a wealth of data about the surroundings. Many scientists have been working on building olfactory sensors that can detect and discriminate odours in recent times (Ward, Jjunju, Griffith, Wuerger, and Marshall, 2020)

Environmental sensors avoid the fundamental drawback of wearable sensors by not needing consumers to wear them all of the time. They do, but they have disadvantages (apart from confidentiality and acceptability concerns): their price is greater, they need installation (and therefore, related expense), and they are set in place, so they only work when the user is at home.

Sensor Technology Trends. As wearable sensors lose their functionality if they are not worn, researchers are focusing on reducing their size and weight and improving their longevity and waterproofing. Microelectromechanical Systems (MEMS, also known as micro-machines in Japan and Microsystems Technology (MST) in Europe) is a cutting-edge technique involving the miniaturisation of electromechanical, mechanical elements utilising microfabrication technologies.

Miniaturisation has allowed implantable and ingestible sensors, typically employed in qualified medical settings. Ingestible sensors are sensors that are incorporated with ingested devices like pills. They're designed to be fuelled by the human body and interact via tissue. These sensors could track consumed food, weight, a variety of physiological data, and body posture and behaviour, allowing users to maintain good habits and physicians to provide very quality care (Weitschies, Müller, Grimm, and Koziolek, 2021). After being implanted, implantable sensors could track and send data on the surgical healing site's load, temperature, and pressure.

2.3.2 Reasoning

The reasoning is transforming data collected in the field into qualitative data, which might take on numerous meanings depending on the user's context (— for example, 12 o'clock (afternoon) can signify 12:00, daytime, mid-day, etc.). Personal Context is described as user-specific situational data, such as aspects of the surroundings (— for example, items, offerings, and other people) accessed by the consumer; physiological (— for example, pulse and weight) and psychological states (— for example, mood and anxiety); tasks being conducted; social factors of the current user (— for example, friends, coworkers); and the Spatio-temporal features of other context elements from the users perspective (Ionel, 2022).

Data gathering and analysis, activity detection, modelling, and forecasting, predictive modelling, and Spatio-temporal reasoning are the primary reasoning features. Several reasoning modules using distinct features could be integrated into a solitary application. Artificial Intelligence (AI) could aid in developing higher executing components, allowing for the creation of the most helpful systems.

Data gathering and analysis. Data obtained from sense activity is normally simple to gather and interpret, but the sheer volume of such data, particularly when sound/video data is incorporated, is a difficulty. It's critical for AAL systems to be capable of gathering and integrating data from a variety of sensors and resources to understand events and circumstances and, as a result, determine contexts and states. Sensor data fusion is a technique described as merging data to improve state estimations and forecasts (Ghayvat, Mukhopadhyay, Shenjie, Chouhan, and Chen, 2018).

Recognizing, modelling, and forecasting activities. In AAL, reasoning techniques must be capable of comprehending contexts and presenting states not just through a static set of rules yet further through reactive and dynamic designs that take into account complex data (-for example, User behaviour patterns). They must also be capable of retrieving pertinent data and upgrading similar models. Reinforcement learning (i.e., learning from global findings), training to learn (that is, learning from prior observations), formative learning (that is, learning from the global investigation), and e-Learning (that is, learning from the Web and information technology) are all capabilities that AAL systems should have (Kumar Basak, Wotto, and Belanger, 2018).

The capacity to comprehend user behaviours is one of the most important contributions of reasoning algorithms. Template matching methods (Dang, Min, Wang, Piran, Lee, and Moon, 2020), generative approaches (Soleimani and Nazerfard, 2021), decision trees (Jobanputra, Bavishi, and Doshi, 2019), and discriminative approaches (Jobanputra, Bavishi, and Doshi, 2019) are some of the methods for recognising behaviours (Beddiar, Nini, Sabokrou, and Hadid, 2020).

User behaviour modelling and activity recognition are critical for forecasting likely statuses and contextual consequences. This quality is required to forecast potential undesirable occurrences and situations, act to prevent them, forecast user wishes, and boost their happiness.

Although typical activities play an essential part in health applications, unusual occurrences are also crucial since they frequently signal a crisis or a sudden shift in a routine linked to health problems. Classifiers can recognise both usual and unusual behaviours, and they normally need to be trained using datasets including instances of the actions to be identified. But, datasets covering behaviours connected to life-threatening circumstances (like falls) are uncommon. As a result of these factors, anomaly detection in AAL is gaining popularity (Ghayvat, Mukhopadhyay, Shenjie, Chouhan, and Chen, 2018).

Decision Support System (DSS). DSS is a broad word that refers to any computer programme that aids in decision-making. DSSs have become broadly used in the healthcare industry, supporting doctors and other specialists by assessing patient records (Moreira, Rodrigues, Korotaev, Al-Muhtadi, and Kumar, 2019).

Reasoning in terms of space and time. Knowing the present situation requires the ability to reason in geographical and temporal dimensions. A smart house system, for instance, can detect when someone switches on a cooker and leaves it alone for more than 10 minutes; if this occurs, the system takes action by switching off the cooker independently and alerting the user. A variety of ideas have been presented to facilitate Spatio-temporal reasoning in AAL situations (Chua, Foo, and Guesgen, 2020).

2.3.3 Interacting

Under the scope of Human-Computer Interactions (HCI), interaction is a well-studied subject that comprises all types of tools, both hardware/software, that enable the communication procedure between the system and the user (Costa, Barcellos, de Almeida Falbo, Conte, and de Oliveira, 2022). When building an AAL system, extra attention must be paid to the interaction behaviour. It has been noted that AAL systems that are hard or unusual to employ for people, particularly the elderly, would go underutilised.

HCI can be either implicit or explicit. The user uses explicit HCI (eHCI) when they query the system for anything. Invisible computers, vanishing interfaces, and ambient intelligence in common are all in direct opposition to this type of engagement. A conversation between a system and the user is always required in eHCI, and this dialogue places the computer at the core of the user's activities.

The system captures implicit input (that is, human activities completed to attain an objective, not mainly considered as communication with a computer). In addition, it might provide implicit output (that is, the outcome from a computer that is not directly associated with direct input and easily incorporated with the user’s surroundings and task) (Costa, Barcellos, de Almeida Falbo, Conte, and de Oliveira, 2022). The underlying premise is that the computer could detect users' interactions with their actual surroundings and, as a result, predict their objectives.

In the direction of natural interactions. The study of major topics in human interaction serves as a springboard for developing new shapes of HCIs. Three key ideas have been recognised as crucial for improved interaction:

(i) *The knowledge that is shared.* A shared knowledge base is vital in human relationships; it is frequently broad but not explicitly stated. Any human communication assumes shared knowledge, which generally involves a comprehensive globe and linguistic model, which is intuitive for people but difficult to understand properly.

(ii) *Errors in communication and recovery.* Almost no communication is error-free. Small misconceptions may arise during interactions; nevertheless, in regular dialogue, the speakers resolve these difficulties via repetition. It is thus natural to rely on the capacity to notice and correct communication problems in human interactions. On the other hand, such powers are less common in undetectable interactive computer systems.

(iii) *Situation and surrounding circumstances.* The context (that is, the environment and circumstances that guide interaction), which offers a general ground that develops implicit norms, has a significant impact on the meaning of terms and how human interaction is carried out.

When comparing how humans communicate with machines to how people engage with one other, it's evident that HCIs are yet in their infancy. One of the notions on which the area of Context-Awareness Computing is founded is that what humans anticipate from communications depends on the circumstance (Chavhan, Chavhan, Shafi, and Mahalle, 2021).

Interaction in the AAL area. The accessibility and acceptance of any technology solution from the standpoint of end-users is one of the most significant factors in its efficiency. This is especially true in AAL since most present and prospective end-users of an AAL system are people who have little or no empathy for technology. Designers must adhere to accessibility standards to create effective interfaces for AAL services. The Technology Acceptance Model (Grani and Maranguni, 2019), the Unified Theory of Approval and Usage of Technology (Baishya and Samalia, 2020), and the Usability Theory are the three most essential theories (Abdul Hamid and Mustopa, 2020).

2.3.4 Acting

Although the parameters impacted via the actuation might not usually be tracked by sensors, and not each sensed parameter may be altered by the actions, adding acting abilities to an AAL system may be considered the equivalence of a Closed Loop Control System at Control Theory.

While sensors are necessary for understanding and monitoring the real world, actuators are mechanical things that operate in the real world due to a software system operation. The quantity of various sensors accessible far outnumbers the number of actuators available. On the other hand, some key actuators are enough to create a vast number of complicated smart objects.

Most homes already have the most basic and simple actuators, but they are almost often solitary systems (Indoor lighting and air conditioning (AC) systems). The first effort to make lighting systems more context-aware was made by linking light bulbs with motion sensors (PIRs): In this manner, the lights did not need any explicit communication to turn on or off; however, the user's motion is an implicit input that causes the lights to turn on or off.

Actuators. As previously stated, AAL systems employ many generic actuators as building blocks. Here are several instances:

(i) *Relays* are electromechanical devices that operate as remote switches and are controlled via a software program by a low-power signal.

(ii) *MOSFETs* (Metal-Oxide-Semiconductor Field-Effect Transistors) are switches that are transistors. MOSFETs are often much smaller than relays, and some devices could switch nearly ten magnitudes quicker than relays. Magnetic fields, static electricity, and heat, on the other hand, may readily shatter them. They are typically used to control low-amperage devices (— for example, to turn on/off led lights, servos, and motors).

(iii) *Lights* were one of the first actuators to be added to the AmI system. Modern AAL lights generally have dimmer capabilities, several light colours, and a tiny microcontroller that handles communication. Most current lighting systems use Light Emitting Diodes (LEDs) that can quickly warm up to maximum brightness.

(iv) *DC* (Direct Current) motors are often utilised. Garage doors, drapes, and wheelchairs all use them. A DC motor turns electrical power into mechanical power. Stepper motors are frequently used to enhance accuracy. Servo motors, which are electric motors that could press or spin an item with remarkable accuracy, are another popular type of electric motor. Servo motors are frequently used for accurate, tiny motions requiring large force.

(v) By converting electrical information into physical phenomena, light emissions, and sound waves, *screens and speakers* give suggestions or data.

(vi) Although *haptic feedback engines* have been around since 1968 (Miall, Rosenthal, rstavik, Cole, and Sarlegna, 2019), they have only recently been studied in AmI solutions. Tactile feedback is presented through Haptic Interfaces (skin perception of pressure and temperature). It's a technology that works in conjunction with video and sound channels. The next phase in haptic interfaces for Virtual Reality is force and positional feedback, which could further offer data on strength, mass, pressure, and structure.

2.3.5 Communication

AALs are often made up of scattered devices collaborating to offer the needed services; communication skills are critical. AAL systems explore three main types of networks:

(i) When an AAL system requires sending external data side the system, it uses WANs. Nowadays, most systems rely on an Internet connection provided by several suppliers accessible. Recognition and tackling have been the greatest researched topics as the number of devices linked to the Internet has grown, resulting in IPv6.

(ii) Within residential systems, LANs are utilized. They track a variety of technologies, including wired connections, power-line communications, and wireless LANs (WLAN). Home automation frequently uses specialized buses, which necessitates the usage of gateways to connect home automation technologies to the remaining portion of the AAL framework.

(iii) The growing usage of smart wearables has given rise to BANs (Elhayatmy, Dey, and Ashour, 2018). Sensors and actuators (primarily haptic, audio, or optical) are attached to garments or straightly on the body in a BAN, with under-the-skin implants being used less commonly. Intra-BAN (communication inside the BAN), inter-BAN (the connection between Access Points and body sensors), and beyond-BAN (streaming body sensor data to urban areas, for instance, to a remote database where the users' profiles and health history are recorded and accessible to professional caregivers) are the three communication layers that define BANs.

2.4 Previous Platforms for AAL

Several frameworks have been presented in the existing; CASAS, short for Center for Advanced Studies in Adaptive Systems, was one of AAL initiatives' early and more common uses. Its purpose is to create a portable, expandable smart home kit with a set of critical features (Liciotti, Bernardini, Romeo, and Frontoni, 2020). As intelligent agents in CASAS settings, their condition (and their inhabitants) is determined by various contextual sensors. Activities are made with the help of controllers to improve the inhabitants' convenience, security, or/and efficiency.

The Physical layer handles sense and activity behaviours. The Middleware layer handles interaction using the publish/subscribe model and the Application layer, which contains programmes that purpose on the data supplied through the middleware. Other ideas are more straight targeted at the phenomena of population ageing, especially to the elderly.

The iNtelligent Integrated Network for Aged People (NINFA), for instance, is a project that focuses on the users' well-being. The goal is to create a service platform tailored to the needs of senior citizens, with a user interface that permits the delivery of various services at home, like user oversight, the interaction between users for social incorporation, exergame delivery, and common wellness tracking (Cinini, Cutugno, Ferraris, Ferretti, Marconi, Morgavi, and Nerino, 2021). Discourse and conversation analysis is used to track the linguistic activity of persons suffering from various diseases to get an initial diagnosis (e.g., aphasia, traumatic brain injury, and dementia).

Furthermore, the system uses HCIs to give a series of custom-designed exergames for older or motor-impaired patients to undertake motor/cognitive assessment. ROBOCARE is another method aimed at preventing adult disorders. Sensors, robotics, and other expert systems help users in the

ROBOCARE method. ROBOCARE is an instance of an AAL solution that investigates the benefits and drawbacks of incorporating helpful and social robots into systems.

It's built around a mobile robot, a surroundings stereo camera, and a wearable activity tracking module. The system uses automatic reasoning to decide if the user's behaviours fall into predetermined and legitimate patterns using the observations received through the camera and the wearable device. Caregivers create such patterns while also considering the user's medical condition.

Other solutions to assist people with unique requirements, independent of their age, are also available. The BackHome project, for example, is concentrated on building, executing, and assessing person-centred solutions for people with disabilities. In addition, the research aims to investigate how brain-neural computer interfaces and other assisting techniques could aid experts, consumers, and their families in transferring from hospital to home care.

BackHome's primary purpose is to assist end-users in achieving tasks that would else be difficult or need a caregiver (Casals, Cordero, Dauwalder, Fernández, Solà, Vargiu, Miralles, 2018). The project's output is a telemonitoring and home-assistance system (Bhatti, Siyal, Mehdi, Shah, Kumar, and Bohyo, 2018).

Innovative wearable technology platforms. A recent study on smartphones and wearable devices is paving the way for novel ways to collect data on health and illness. Apple has given Research Kit (RK), a free software platform for clinical trials that allows investigators who create iOS apps to acquire important information for their research from everyone who uses RK-based apps (Researchkit, 2022). Furthermore, as individuals use and engage with their devices, data will become more readily obtainable. Some examples of RK-based apps and research will be offered in the following.

mPower. The mPower app is medical observational research on Parkinson's disease that uses an app interface. The app gathers data from users with and without Parkinson's disease via questionnaires and regular sensor-based observations. The eventual objective is to use this actual data to quantify the peaks and valleys of Parkinson's symptoms (Maetzler and Pilotto, 2021).

Autism & Beyond. Autism & Beyond is a research project that attempts to test novel video technologies to assess a child's attitude and activity. The app displays four brief video clips while recording the child's attitudes to the videos intended to create them grin and be amazed. The assessment module then highlights significant landmarks on the child's face and evaluates their behavioural replies after the learning. The purpose isn't to deliver at-home treatment but rather to investigate if this method is effective sufficient to collect relevant data (Duke University, 2022).

EpiWatch. EpiWatch tracks seizures and suspected causes and drugs and adverse effects to assist users in handling their epilepsy. Sensors and questionnaires capture data on the activities done and the user's status before and after the attacks and notes on clinical adherence. (2022, Johns Hopkins University)

Cardiogram. Cardiogram uses deep learning technologies in cardiology to discover abnormal heart-rate variability patterns and investigate atrial fibrillation, the very general cardiac arrhythmia. Data is gathered from persons with heart disease and healthy people (Apple Watch, 2022).

2.5 AAL for fall detection

This section summarises recent research on ambient assisted living technologies and allied regulation, focusing on fall detection. Assisted living technologies can be divided into three generations (Ganesan, Gowda, Al-Jumaily, Fong, Meena, and Tong, 2019).

The first generation of technologies comprised assistive systems and gadgets, which only worked when the user requested or responded to aid. A wearable gadget with a panic or help button to ask for assistance is an example of first-generation AAL. In a crisis, such as a fall, a person using that device might press a button to request assistance from a guardian or medical expert. The main disadvantage of these systems was that they could not function without a response or request from the user. Because elderly persons may occasionally be unable to hit the help button due to a fall, such a wearable device would be useless. As a result, the development of the second generation of AAL became necessary.

A unique feature is the ability of second-generation technology to detect when a user needs help. These devices kept track of health-associated, user behaviour-associated, and user communication-associated data to send alarms to carers or medical personnel. On the other hand, second-generation technology had its own set of constraints. Initially, they couldn't avoid crises because the alerts were only activated after the user had experienced one. Next, there have been reports of false positives, and lastly, older persons have perceived that the systems are overly intrusive.

To address these constraints, the third generation of AAL has just emerged. These are smart assistive systems that employ a variety of technologies, for example, artificial intelligence, machine learning, and deep learning, to identify and forecast any assistance requirements, such as in the event of a fall.

Third-generation systems are capable of not just detecting and reporting issues but also proactively attempting to avert issues and crisis scenarios. The first activity involves tracking the user's vital signs, and any ultimate alter in his movement and activity patterns, forecast continuing, alters in health condition; the second activity involves restricting the user's publicity to huge risk situations based on actions taken actuators.

Fall detection systems illustrate the three stages of AAL system evolution: initial concepts were passive and dependent on user activities; current solutions are independent and proactively discover falls; and lastly, the most creative techniques are geared toward fall forecasting and prevention.

There are two types of risk variables that influence the likelihood of falling: intrinsic and extrinsic risk factors. Age, poor mobility, bone frailty, low balance, chronic illnesses, cognitive and dementia issues, Parkinson's disease, vision issues, mind-altering medicines, inappropriate lifestyle (inactivity, use of alcohol, and obesity), and last falls are all intrinsic risk factors. Extrinsic risk factors include improper footwear and clothing, and medicine combinations.

Lastly, sliding floors, steps, and the desire to attain high things have been highlighted as environmental risk factors for indoor falls. Only 8% of those who had no risk factors fell, compared to 19% of those who had one risk factor, 32% of those who had two, 60% of those who had three, and 78% of those who had four or more risk factors (Stevens, and Phelan, 2013). Most prevalent technology solutions rely on smartphone accelerometers to detect and inform falls quickly. In addition, most approaches rely on domain knowledge algorithms, which are often based on experimentally determined thresholds. Machine learning approaches are used in more complex solutions, and many of them require fall data to train the classifiers adequately. Because real-life fall data is hard to get, these systems depend on simulated falls.

Then, based on these various generations of assisted living systems, we examine current developments in fall detection.

(Hakim, Huq, Shanta, and Ibrahim, 2017) they developed a smartphone-based system for fall detection. The authors used Matlab to create an SVM-based algorithm. The SVM classifier studied aspects of motion data to identify falls using data from the user's phone's designed inertial measurement unit (IMU).

(Espinosa, Ponce, Gutiérrez, Martnez-Villaseor, Brieva, and Moya-Albor, 2019) They created a neural network-based method for detecting falls. The method took data from numerous cameras and used an optical flow technique to extract various attributes. This optical flow approach supplied data on the relative motion between two sequential pictures by evaluating the picture data.

(Nakamura, Bouazizi, Yamamoto, and Ohtsuki, 2020) created a neural network-based fall discovery system. The researchers used spectrogram pictures and Wi-Fi CSI information to train this learning method to construct the fall discovery classifier.

(Balli, Sabaş, and Peker, 2019) employed a smartwatch to collect data for their fall detection system. A random forest model was used to evaluate and analyse the human behaviour and activity-associated data gathered through this smartwatch to identify falls.

(Dhole, Kashyap, Dangwal, and Mohan, 2019) created a prototype for gathering and interpreting EEG information to identify falls. The prototype looked like a helmet that the user wore, and the authors used EEG data to create a random forest classifier to identify falls.

A k-NN-based ML technique was presented by (Ramirez, Velastin, Fabregas, Meza, Makris, and Farias, 2021). The system uses computer vision concepts to identify falls by extracting features from human stances during various activities.

One more neural network-based fall discovery framework was presented by (Tahir, Ahmad, Morison, Larijani, Gibson, and Skelton, 2021), which enhanced the related computational prices devoid of compromising the fall discovery accuracy (Espinosa, Ponce, Gutiérrez, Martnez-Villaseor, Brieva, and Moya-Albor, 2019).

(Sarabia-Jácome, Usach, Palau, and Esteve, 2020) developed a system for investigating and detecting falls using a deep learning-based technique. The framework's design included two deep learning systems and virtualization principles for detecting falls.

Fall detection using video data-based behaviour analytics was presented by (Manekar, Saurav, Maiti, Singh, Chaudhury, Kumar, and Chaudhary, 2020). The task entailed evaluating the user's 360-degree video data to assess behaviours and monitor falls. (Ngu, Tseng, Paliwal, Carpenter, and Stipe, 2018) developed a fall discovery system using data from a wristwatch utilizing a naive Bayes technique.

(Khan, Qamar, Zaheen, Al-Ali, Al Nabulsi, and Al-Nashash, 2019) developed a fall detection wearable device. A camera, accelerometer and gyroscope interacted with a microprocessor on the device. The technique included developing a binary classification system to identify falls using a naive Bayes classifier.

For fall detection, (Ning, Zhang, Nie, Li, and Zhao, 2019) developed a gradient boosted trees technique. The system analysed human activity and postural information to identify falls throughout various activities.

One more gradient boosted trees-based learning solution (Cahoolessur and Rajkumarsingh, 2020) provided was a binary classifier that might identify falls and other user behaviours in a virtual IoT-based setting. The authors created a wearable device that the user wore around their waist; the ML technique was also implemented using a cloud computing-based framework.

(Cai, Qiu, Li, Yu, and Wang, 2019) employed a single-layer decision tree and an AdaBoost technique to identify falls using accelerometer data. To identify falls from other everyday activities, (Lee and Tseng, 2019) employed the same threshold-based technique. The method employed the user's phone's acceleration data and the threshold criterion to identify falls. Table 2.2 describes these associated researches in terms of the data sources or datasets utilised to build the related techniques and whether the information originated from non-wearable or wearable sensors.

Table 2. 2: Outline of current works on fall discovery and their data source

Work	**Data**	**Use of Wearable's**	**Use of Non-Wearable's**
(Hakim, Huq, Shanta and Ibrahim, 2017)	The user's phone's inertial measurement unit (IMU)	Yes	No
(Espinosa, Ponce, Gutiérrez, Martínez-	Data from several cameras	No	Yes

Villaseñor, Brieva and Moya-Albor, 2019)			
(Nakamura, Bouazizi, Yamamoto and Ohtsuki, 2020)	Wi-Fi CSI data and spectrogram pictures	No	Yes
(Balli, Sağbaş and Peker, 2019)	Information from a smartwatch on behaviour and motion	Yes	No
(Dhole, Kashyap, Dangwal and Mohan, 2019)	EEG data	Yes	No
(Ramirez, Velastin, Fabregas, Meza, Makris, and Farias, 2021)	Study of images of various poses	No	Yes
(Tahir, Ahmad, Morison, Larijani, Gibson, and Skelton, 2021)	Accelerometer data	Yes	No
(Sarabia-Jácome, Usach, Palau, and Esteve, 2020)	Data from resource-restricted devices (fog nodes)	Yes	No
(Manekar, Saurav, Maiti, Singh, Chaudhury, Kumar, and Chaudhary, 2020)	360-degree video data	No	Yes

(Ngu, Tseng, Paliwal, Carpenter, and Stipe, 2018)	Smartwatch performance data	Yes	No
(Khan, Qamar, Zaheen, Al-Ali, Al Nabulsi, and Al-Nashash, 2019)	Camera, gyroscope, and accelerometer data	Yes	Yes
(Ning, Zhang, Nie, Li, and Zhao, 2019)	Human behaviour and posture data	Yes	No
(Cahoolessur and Rajkumarsingh, 2020)	Human behaviour data	Yes	No
(Cai, Qiu, Li, Yu, and Wang, 2019)	Three axial acceleration, three axial angular acceleration	Yes	No
(Lee and Tseng, 2019)	Accelerated data collected from a smartphone, GPS data and WiFi signals	Yes	No

The next section discusses the recent works of Artificial Intelligence (AI) with AAL on fall detection.

2.6 Artificial Intelligence with AAL for fall detection

Artificial intelligence, also known as mechanical intelligence, may be comprehended by intelligence, unlike natural intelligence displayed by people and animals, as shown by machines. It considers how to develop smart devices and technologies that could creatively solve issues frequently viewed as people's privileges. Therefore, AI denotes a machine that simulates people's behaviour somehow.

Machine learning (ML) is a subset of AI that comprises technologies that allow computers to identify information and provide AI systems. In ML, several methods (for example, neural networks) assist in issue solving.

Deep learning (DL), known as a deep neural network, is a subset of ML that utilises neural networks to assess various characteristics using a framework similar to that of a human brain system. It has networks that could learn without guidance from unstructured or unlabeled input.

This section describes a few of the most common and contemporary fall detection technologies that use ML and DL technologies.

2.6.1 AAL based on machine learning for fall detection

Since threshold-based systems are common due to their cheap computing cost, they may be more susceptible to false positives and negatives because various circumstances could impact thresholds. Thereby, fall detection based on ML has received a lot of attention. The efficacy of several ML approaches for fall detection has been extensively researched. For example, compare Threshold-based and ML-based fall detection systems using data from an accelerometer, gyroscope and magnetometer (de Quadros, Lazzaretti and Schneider, 2018). The authors concluded that the ML-based technique outperformed threshold-based methods significantly.

Multiple elements distinguish ML-based methodologies, including the feature set, sensors used, sensor location, algorithms used, the dataset utilized, performance metrics tracked, etc. Research focuses on the appliance of supervised learning to fall discovery utilising wearable sensors, and the findings are promising.

The dataset utilised in (Vallabh, Malekian, and Bogatinoska, 2016) was created using an accelerometer and gyroscope put at the waist level. Windowing was used to extract features, feature selection was made with a rank-based method, and classification was done with the Nave-Bayes, LSM, ANN, SVM, and kNN algorithms. In comparison to LSM and Nave-Bayes, the performance of kNN, ANN, and SVM was the best. For kNN, the accuracy is 87.5 per cent, the sensitivity is 90.70 per cent, and the specificity is 83.78 per cent.

(Jefiza, Pramunanto, Boedinoegroho, and Purnomo, 2017) used a backpropagation neural network (BPNN) to detect falls using data from a 3-axis accelerometer and gyroscope, reporting an accuracy of 98.182 per cent, a precision of 98.33 per cent, a sensitivity of 95.161 per cent, and specificity of 99.367 per cent.

(Hossain, Ali, Islam, and Mustafa, 2016) evaluates SVM, kNN, and complex tree algorithms using the information provided via accelerometers to identify falls from ADLs. On ADLs and four kinds of falls, the authors compared the effectiveness of various algorithms in terms of accuracy, precision, and recall (forward, backward, right, and left). The accuracy and precision of SVM were the greatest, whereas complex trees outperformed in recall research. Sensors incorporated with mobile phones have also been subjected to ML algorithms similar to threshold-based approaches.

One of the reported limitations of wearable sensors is that the location of the sensors affects the accuracy of fall categorization and identification. The authors created a dataset using an accelerometer and gyroscope wear around the waist and used SVM, boosted and bagged decision trees, kNN, k-mean, and a hidden Markov model (HMM) (Rodrigues, Salgado, Cordeiro, Osterwald, Teodiano Filho, de Lucena, and Murray, 2018). It was discovered that fine kNN provided 99.4 per cent accuracy.

(Yu, Chen, and Brown, 2017) provide an HMM-based fall discovery technique for reducing errors caused by improper sensor locations. The positions of the misplaced sensor (3-axis accelerometer) and mismatched sensor orientation are resolved to utilize sensor orientation calibrations using HMM classifiers. On an experimental dataset, the author states a sensitivity of 99.2 per cent and 100 per cent for a fall dataset.

(Guvensan, Kansiz, Camgoz, Turkmen, Yavuz, and Karsligil, 2017) concentrate on fall detection's power efficacy. Data from a 3D accelerometer coupled with a smartphone was processed using threshold-based methods and ML-based K-Star, Nave-Bayes, and J48. The method included a pre-elimination layer that applied early filtering, a dual thresholding tier that identified hard falls and slow-moving physical activities and an ML tier that used ML technologies to distinguish fall and fall-like occurrences.

Compared to ML-only strategies, the hybrid model saved 62 per cent of energy while maintaining 93 per cent accuracy. The hybrid strategy outperformed the threshold-based and ML-based techniques in terms of sensitivity while being equivalent in terms of specificity. Numerous strategies are utilised to enhance the effectiveness of fall detection techniques, such as improving data pre-processing, impacting feature selection or extraction, and using ensembles to fall discovery.

The goal of (He, Bai, and Wang, 2017) was to differentiate between falls and ADLs. This study's wearable fall detection method consists of a smartphone and a wearable motion detector. Instead of assessing simultaneous acceleration values with angular velocities, the system analyses data streams using sliding windows. It uses a Kalman filter to reduce noise in the raw data and a Bayes network classification algorithm to detect falls. With an accuracy of 95.67 per cent, a sensitivity of 99.0 per cent, and a specificity of 95.0 per cent, the system was able to discriminate simulated falls from ADLs.

(Zhao, Li, Niu, Gravina, and Fortino, 2018) use a windowing approach to manipulate actual data from a triaxial gyroscope. The data was separated into a series of windows that were somewhat overlapping and sequential. The data windows yielded three-time domain attributes: resulting angle alteration, maximal resulting angular acceleration, with fluctuation frequency. Each window was then classified as a fall or non-fall event using a decision tree classifier. The detection system provides a 99.52 per cent accuracy, 99.3 per cent precision, and 99.5 per cent recall.

One more study (Chelli and Pätzold, 2019) examines the effectiveness of four algorithms in two steps: ANN, kNN, quadratic SVM, and ensemble bagged tree. To begin, just data on acceleration with angular velocity is employed. The spectral power distribution of the acceleration is then used to extract new features that increase the classifier's effectiveness. After using feature extraction methods, the algorithms' accuracy appears to have improved.

The goal of (Wang, Li, Li, Cao, and Wang, 2016) was to see how optimum feature selection affected fall detection accuracy. Wearable devices collect data on accelerations in various regions of the body. The outstanding features were selected from the information provided through the wearable devices using a Bayesian framework. The weight of all features was determined, following which training was completed using the optimum feature set. Compared with Nave-Bayes and C4.5 classifications, enhanced classification improved accuracy, sensitivity, and specificity.

(Tsinganos and Skodras, 2017) use accelerometer information to retrieve 14 attributes that span the time domain, statistical measurements, and continual wavelet transform. To identify falls from ADLs, ENN was used to eliminate outliers, and then a kNN classifier was used to train the classifier. Personalization was used to eliminate individual patterns by adding ADL attributes to the training dataset. ANN, SVM, and J48 decision trees were the other models compared. The greatest effectiveness was achieved by kNN.

The authors suggest EvenT-ML (Putra, Brusey, Gaura, and Vesilo, 2017) in that a fall event is matched with three steps of falls (pre-impact, impact, and post-impact) utilizing a finite state machine. Data from accelerometers were used in the experiment, and attributes were retrieved from each step. The classifiers were trained using classification and regression tree (CART), kNN, logistic regression (LR), and SVM. EvenT-ML outperforms the frequently utilized data segmented approaches of fixed-size overlapping sliding window (FOSW) or fixed-size non-overlapping sliding window (FNSW), which execute feature extraction on entire data blocks. The finite state machine guarantees that feature extraction occurs merely while the subject is engaged, which decreases the method's computing complexity.

The use of ensembles to fall detection has also been the subject of a recent study. To identify a fall event, (Hsieh, Liu, Huang, Chu, and Chan, 2017) combine a threshold-based and knowledge-based technique using SVM with data from a tri-axial accelerometer. Acceleration thresholds are used to identify relative falls and ADLs. When the maximum level of acceleration coincides, a knowledge-based technique is used to identify falls from ADLs. Sensitivity, specificity, precision, and accuracy were all higher than 99 per cent when using this method.

(Genoud, Cuendet, and Torrent, 2016) offer a solution for mild fall detection in wearable devices that use machine learning. Gyroscope and linear acceleration readings were utilised as feature sets, while decision tree, decision tree ensemble, kNN, and multilayer perceptrons were examined as algorithms (MLP). The outcomes of the studies exposed that the decision tree ensemble do superior to the other techniques.

(Kao, Hung, and Huang, 2017) employ a spectrum analysis ensemble that includes GA-SVM, SVM, and C4.5 classifiers. 3-axis accelerometers provided the sensor data. GA-SVM produced the greatest results, with 94.1 per cent accuracy, 94.6 per cent sensitivity, and 93.6 per cent specificity.

(Jahanjoo, Tahan, and Rashti, 2017) proposed a fall detection method using data from 3-axis accelerometers, which used PCA for dimension assessment and a multilevel fuzzy (MLF) min-max neural network. Results were compared with MLP, kNN, and SVM. MLF outperformed the other methods in terms of sensitivity using only five dimensions of data, although specificity was equivalent for all four methods.

(Hussain, Ehatisham-ul-Haq, and Azam, 2019) combine kNN, SVM, and RF algorithms to identify falls and further determine the pattern of falls and the activities that might have caused the fall. The maximum accuracy for fall discovery was recorded for kNN, whereas the highest accuracy for distinguishing various activities was given for random forests.

One more study (Liu, Hsieh, Hsu, and Chan, 2018) tries to identify a link between sampling rate and ML method performance accuracy. The authors evaluate the results of SVM, Naive-Bayes, and kNN and decision trees with different sensor sampling rates in this study. SVM and radial basis function, with sampling rates of 11.6 Hz and 5.8 Hz, respectively, produce accuracies of 98 per cent and 97 per cent. According to the findings, many ML models can achieve a 97 per cent accuracy with a sampling rate of 22 Hz.

(Hakim, Huq, Shanta, and Ibrahim, 2017) suggest combining threshold and ML-based fall discovery methods. A threshold-based method is employed to identify falls and categorise ADL, and a supervised ML technique is utilized. IMU sensors in a smartphone were used to collect data. For detection and classification, four distinct classification techniques were used: SVM, decision trees, kNN, and discriminant analyses.

The findings of their technique utilising SVM reveal that the mixture of angular velocity, acceleration, and orientation parameters is used instead of using them individually; activity recognition may be boosted by an accuracy level reaching higher than 99 per cent.

(Ali, Shrestha, Fioranelli, Le Kernec, Heidari, Pepa, and Spinsante, 2017) present data integration from a micro-Doppler radar, a tri-axial accelerometer, and a depth camera (Li, Shrestha, Fioranelli, Le Kernec, Heidari, Pepa, and Spinsante, 2017). The influence of sensor fusion on classifier performance is examined in this research. Compared to radar-only usage, this fusion strategy enhances classification accuracy by 11.2 per cent and by 16.9 per cent when compared to accelerometer use. It was further shown that combining data from three sensors improve classification performance by up to 91.3 per cent when using a quadratic-kernel SVM classifier and up to 86.9% when utilizing an ensemble classifier. Sensor fusion has been used in conjunction with DL algorithms in a few experiments to detect falls.

Table 2.3 depicts a summary of the current analysis of the use of ML to detect falls in wearable devices. All of the literature studies use public dataset analyses or fall simulations in controlled contexts. Where available, the table qualitatively compares several fall detection algorithms and highlights their performance criteria like accuracy, sensitivity, and specificity.

Table 2. 3: Wearable devices with ML-based algorithms for fall detection

Reference	Dataset utilized	Sensors utilized	Sensor positioning	Methodology	Performance that was observed
(Vallabh, Malekian, and Bogatinoska, 2016)	MobiFall dataset	Accelerometer, gyroscope	the pocket of the user's trousers	For fall detection, compare the Naïve Bayes, LSM, ANN, SVM, and kNN algorithms.	The most accurate models were k-NN, ANN, and SVM—results for kNN: 87.5 per cent accuracy, 90.70 per cent sensitivity, 83.78 per cent specificity
(Jefiza, Pramunanto, Boedinoegroho, and Purnomo, 2017)	Created from experiments	3-Axis accelerometer, 3-axis gyroscope	Waist	Fall detection using a backpropagation neural network (BPNN).	98.182 per cent accuracy 98.33 per cent precision

					95.161 per cent sensitivity
(Hossain, Ali, Islam, and Mustafa, 2016)	Created from experiments	3D accelerometer	Chest	SVM, kNN, and complex tree algorithms were used on accelerometer data.	SVM provided the best accuracy and precision. The complex tree provided the best recall.
(Rodrigues, Salgado, Cordeiro, Osterwald, Teodiano Filho, de Lucena and Murray, 2018)	Created from experiments	Accelerometer, gyroscope, magnetometer	close to the waist	kNN	99.4 per cent accuracy
(Yu, Chen, and Brown, 2017)	FARSEEING dataset	3-Axes accelerometer	The sensors are mounted on five upper body areas, particularly the neck, chest, hips, right side and left side.	HMM classifiers and a sensor orientation calibration technique to handle difficulties stemming	99.2 per cent sensitivity (experimental dataset)

				from missing sensor placements and misplaced sensor orientations	
(Guvensan, Kansiz, Camgoz, Turkmen, Yavuz, and Karsligil, 2017)	Created from experiments	3D accelerometer	Smartphone	K-Star, Naive Bayes, and J48 are ML-based algorithms that combine threshold-based and ML-based techniques.	62 per cent energy savings as compared to ML-only methods 77 per cent sensitivity (threshold only), 82 per cent sensitivity (ML only), and 86 per cent sensitivity (hybrid) 88.4 per cent accuracy (threshold only), 90 per cent accuracy (ML

					only), 92.75 per cent accuracy (hybrid)
(He, Bai, and Wang, 2017)	Created from experiments	Accelerometer, gyroscope	Vest	A Kalman filter for noise removal, a sliding window, and a Bayes network classifier is used for fall detection.	The accuracy is 95.67% and the specificity is 95.0% with the Kalman filter.
(Zhao, Li, Niu, Gravina, and Fortino, 2018)	Created from experiments	Triaxial gyroscope	Waist	Decision tree	99.52 per cent accuracy 99.3 per cent precision 99.5 per cent recall
(Chelli and Pätzold, 2019)	Real-world datasets	Accelerometer, gyroscope	Chest, thigh	Ensemble bagged tree (EBT), ANN, kNN, QSVM	The accuracy of all four classifiers was enhanced by extracting novel features from

					acceleration and angular velocity. EBT had the highest accuracy (97.7 per cent)
(Wang, Li, Li, Cao, and Wang, 2016)	Created from experiments	Accelerometer	Various parts of the body	Bayesian framework for feature selection, C4.5 and Naive-Bayes,	Higher accuracy with advanced classification than C4.5 and Naive-Bayes
(Tsinganos, and Skodras, 2017)	Created from experiments	Accelerometer (MobiAct dataset)	Not applicable	ENN + kNN (ENN used to remove externalities), ANN, SVM and J48	95.52 per cent sensitivity 97.07 per cent specificity 91.83 per cent accuracy for ENN + kNN
(Putra, Brusey, Gaura, and Vesilo, 2017)	Cogent dataset, SisFall dataset	3D accelerometer, 3D gyroscope-Cogent dataset Accelerometer, gyroscope-SisFall dataset	Chest, waist	Classification and regression tree (CART), kNN, logistic regression, S VM	Event-ML gives better accuracy and F-scores than FOSW and FNSW-based techniques

				and event-ML.	
(Hsieh, Liu, Huang, Chu, and Chan, 2017)	Created from experiments	3-Axes accelerometer	Waist	A mixture of knowledge-based and threshold-based techniques with SVM is used to identify a fall event.	Utilizing a knowledge-based algorithm: 98.74 per cent specificity
(Genoud, Cuendet, and Torrent, 2016)	Created from experiments	3-Axes accelerometer	Not mentioned	We compared decision trees, decision tree ensembles, kNN, neural networks, and MLP algorithms for soft fall detection.	At greater than 0.9 AUC, the decision tree ensemble was capable of identifying soft falls.
(Kao, Hung, and Huang, 2017)	Created from experiments	3-Axes accelerometer	Smartwatch	Combining spectrum analysis with GASVM, SVM, and C4.5 classifiers	GA-SVM produced the greatest results with an accuracy of 94.1 per cent, 94.6 per

					cent sensitivity and 93.6 per cent specificity
(Jahanjoo, Tahan, and Rashti, 2017)	MobiFall dataset	3-Axes accelerometer	Not mentioned	MLP, KNN, SVM, and PCA for fall detection are compared to multilayer fuzzy min-max neural networks, MLP, KNN, SVM, and PCA.	The best results were obtained using a multilayer fuzzy min-max neural network. 97.29 per cent sensitivity 98.70 per cent specificity
(Hussain, Ehatisham-ul-Haq, and Azam, 2019)	SisFall dataset	Accelerometer, gyroscope	Waist	kNN, SVM, random forest	kNN provides the greatest accuracy for fall detection (99.8 per cent). Random forests provide the best accuracy

					for detecting fall activities (96.82 per cent)
(Liu, Hsieh, Hsu, and Chan, 2018)	SisFall dataset, created from experiments	Accelerometer	Chest/thigh, waist	SVM, kNN, Naïve Bayes, decision tree	For both datasets, SVM provides the highest accuracy and sensitivity (97.6 per cent and 98.3 per cent, respectively).
(Hakim, Huq, Shanta, and Ibrahim, 2017)	Created from experiments	Accelerometer, gyroscope, proximity sensor, compass	Right, left, and front pockets	SVM, decision tree, kNN, discriminant analysis	SVM has a better accuracy of 99 per cent.
(Li, Shrestha, Fioranelli, Le Kernec, Heidari, Pepa, and Spinsante, 2017)	Created from experiments	Accelerometer, radar, depth camera	Wrist	Ensemble subspace discriminant, linear discriminant, kNN, SVM	After fusing the radar, accelerometer, and camera, the ensemble classifier provides the greatest

					overall accuracy of 91.3 per cent. It is an enhancement of 11.2 per cent over radar-only findings and 16.9% over accelerometer-only findings.
(Nguyen, Le, and Pham, 2018)	Created from experiments	Accelerometer, gyroscope, magnetometer	Hip	SVM, random forest	The best results come from Random Forest. Accelerometer precision = 86.23 percent and Accelerometer recall = 87.46 percent

					without sensor fusion Precision = 94.78 percent and recall = 94.37 percent using sensor fusion.

2.6.2 AAL with deep learning for fall detection

This section describes some of the most common and contemporary fall detection approaches that use DL techniques.

For fall and non-fall circumstances, (Fakhrulddin, Fei, and Li, 2017) use Convolutional Neural Network (CNN) to stream time series accelerometer data acquired from body sensor networks (BSN). Implementing CNNs with three convolution layers on data obtained through accelerometers is presented in (Santos, Endo, Monteiro, Rocha, Silva, and Lynn, 2019). Each layer's activation function is a linear unit that has been rectified (ReLU). But, according to the study, the lack of publicly available datasets using accelerometer and gyroscope data makes developing DL solutions for this type of data difficult. DL methods further need powerful computing processors, which may be incompatible with the restricted structure of wearable devices.

For instance, to identify falls and ADLs, (Dawar and Kehtarnavaz, 2018) employ a CNN-based sensor fusion system. Distinctive CNNs are given signals from a depth camera with wearable sensors (acceleration and rotational velocity). The programme then combines the results of these two CNNs to give a classification.

(Zhou, Qian, You, Ding, and Han, 2018) also propose a method for processing two types of inputs using two CNNs and then integrating the findings of the two CNNs to provide ultimate diagnosis results. Radar signals measure velocities and acceleration of individual body parts, and pictures from an optical camera are used as inputs. References (Ramezani, Xiao, and Naeim, 2018) and (Nguyen, Le, and Pham, 2018) are two further studies in this vein.

DL algorithms for fall detection utilising wearable devices have recently piqued people's curiosity. (Musci, De Martini, Blago, Facchinetti, and Piastra, 2020) Report an RNN method with LSTM blocks for fall detection using data from 3D accelerometers. Despite the difficulty of distinguishing highly dynamic behaviours from falls, the technique outlined in the research produces a superior overall categorization.

(Torti, Fontanella, Musci, Blago, Pau, Leporati, and Piastra, 2018) describe how to develop RNN architectures for limited embedded devices on a microcontroller unit (MCU) for fall detecting with triaxial accelerometers. For embedding a general RNN architecture on an MCU, the work further presents an abstraction of formulae for memory, processing power, and energy usage.

Table 2.4 summarises recent research on deep learning in wearable devices to detect falls. All literature studies use public datasets or fall simulations in controlled settings. Where available, the table qualitatively compares several fall detection algorithms and highlights their performance criteria like accuracy, sensitivity, and specificity.

Table 2. 4: DL-based systems with Wearable devices for fall detection

Reference	Dataset utilized	Sensors utilized	Sensor positioning	Methodology	Performance observed
(Fakhrulddin, Fei, and Li, 2017)	Public datasets	Accelerometer	There is no information available.	Time series accelerometer data transformed into pictures was subjected to a CNN-based analysis.	92.3 per cent accuracy
(Santos, Endo, Monteiro, Rocha, Silva, and Lynn, 2019)	Public datasets	Accelerometer	There is no information available.	CNN-based feature extraction models	The highest level of precision is 99.86 per cent.

(Dawar, and Kehtarnavaz, 2018)	Created from experiments	Depth camera, accelerometer	Waist	CNN	Fall accuracy is 100 per cent.
(Zhou, Qian, You, Ding, and Han, 2018)	Created from experiments	Radar signals, accelerometer, optical camera	There is no information available.	The data training and fall activity detection is accomplished using several convolutional neural networks (CNNs).	Radar only (Alex-Net only) 97.56% Radar only (Alex-Net + SSD-Net) 99.63% Radar + Optical camera (Alex-Net+SSD-Net + Aspect ratio analysis) 99.85%
(Musci, De Martini, Blago, Facchinetti,	SisFall dataset	Tri-axial accelerometers and a Tri-axial gyroscope	Waist	Long Short-Term Memory (LSTM)-based structures are a favourite type	Specificity = 97.15, Sensitivity = 96.73 and 96.94 percent accuracy

and Piastra, 2020)				of Recurrent Neural Networks (RNNs) commonly used to detect falls.	
(Torti, Fontanella, Musci, Blago, Pau, Leporati, and Piastra, 2018)	SisFall dataset	Tri-axial accelerometers and a Tri-axial gyroscope	Waist	Recurrent neural networks (RNNs) are integrated with microcontroller units (MCUs) for genuine fall detection.	98.73 per cent Sensitivity 97.93 per cent specificity 98.33 per cent accuracy

2.7 Challenges and issues in fall detection systems

Challenges and problems in diagnosing the fall were discovered after the reviewing work. The number of research utilizing context-aware approaches continues to rise; however, there is a novel trend toward incorporating fall detection into smartphones and employing ML and DL techniques in the detector. This section also identifies performance, usability and user acceptance issues under real-life conditions, energy usage, real-time activities, sensitivity limits, privacy, and real-life fall log.

2.7.1 Challenges

The creation of fall detectors confronts several significant issues, which are discussed in this section.

Challenge 1: Performance in real-world situations. Fall detectors should be as precise and dependable as feasible. A reliable fall detector must provide good sensitivity as well as specificity. It is often achieved in experimental settings; however, the discovery rate drops when applied to real-world scenarios (Kang and Kim, 2021). Because there is no defined technique or public database for comparison, these devices are built and evaluated in under-regulated settings. For instance, they employ information from falls and ADL of young individuals replicated at the discretion of all authors. It's also worth noting that fall detection systems are designed for older individuals. Therefore they should be included in the creation process. Only a few research (Kang and Kim, 2021) and (Igual, Medrano, and Plaza, 2013) include data from older adults, even if their involvement is confined to performing a set of simulated everyday activities for several minutes or hours. To assess the system's performance in real-world surroundings is insufficient. The devices should be worn for longer durations. Several types of research (Bagala, Becker, Cappello, Chiari, Aminian, Hausdorff, and Klenk, 2012) and (Kang and Kim, 2021) have gone in this approach, given a numerous erroneous activations, among other issues.

Challenge 2: Usability. Smartphone-based fall detection systems appeal to ubiquitous smartphone use, especially among the elderly. It enabled extremely stereotyped observations that helped with accuracy rates; however, the findings were less relevant to how individuals carry their smartphones daily (for example, in purses) (Albert, Kording, Herrmann, and Jayaraman, 2012). The installation of future smartphone-based detection systems must not be limited to a particular body area. Smartphones must be utilized normally, with no limitations on their placement or functionality. It might result in reduced discovery rates.

Challenge 3: Acceptance. There is very little information available concerning the technology's feasibility and acceptance. Acceptance by elders is a key issue because they might not be accustomed to electrical equipment. The system's method is critical to overcoming this difficulty (Igual, Medrano, and Plaza, 2013). The detection system should turn on and run independently, absent the need for human participation. In this regard, vision systems, such as other non-intrusive approaches, are excellent. But, several wearable devices, such as smartphones, offer additional benefits that may aid in adopting fall detection systems. They may be used outdoors and indoors, and they can combine fall detection with other medical applications in a similar device. The typical aversion to carrying various devices, each aimed at a particular purpose, would be solved in this manner. Smartphone usage by elderly persons, on the other hand, is not devoid of complexity: these devices, as designed, constitute a significant usage hurdle for them. The lack of rating in previous fall detection programmes that indicate minimal actual use is evidence of this. Possible solutions to increase the accessibility and availability of smartphones are required in this regard. Nonetheless, we discovered that fall detection systems are extremely valued by older adults, who demonstrated a good approach toward smartphone-based solutions following a realistic presentation of many assistive technologies in still-in-progress research.

2.7.2 Issues

This section discusses the most important factors that may obstruct the system's performance.

Issue 1: Smartphones' Limitations. The tendency to smartphone-based detection has significant drawbacks. To begin with, smartphones were not designed with fall discovery or other security applications in mind. Real-time functions, sensor architecture, the reliability of the accelerometer's sample frequency, operating system characteristics, and so on might all provide problems. However, relying on the smartphone method in which it is placed, the identical fall detector may function slightly differently. In a real-world setting, this probability must be considered.

Furthermore, smartphones cannot be overwhelmed with continual sensing obligations that degrade the phone's effectiveness, such as draining the battery. To trade off the quantity of battery used, it's critical to control the sleeping pattern of sensing elements. However, the battery life of smartphones is usually limited, which may limit their appeal. It isn't a trivial issue, particularly because the system is designed for senior citizens with limited mobility. Third, there is a demand for simple-to-use smartphones, and we are now in the hand of producers to meet that want. The improved, adapted devices will be influenced by the potential market for applications for those with poor technical capabilities. Nevertheless, fall detection systems are improbable to acquire the same level of resilience and stability as other assistive techniques like press-for-help devices.

Issue 2: Privacy concerns. Sensor-based systems, particularly fall detectors, have raised privacy issues. Not all sensors are equivalently susceptible: context-aware systems in common and vision-based devices specifically are far more subject to confidentiality problems than body-worn acceleration-based devices, for example. The security of critical environmental data should be ensured in any situation. Assistive technologies' potential advantages should not be hampered by privacy concerns since privacy cannot be compromised to achieve other goals (Cavoukian, Mihailidis, and Boger, 2010). In common, research on fall detection doesn't always have data privacy policies in place. It demonstrates that they are still a long way from real-world implementation.

Issue 3: Comparison between various technologies: public databases. It's impossible to compare various techniques since each author collects data in various manners: kinds of simulated falls, detector position, frequency of sampling, temporal signal duration, retrieved characteristics, etc. The focus of the study should not only be on the algorithm that will be utilized but also on how signals will be gathered and processed before being fed into a classification algorithm. A real-world dataset of accelerometer signals and videos of individuals falling might aid in the comparison of various approaches and the improvement of the results' repeatability. Sharing the algorithms' source code might further be beneficial.

Issue 4: Real-life falls. The majority of research relies on simulated falls data from young and older persons. Even if they were completely accessible to allow for a reasonable comparison of various methodologies, it is uncertain if the simulated behaviours indicate their exact equivalents. The assessment of the detection systems is extremely restrained since it is not appropriate to put elderly individuals to simulated falls. Only a few researchers (Albert, Kording, Herrmann, and Jayaraman, 2012) and (Igual, Medrano, and Plaza, 2013) give acceleration data from falls of the elderly; however, the number of occurrences recorded remains small. Furthermore, the causes of the falls are unknown due to a lack of comprehensive documentation (Igual, Medrano, and Plaza, 2013).

2.8 Conclusion

This chapter surveys the associated work of elderly fall discovery. It summarizes the topics of AAL, elderly fall detection using AAL, elderly fall detection using AAL with AI, privacy concerns, and other challenges and issues around fall detection are presented. Finally, it describes the detailed study of various fall detection methods.

CHAPTER 3

Multi-strategy Combination based Feature Selection (MCFS) for reducing the dimensionality of Fall Detection Data

3.1 Introduction

Data pre-processing is a data mining approach that entails converting raw data into an understood format. Outlier data, missing values, insufficient data, and noisy data are general in real-world data. As a result, this data must pre-process before it can be mine. Data pre-processing is a crucial step in improving data effectiveness. Data pre-processing is one of the common data mining procedures that involves the preparation and conversion of the data while also attempting to improve the efficiency of information retrieval. Cleaning, integration, transformation, selection, and reduction are some of the techniques used in pre-processing.

One of the data pre-processing strategies, feature selection, has proven efficient and successful in data preparation (particularly high-dimensional data) for different data mining and machine learning issues. Building more basic and intelligible models, enhancing data mining speed, and collecting clean, coherent data are all goals of feature selection. It entails finding a subset of the most valuable features that give outcomes comparable to the whole set of features.

The proposed feature selection algorithm for the fall detection dataset is described in this chapter. The best features from the dataset are selected using the Multi-strategy Combination based Feature Selection (MCFS) algorithm. This chapter provides this feature selection algorithm for fall detection data in great depth. The numerous types of feature selection strategies explain in Section

3.2. The proposed feature selection algorithm for fall detection data explain in Section 3.3. Lastly, in section 3.4, the contents of this chapter are summarised.

3.2 Overall System Architecture

The entire system architecture for fall detection is shown in Figure 3.1.

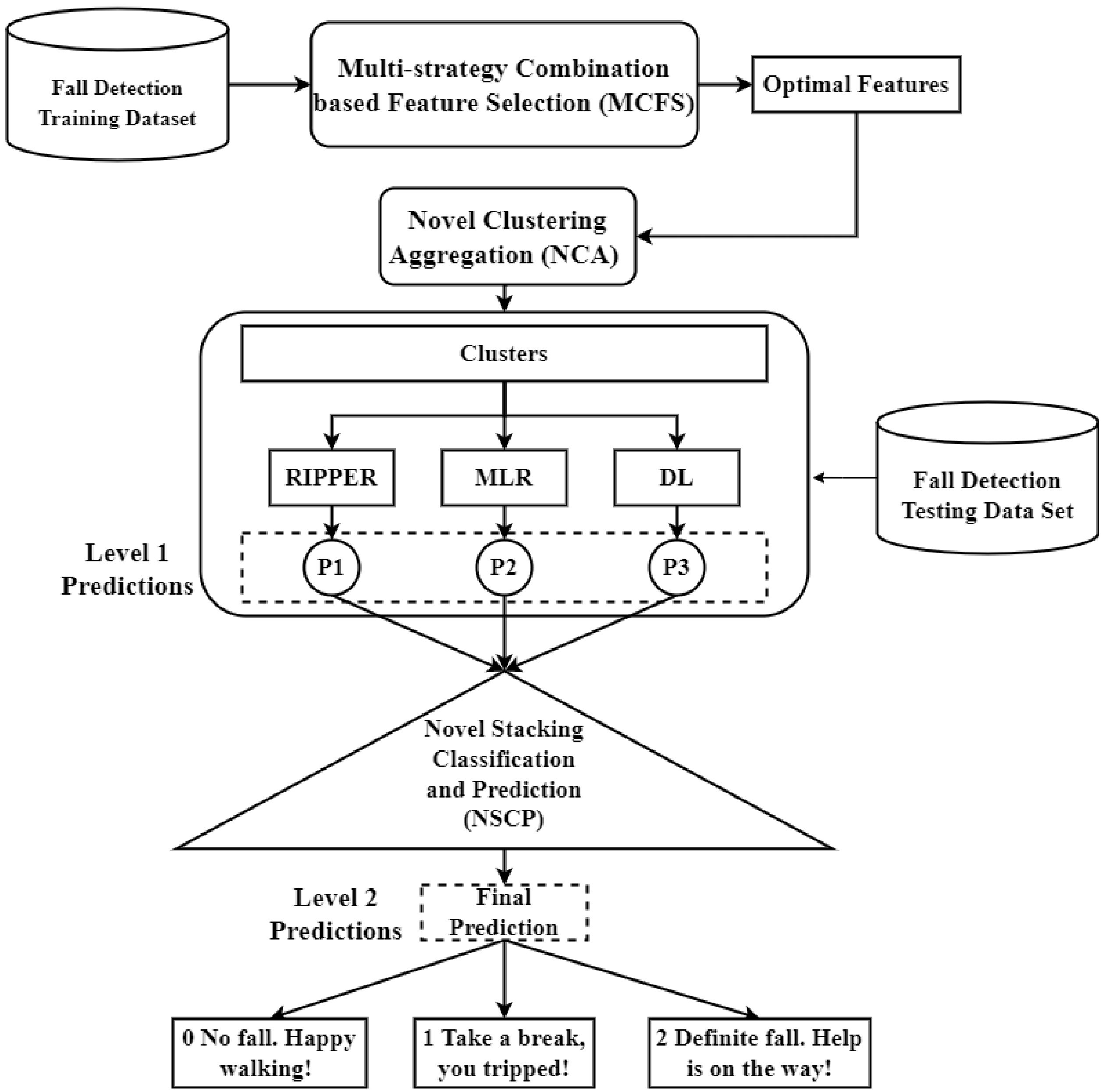

Figure 3. 1: Overall system architecture

3.3 Feature Selection

Data pre-processing has a notable impact on the supervised machine learning technique (Mishra, Biancolillo, Roger, Marini and Rutledge, 2020). Eliminating noise instances is one of the most difficult problems in induction machine learning. Deleted instances, on the whole, have a lot of deviating instances with a lot of null feature values. These deviating features are also outliers (Abd Mutalib, Satari, and Yusoff, 2021). Furthermore, selecting a single sample from a large dataset is a common way to deal with the incapacity to learn from it. Missing data handling is another issue that is usually resolved during the pre-processing data stage (Xiao and Bulut, 2020). The symbolic, logical learning techniques can execute symbolic, categorical data alone. Real-world challenges, on the other hand, involve both numerical and symbolic features. As a result, discretizing numerical features is a substantial challenge. Compiling the symbolic feature values is a useful method. However, it was recognised that when it came to selecting the most informative attributes, the multi-valued features were exaggerated, both in terms of determining decision rules and producing decision trees (Cai, Luo, Wang, and Yang, 2018).

Moreover, in actual data, several attributes are frequently used to represent data. However, merely some of them might be associated with the target idea. There may be redundancy, in which certain features are associated; hence, it is not necessary to include all of them in modelling; and interdependence, in which many features among them convey important data ambiguous if a few of them are added on their own. Feature selection is a technique for finding and removing as much duplicate and useless information as possible (Remeseiro and Bolon-Canedo, 2019). It decreases the dimensionality of the information and also might permit learning techniques to function quicker with more efficiently. In a few cases, accuracy in future classification could enhance; in others, the outcome is denser; simply interpret the illustration of the target idea. The significant goals of feature selection are: (1) to enhance predictive accuracy, (2) to eliminate unnecessary features and (3) to decrease time utilization during analysis. Figure 3.2 explains the Feature Selection.

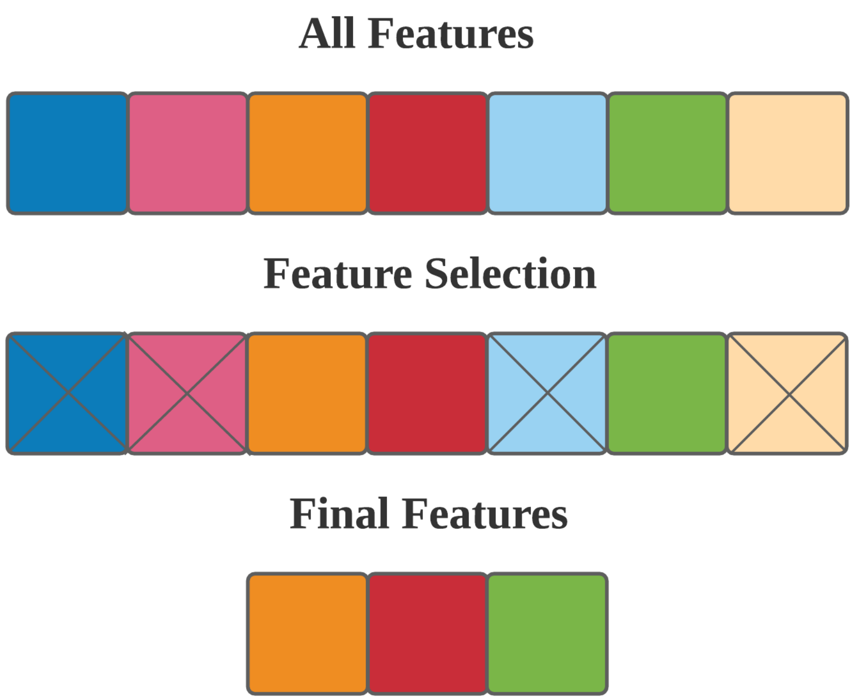

Figure 3. 2: Feature selection

3.4 Feature Selection Types

Without utilising a learning algorithm, filter-based feature selection selects a subset of features from big dimensional databases (Ma and Gao, 2020). Filter techniques are generally faster, but the classifier's accuracy is not guaranteed. Concurrently, the wrapper-based feature selection employs a learning algorithm to evaluate the selected subset of features (Karasu, Altan, Bekiros, and Ahmad, 2020). For specific classifiers, wrapper approaches may offer higher classification precision than filter approaches, but they are less effective. Embedded-based feature selection is used to carry out feature selection, whereas training is specific to the learning algorithm (Siva Shankar, Ashokkumar, Vinayakumar, Ghosh, Mansoor, and Alnumay, 2020). Over the last two decades, numerous feature selection strategies have been proposed that work in different ways and use various measurements (similar to correlation, probability distribution, etc.). Various feature selection algorithms may select a subset of different features for a particular database, resulting in multiple accuracies. As a result, the ensemble-based feature selection algorithm is used to choose a robust feature set that improves classification accuracy.

3.5 Multi-strategy Combination based Feature Selection (MCFS)

This chapter presented Multi-strategy Combination based Feature Selection (MCFS) to reduce the dataset's dimensionality. Given a high dimensional dataset F = {F1,...,FD} containing D features, FS1 = {F11,...,F1D} is the feature sequence ranked by Information Gain and FS2 = {F21,...,F2D} is the one by Fisher Score and FS3 = {F31,...,F3D} is the feature sequence ranked by Min-Max Normalization and FS4 = {F41,...,F4D} is the feature sequence ranked by Correlation coefficient and FS5 = {F51,...,F5D} is the feature sequence ranked by Mean Absolute Deviation. To filter low-scored features provide in both feature sequences, we use the union strategy to combine the lowest C per cent of the two sequences and filter them out of the original feature sets. After combining features, the new feature subset FS can be defined as follows: FS = F − {C%{FS1} ∪ C%{FS2} ∪ C%{FS3} ∪ C%{FS4} ∪ C%{FS5}}. Based on these selected best features, the dimensionality of F is reduced. Algorithm 3.1 shows the proposed Multi-strategy Combination based Feature Selection.

Algorithm 3.1: Multi-strategy Combination based Feature Selection (MCFS)

Step 1	**:**	HD ← Load High Dimensional dataset D
Step 2	**:**	IGFS ← Information Gain based Feature Selection from HD // **Strategy 1**
Step 3	**:**	FSFS ← Fisher Score based Feature Selection from HD // **Strategy 2**
Step 4	**:**	MMFS ← Min-Max Normalization based Feature Selection from HD // **Strategy 3**
Step 5	**:**	CCFS ← Correlation Coefficient based Feature Selection from HD // **Strategy 4**
Step 6	**:**	MADFS ← Mean Absolute Deviation based Feature Selection from HD // **Strategy 5**
Step 7	**:**	OF ← Extract optimal features from IGFS, FSFS, MMFS, CCFS and MADFS as Optimal Features

Figure 3.3 shows the flow diagram of the proposed MCFS algorithm.

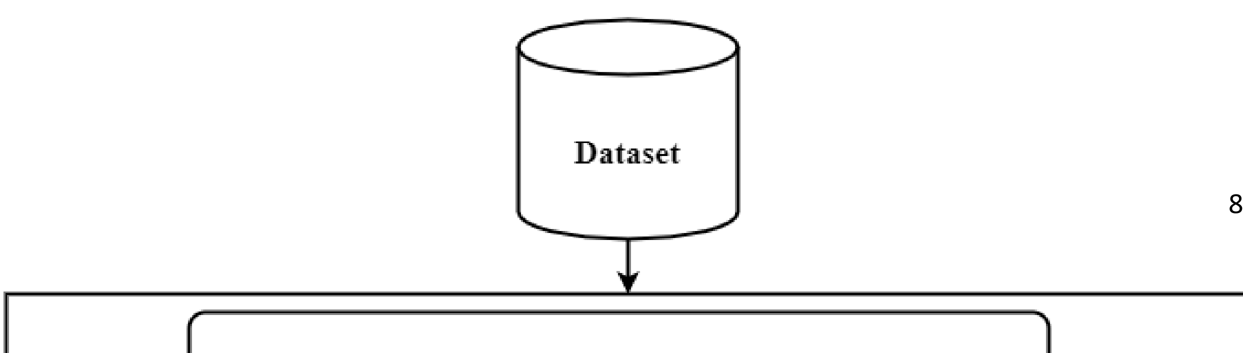

Figure 3. 3: Multi-Strategy Combination Based Feature Selection

3.5.1 Strategy 1: Information Gain based Feature Selection

Information gain is an essential strategy for feature selection by assessing each variable's progress in the target variable's situation (Jadhav, He, and Jenkins, 2018). To discover the split point and the feature to split, Information Gain uses Entropy, shown in Eq. 3.1. Entropy is the nonexistence of order or predictability. In a variety of situations, it refers to the measuring of impurity. Moreover, if a node only has instances of one class, it is the purest.

$$Entropy = -\sum_{i}^{n} \log_2(P_i) \tag{3.1}$$

Where n= number of features

i = feature

P = Probability of i

For each feature, entropy is determined, and the feature with the lowest value is chosen for splitting. As a result, Entropy has a mathematical range of 0 to 1.

The second step is to calculate the information gain (IG) between 0 to 1. A higher information gain implies a smaller entropy group. The tree uses information gain to choose which attribute to split on for high information gain. We may compute information gained for each attribute one by one using Eq. 3.2.

$$IG = Entropy(parent) - \ weighted\_avg \ * \ Entropy(children) \tag{3.2}$$

Here the parent is the target feature, and children are other features from a dataset. After calculating information gained for each feature, sort all information gained based on descending order. Now we can select more prominent information gain features as optimal features.

3.5.2 Strategy 2: Fisher Score based Feature Selection

Fisher score is a variant of Newton's approach for numerically solving maximum likelihood equations in statistics (Aksu, Üstebay, Aydin, and Atmaca, 2018). For example, the score of the ith feature Si would be computed by Fisher Score, which is shown in Eq. 3.3.

$$Si = \frac{\sum n_j (u_{ij} - u_i)^2}{\sum n_j * p_{ij}^2} \tag{3.3}$$

Where μij and ρij are the mean and variance of the ith feature in the jth class, correspondingly, nj is the jth class's number of occurrences, and μi is the ith feature's mean; after calculating the fisher score for each feature, sort all fisher scores based on descending order. Now we can select more significant fisher score features as optimal features.

3.5.3 Strategy 3: Min-Max Normalization

One of the most well-known data normalization approaches is min-max normalization (Munkhdalai, Munkhdalai, Park, Lee, Li, and Ryu, 2019). To begin, each feature's minimum value is changed to a 0. The maximum value is then changed to a 1. Every other value is also converted to a decimal between 0 to 1. Min-Max Normalization is explained in Eq. (3.4).

$$Xnorm = \frac{\mathrm{x} - \min(\mathrm{x})}{\max(\mathrm{x}) - \min(x)} \tag{3.4}$$

For instance, if the minimum value of a feature was min(x) = 20, and the maximum value was max(x) = 40, then x = 30 would be altered to about Xnorm= 0.5. Then calculate the score for ith feature based on the average value of each instance in ith feature; after calculating the Min-Max normalization score for each feature, sort all Min-Max normalization scores based on descending order. Now we can select larger Min-Max normalization score features as optimal features.

3.5.4 Strategy 4: Correlation Coefficient

The Correlation coefficient is a statistic utilized to calculate the linear correlation between feature X and the target Y feature (Mohammadi, Mirvaziri, Ghazizadeh-Ahsaee, and Karimipour, 2019). It ranges from +1 to -1, with 1 indicating an entire positive correlation and -1 indicating an extreme negative correlation. As a result, a value of 0 denotes the absence of a linear relationship. To compute the correlation coefficient, divide the covariance of the input X and output Y attributes by the product of the standard deviations of the two features — the formula is presented in Eq (3.5).

$$CC = \frac{\mathrm{Cov(X, Y)}}{\mathrm{\sigma X - \sigma Y}} \tag{3.5}$$

Cov denotes covariance, and σX denotes X's standard deviation; also, σY denotes Y's standard deviation. Based on the above equation CC score of each feature can compute after calculating the correlation coefficient score for each feature, sorting all correlation coefficient scores based on descending order. Now we can select more prominent correlation coefficient score features as optimal features.

3.5.5 Strategy 5: Mean Absolute Deviation

The average distance among all data points and the mean is the dataset's mean absolute deviation (Khair, Fahmi, Al Hakim, and Rahim, 2017). The mean absolute deviation can be calculated in the following way.

Step 1: Determine the mean.

Step 2: Utilizing positive distances, compute how distant each data point is from the mean. Absolute deviations are what they're called.

Step 3: Add up all of the deviations.

Step 4: Divide the total number of data points by the number of data points.

Following these steps in the example below is probably the most effective approach to learning about mean absolute deviation; however, there is a more formal way to write the steps in a formula, as shown in Eq (3.6):

$$MAD = \frac{\sum |\mathrm{x_i} - \bar{\mathrm{x}}|}{\mathrm{n}} \tag{3.6}$$

Based on the above equation, the MAD score of each feature can compute after calculating the MAD score for each feature and sort all MAD scores based on descending order. We can select more significant MAD score features as optimal features. Feature selection reduces the number of impute variables (dimension reduction) and identifies features that contribute to improving performance.

3.6 Process descriptions of Feature Selection

Table 3.1 discusses the working process of feature selection.

Table 3. 1: Process Descriptions of Feature Selection

Process	Descriptions
Load Input Dataset (cStick Dataset)	In this process, cStick dataset given as input to the proposed framework. It has seven features namely, Distance, Pressure, HRV, Sugar Levels, SpO2 levels, and accelerometer reading, concerning the decision of falls.

Feature Selection based on Information Gain	In this process, 5 features are selected based on information gain strategy. These 5 features are Distance, Pressure, HRV, SpO2 and Decision.
Feature Selection based on Fisher Score	In this process, 5 features are selected based on Fisher Score strategy. These 5 features are Distance, Sugar Level, SpO2, Accelerometer and Decision.
Feature Selection based on Min-Max Normalization	In this process, 5 features are selected based on Min-Max Normalization strategy. These 5 features are Pressure, HRV, SpO2, Accelerometer and Decision.
Feature Selection based on Correlation Co-efficient	In this process, 5 features are selected based on Correlation Co-efficient strategy. These 5 features are Distance, Pressure, SpO2, Accelerometer and Decision.
Feature Selection based on Mean Absolute Deviation	In this process, 5 features are selected based on Mean Absolute Deviation strategy. These 5 features are Distance, HRV, Sugar Level, SpO2, and Decision.

Multi-strategy Combination based Feature Selection	In this process, 5 features are selected based on the combination of all of the above strategies. These 5 features are Distance, Pressure, HRV, SpO2, and Decision.

3.7 Dataset Description

The cStick fall prediction dataset is gathered from https://www.kaggle.com/datasets/laavanya/elderly-fall-prediction-and-detection?select=cStick.csv. The dataset contains 2039 instances and seven features. Although there are many wearables in the industry that offer care for elderly persons, the majority of them concentrate on fall detection rather than prediction. A few suggested wearables can identify slips and trips but not falls. As a result, the cStick, a calm stick designed to track falls in older adults, has been proposed. cStick was created to assist both visual and auditory impairments in elderly individuals. cStick has the potential to not merely detect but also predict falls to limit the number of times they occur. cStick can keep an eye on the surroundings, alert the user if there has been the last fall at a specific spot, and provide the user with information about the area and its surroundings. The decision of fall, that is, a forecast, warning, or detection of fall, is performed with an accuracy of around 95% based on variations in the experimental parameters.

Distance, pressure (0-small, 1-medium, and 2-high pressures), HRV, Sugar Levels, SpO2 levels, and accelerometer reading (<+-3g, that is, the threshold is 0 and >Threshold is 1) concerning the decision of falls (0-no fall detected, 1-person tripped/slipped/prediction of fall, and 2-definite fall) are all included in the cStick.csv file. Descriptions of attributes are shown in Table 3.2.

Table 3. 2: cStick dataset attribute description

Attribute	Description	Range
Distance	The user's distance from the nearest object, person, or thing	0 to 70
Pressure	Applying hand pressure on the stick	0 to 2
HRV	The user's heart rate variability	60 to 125
Sugar level	The user's sugar levels	10 to 179
SpO2	The user's oxygen saturation levels	60 to 100
Accelerometer	The user's accelerometer reading	0, 1
Decision	The decision of fall for the user	(0 - no fall detected, 1 - person slipped/tripped/prediction of fall and 2 - definite fall)

Furthermore, Table 3.3 shows a sample of 10 instances of the cStick dataset.

Table 3. 3: Sample 10 instances of the cStick dataset

Distance	Pressure	HRV	Sugar level	SpO2	Accelerometer	Decision
25.5400	1.0000	101.3960	61.0800	87.7700	1.0000	1
2.5950	2.0000	110.1900	20.2070	65.1900	1.0000	2
68.0670	0.0000	87.4120	79.3450	99.3450	0.0000	0
13.0900	1.0000	92.2660	36.1800	81.5450	1.0000	1
69.4300	0.0000	89.4800	80.0000	99.9900	0.0000	0
27.1600	1.0000	102.5840	64.3200	88.5800	1.0000	1
57.1340	0.0000	70.8240	73.6900	93.6900	0.0000	0
66.3560	0.0000	84.8160	78.4600	98.4600	0.0000	0
60.3820	0.0000	75.7520	75.3700	95.3700	0.0000	0
23.1700	1.0000	99.6580	56.3400	86.5850	1.0000	1

3.8 Metrics for Evaluation

This section outlines the experiment's evaluation metrics. The evaluation metrics are Accuracy, Sensitivity, Specificity, Precision, Execution time, Size of the dataset, Number of features, Recall and f-measure.

3.8.1 Metrics for Feature Selection

Accuracy:

The accuracy of a test is its capability to distinguish the patient and healthy cases properly. As a result, we must compute the fraction of true positive and true negative in all analyzed cases to estimate a test's accuracy. It could be expressed mathematically as Eq (3.7):

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{3.7}$$

Sensitivity:

A test's sensitivity refers to its capacity to determine patient cases appropriately. We'll need to figure out what proportion of inpatient instances are true positives to assess it. As a result, it's also known as Recall. It can be expressed mathematically as Eq (3.8):

$$Sensitivity = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3.8}$$

Specificity:

A test's specificity refers to its capacity to identify healthy instances appropriately. Calculate the proportion of true negatives in healthy instances to assess it. It can be expressed mathematically as Eq (3.9):

$$Specificity = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{3.9}$$

Precision:

Precision is also called Positive Predictive Value which is shown in Eq. (3.10):

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3.10}$$

Here a “TP” is the number of cases correctly identified as a patient, and an “FP” is the number of instances incorrectly identified as a patient. Thus, with a perfect test, the ideal value of the precision is 1 (100%), and the worst possible value would be zero.

Execution time:

Execution Time is the time taken for dimensionality reduction. It is measured by java.Lang.System.currentTimeMillis() method captures milliseconds; the value primarily depends on the underlying system. For example, several other operating systems calculate time in units of tens of milliseconds. Mathematically, this can be stated as Eq. (3.11):

$$\begin{aligned} &Execution\ Time = CurrentTimeMillis\quad after\quad Dimensionality\quad Reduction\quad - \\ &CurrentTimeMillis\ before\ Dimensionality\ Reduction \end{aligned} \tag{3.11}$$

Features Count:

Many features will cause a problem to suffer from the Curse of dimensionality. So features count is another unavoidable metric.

3.9 Results and Discussions

This section explains how to evaluate the performance of the MCFS algorithm for feature selection. Java has been used to generate this research experiment (version 1.8). For the experiment, a real-world dataset was obtained from the Internet. The particular features are extracted using the MCFS algorithm.

3.9.1 Feature Selection Results

This section examines the results of the feature selection process. The goal of feature selection is to decrease the number of features, choose the most relevant ones, and eliminate those that aren't. However, due to the huge search area when the number of features is large, finding the optimal set of features is regarded as a demanding and complicated issue.

Table 3.4 compares accuracy among IG-based, FS-based, MMN-based, CC-based, MAD-based and MCFS dimensionality reduction algorithms.

Table 3. 4: Comparison of Accuracy among IG-Based, FS-Based, MMN-Based, CC-Based, MAD-Based and MCFS for Fall Prediction Dataset

Algorithm	Accuracy (in %)
IG-based	79.1
FS-based	81.8
MMN-based	70.1
CC-based	85.2
MAD-based	84.7
MCFS	86.2

Furthermore, Figure 3.4 shows an accuracy comparison. This comparison concludes MCFS algorithm is the best among others.

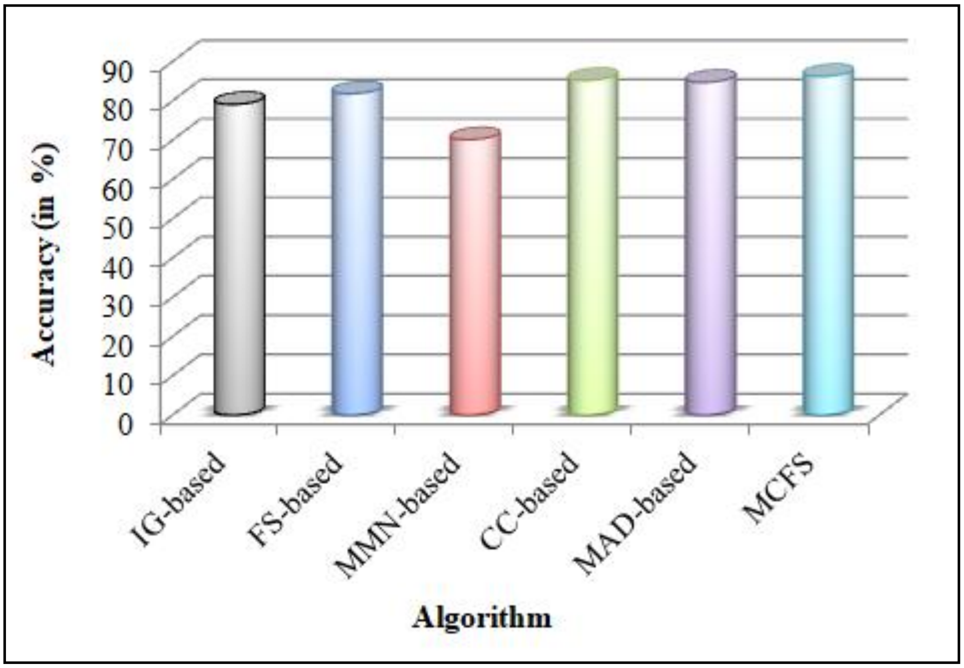

Figure 3. 4: Comparison of Accuracy among IG-Based, FS-Based, MMN-Based, CC-Based, and MAD-Based and MCFS for Fall Detection and Prediction Dataset

Among others, the accuracy of the MMN-based algorithm is significantly less. But compared with the MMN-based algorithm, the IG-based algorithm delivers the maximum level of accuracy. However compared with the IG-based algorithm, the FS-based algorithm delivers the maximum level of accuracy. But compared with the FS-based algorithm, the accuracy of the MAD-based algorithm is very high. However compared with the MAD-based algorithm, the CC-based algorithm delivers the maximum level of accuracy. But compared with the CC-based algorithm, the accuracy of the MCFS algorithm is very high.

Table 3.5 compares sensitivity among IG-based, FS-based, MMN-based, CC-based, MAD-based and MCFS dimensionality reduction algorithms.

Table 3. 5: Comparison of Sensitivity among IG-Based, FS-Based, MMN-Based, CC-Based, MAD-Based and MCFS for Fall Detection Dataset

Algorithm	Sensitivity (in %)
IG-based	82.2
FS-based	81.2
MMN-based	75.9
CC-based	76.2
MAD-based	76.3
MCFS	83.2

Furthermore, Figure 3.5 shows a sensitivity comparison. This comparison concludes MCFS algorithm is the best among others.

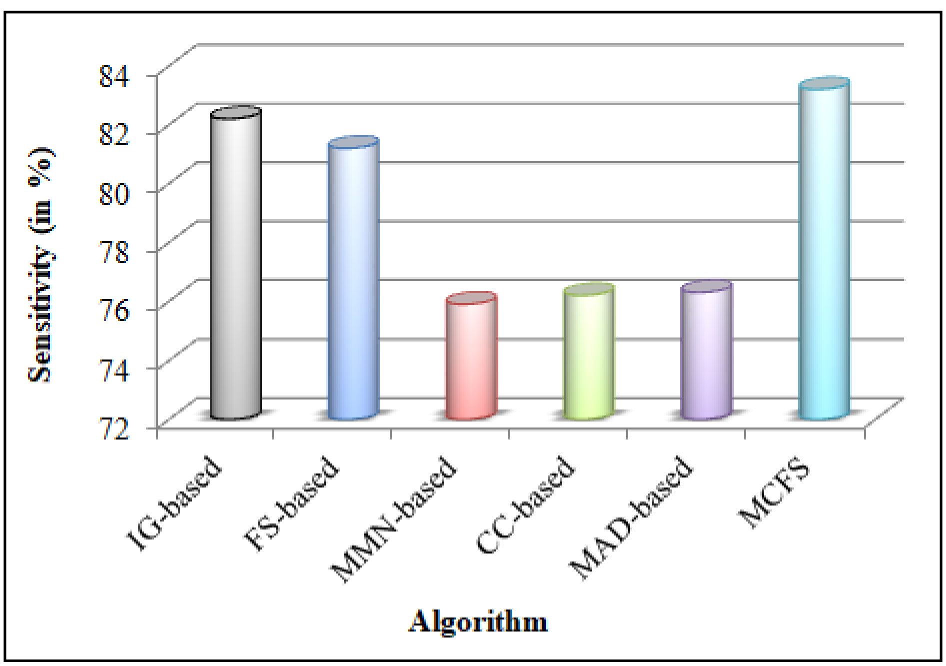

Figure 3. 5: Comparison of Sensitivity among IG-Based, FS-Based, MMN-Based, CC-Based, and MAD-Based and MCFS for Fall Detection and Prediction Dataset

Among others, the sensitivity of the MMN-based algorithm is significantly less. But compared with the MMN-based algorithm, the CC-based algorithm provides the highest sensitivity. However, the MAD-based algorithm provides the most heightened sensitivity compared with the CC-based algorithm. But compared with the MAD-based algorithm, the sensitivity of the FS-based algorithm is very high. However, the IG-based algorithm provides the most heightened sensitivity compared with the FS-based algorithm. But compared with the IG-based algorithm, the sensitivity of the MCFS algorithm is very high.

Table 3.6 compares specificity among IG-based, FS-based, MMN-based, CC-based, MAD-based and MCFS dimensionality reduction algorithms.

Table 3. 6: Comparison of Specificity among IG-Based, FS-Based, MMN-Based, CC-Based, MAD-Based and MCFS for Fall Detection Dataset

Algorithm	**Specificity** (in %)
IG-based	78.6
FS-based	73.5
MMN-based	73.9
CC-based	84.6

MAD-based	81.1
MCFS	85.6

Furthermore, Figure 3.6 shows a specificity comparison. This comparison concludes MCFS algorithm is the best among others.

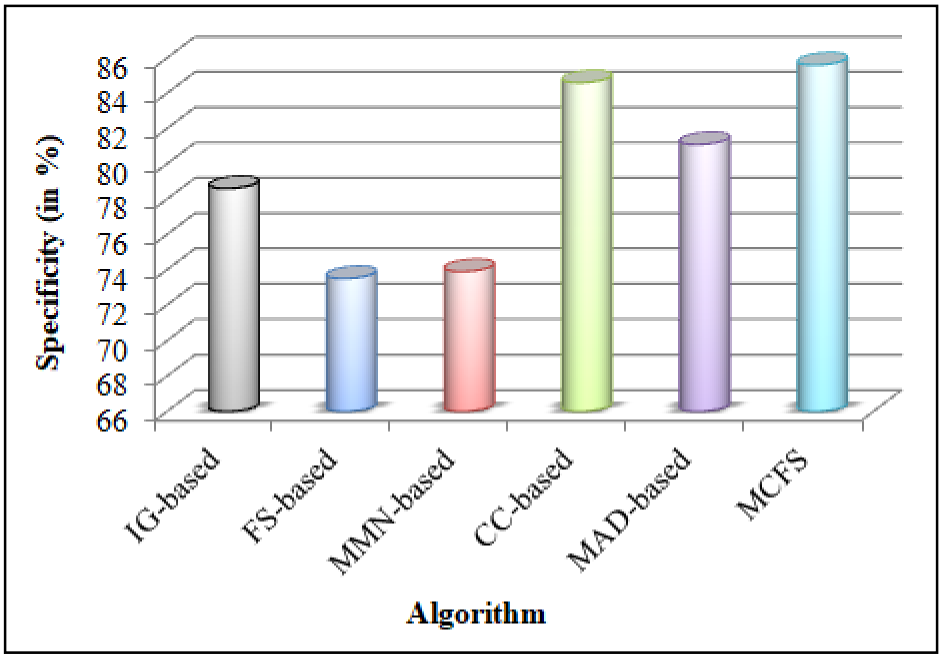

Figure 3. 6: Comparison of Specificity among IG-Based, FS-Based, MMN-Based, CC-Based, and MAD-Based and MCFS for Fall Detection Dataset

Among others, the specificity of the FS-based algorithm is significantly less. But compared with the FS-based algorithm, the MMN-based algorithm provides the highest specificity. However compared with the MMN-based algorithm, the IG-based algorithm provides the highest specificity. But compared with the IG-based algorithm, the specificity of the MAD-based algorithm is very high. However, the CC-based algorithm provides the highest specificity compared with the MAD-based algorithm. But compared with the CC-based algorithm, the specificity of the MCFS algorithm is very high.

Table 3.7 compares Precision among IG-based, FS-based, MMN-based, CC-based, MAD-based and MCFS feature selection algorithms.

Table 3. 7: Comparison of Precision among IG-Based, FS-Based, MMN-Based, CC-Based, MAD-Based and MCFS for Fall Detection Dataset

Algorithm	**Precision** (in %)
IG-based	85.4
FS-based	84.5
MMN-based	89.4
CC-based	74.2
MAD-based	86.4
MCFS	90.4

Furthermore, Figure 3.7 shows the precision comparison. This comparison concludes MCFS algorithm is the best among others.

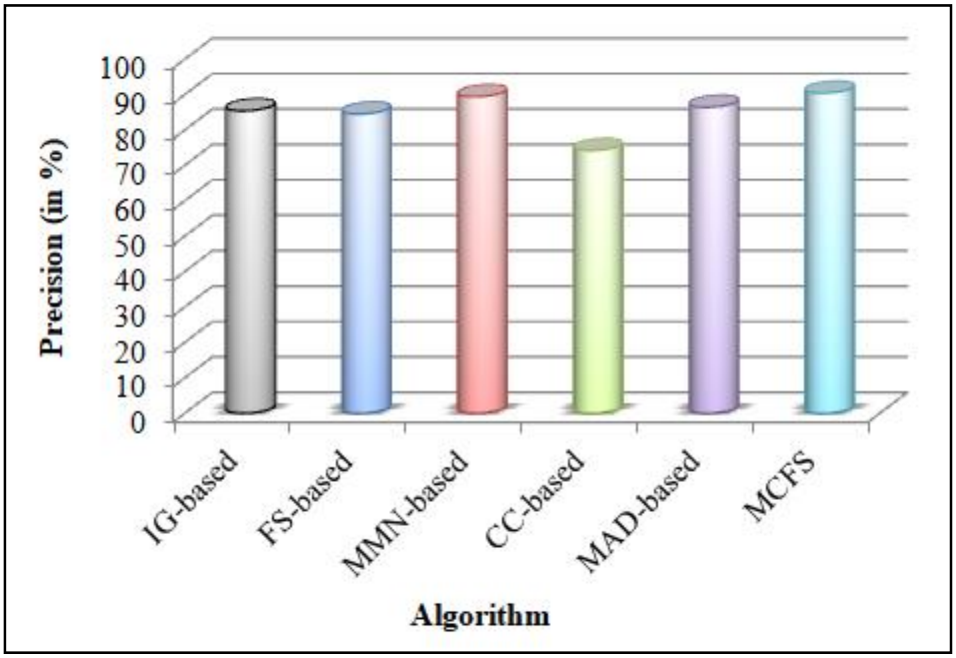

Figure 3. 7: Comparison of Precision among IG-Based, FS-Based, MMN-Based, CC-Based, and MAD-Based and MCFS for Fall Detection Dataset

Among others, the precision of the CC-based algorithm is significantly less. But compared with the CC-based algorithm, the FS-based algorithm presents the maximum level of precision. However compared with the FS-based algorithm, the IG-based algorithm presents the maximum level of precision. But compared with the IG-based algorithm, the precision of the MAD-based algorithm is very high. However compared with the MAD-based algorithm, the MMN-based algorithm presents the maximum level of precision. But compared with the MMN-based algorithm, the precision of the MCFS algorithm is very high.

Table 3.8 compares execution time among IG-based, FS-based, MMN-based, CC-based, MAD-based and MCFS feature selection algorithms.

Table 3. 8: Comparison of Execution Time Among IG-Based, FS-Based, MMN-Based, CC-Based, MAD-Based and MCFS for Fall Detection Dataset

Algorithm	Execution time (ms)
IG-based	4854
FS-based	1752
MMN-based	5421
CC-based	1906
MAD-based	1952
MCFS	1904

Furthermore, Figure 3.8 shows an execution time comparison. This comparison concludes that FS-based and MCFS algorithms are the best among others.

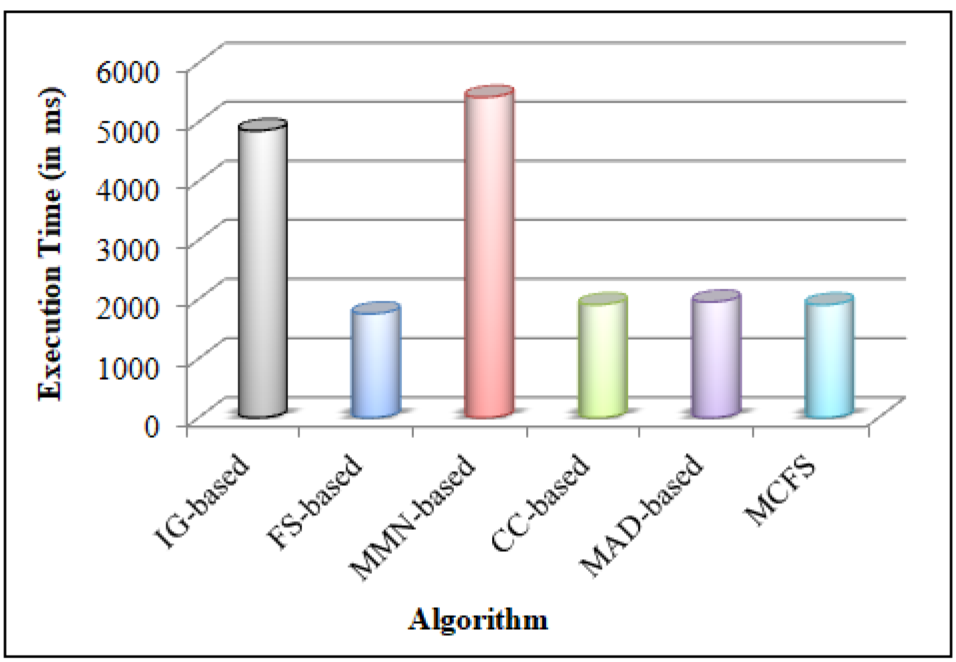

Figure 3. 8: Comparison of Execution Time among IG-Based, FS-Based, MMN-Based, CC-Based, and MAD-Based and MCFS for Fall Detection Dataset

Among others, the execution time of the MMN-based algorithm is very high. But compared with the MMN-based algorithm, the IG-based algorithm provides a lower execution time. However, the MAD-based algorithm provides a lower execution time compared with the IG-based algorithm. But compared with the MAD-based algorithm, the execution time of the CC-based algorithm is significantly less. However, the MCFS algorithm provides a lower execution time compared with the CC-based algorithm. But compared with the MCFS algorithm, the execution time of the FS-based algorithm is significantly less.

Table 3.9 shows the features count of the dataset after IG-based, FS-based, MMN-based, CC-based, MAD-based and MCFS feature selection algorithms.

Table 3. 9: Features count after IG-Based, FS-Based, MMN-Based, CC-Based, and MAD-Based and MCFS feature selection

Algorithm	Number of features
Original Dataset	7 (Distance, Pressure, HRV, Sugar level, SpO2, Accelerometer, Decision)
IG-based	5 (Distance, Pressure, HRV, SpO2, Decision)

FS-based	5 (Distance, Sugar level, SpO2, Accelerometer, Decision)
MMN-based	5 (Pressure, HRV, SpO2, Accelerometer, Decision)
CC-based	5 (Distance, Pressure, SpO2, Accelerometer, Decision)
MAD-based	5 (Distance, HRV, Sugar level, SpO2, Decision)
MCFS	5 (Distance, Pressure, HRV, SpO2, Decision)

Furthermore, Figure 3.9 shows a pictorial diagram of feature count.

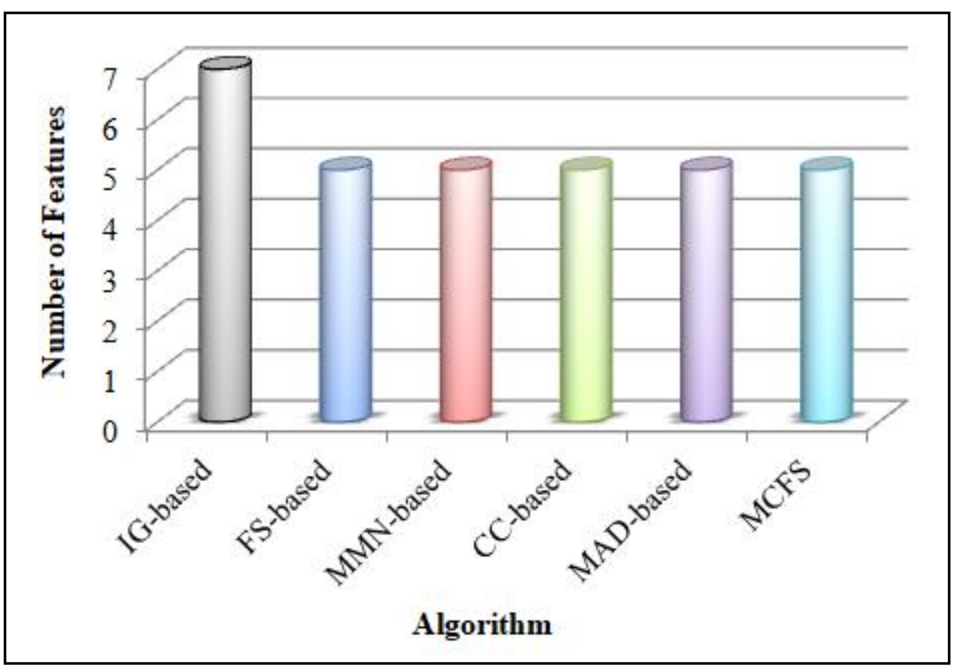

Figure 3. 9: Features count after IG-Based, FS-Based, MMN-Based, CC-Based, and MAD-Based and MCFS feature selection

The Fall Detection dataset has seven features. After feature selection, all algorithms selected five features.

3.10 Hypothesis Justification

Table 3. 10: Hypothesis Justification

Hypothesis	Justification
H3	Multi-Strategy Combination Based Feature Selection (MCFS) algorithm has successfully reduced the dimensionality of the dataset.

3.11 Conclusion

The feature selection approach for fall detection is explained in this chapter. A single feature selection method is insufficient; thus, this chapter provides Multi-strategy Combination based Feature Selection using Information Gain, Fisher Score, Min-Max Normalization, Correlation coefficient and Mean Absolute Deviation.

CHAPTER 4

Novel Clustering Aggregation (NCA) Algorithm for Clustering Fall Detection Data

4.1 Introduction

To recognize the activity of older people, a proficient and precise classification and prediction algorithm are essential. Before classification, a clustering technique is needed to enhance the classification's effectiveness, precision, and performance. The clustering technique clusters the data instances into subsets to cluster related instances jointly while different instances belong to different groups. Unfortunately, the availability of an enormous compilation of clustering algorithms in the literature could confuse experts attempting to select a suitable algorithm for a given dataset. In addition, no clustering algorithm can generally solve all issues, for example, cluster shape, noise or density. To deal with these issues, this chapter proposed a Novel Clustering Aggregation (NCA) algorithm using the aggregation of K-Means, Expectation-Maximization (EM) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithms. In the first step, these three clustering algorithms cluster the dataset individually. In the next step, final clusters are taken out through a majority voting process among the data instances. It means data instance allocated to the majority voted cluster by these three clustering algorithms.

4.2 Clustering

Machine Learning (ML) models could be beneficial in discovering fall detection in older adults. There are often numerous factors that find falls among older people. ML methods may help find hidden patterns from the fall detection dataset (Putra, Brusey, Gaura, and Vesilo, 2018).

Furthermore, ML methods are regarded as clustering, classification and prediction to resolve various issues in real-time applications. They guarantee classification and predictive solutions for performance reliability and stability (Gao, Calhoun, and Sui, 2018). Using ML methods, some researchers have implemented a lot of applications. Especially, Algorithms comprise statistics, clustering, decision trees, optimization algorithms and more. ML applications rely greatly on datasets that examine and discover patterns utilized to solve particular tasks. The fall detection system has possible incentives to take out and find hidden patterns in the database (Mauldin, Canby, Metsis, Ngu, and Rivera, 2018). Therefore, available data is globally scattered and unclear. They may also have inadequate and unimportant data recorded in terms of consistency in forecasting and classification. One of the main challenges is the real discovery of specific vital data.

Many proposed ML algorithms could forecast and examine fall detection (de Quadros, Lazzaretti, and Schneider, 2018). These algorithms consist of the decision tree (DT), ANN, SVM, KNN, linear regression (LR), NB and time series forecast models. Due to rapid discovery and constant changes in software engineering, a large amount of data can be generated (Ngiam and Khor, 2019). Retrieving patterns from these datasets and handling large-scale dimensional data has become a significant area of ML. The ML methodology is regarded as a classification of fall detection datasets to gain helpful knowledge to assist older people (Hussain, Umair, Ehatisham-ul-Haq, Pires, Valente, Garcia, and Pombo, 2019). To use ML methods that improve the efficiency of the classification process, a clustering algorithm is needed (Nejatian, Parvin, and Faraji, 2018). The clustering technique clusters the data instances into subsets to cluster similar instances together while different instances belong to other groups (Cui, Ding, Fan, and Al-Dhahir, 2018).

Let's know the clustering technique with the real-world instance of the supermarket: When we visit any supermarket, we can view that the items with related usage are grouped. Such as the groceries are grouped in one section, and personal care items are in other sections. Likewise, laundry/detergents, insect repellents, scrubbers, etc., are grouped in separate sections at house care sections. Consequently, we could effortlessly find out things. The clustering technique further works similarly. Other instances of clustering are grouping documents along with the topic. Figure 4.1 shows clustering.

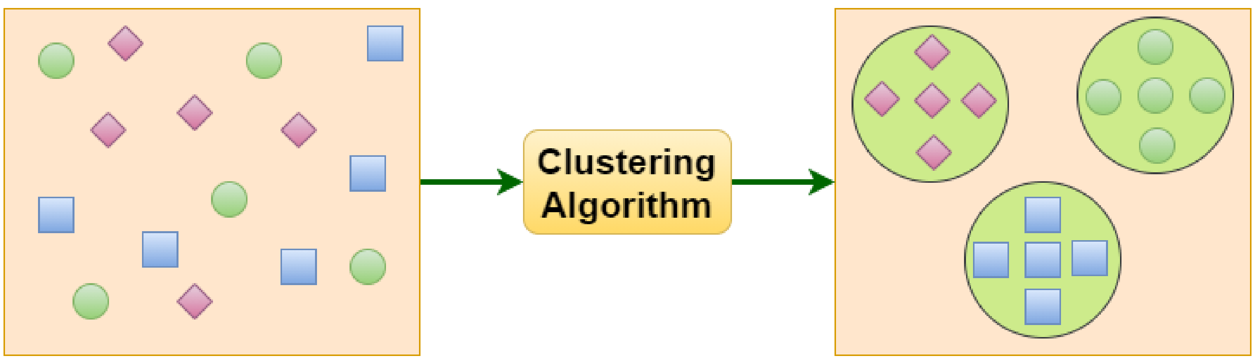

Figure 4. 1: Clustering

4.2.1 Examples of Clustering in Real-life Scenarios

The subsequent examples demonstrate how cluster analysis is utilized in different real-life circumstances.

Example 1: Retail Marketing

Retailers use clustering to identify groups of homes related to one another. For instance, a retailer might collect the following information on homes:

- Household size

- Household size
- Occupation of the Head of Household
- Distance from a neighbouring city

The variables might then be fed into a clustering algorithm, which could lead to the discovery of the following clusters:

Cluster 1: Small family, high spenders

Cluster 2: Larger family, high spenders

Cluster 3: Small family, low spenders

Cluster 4: Large family, low spenders

The company might then send customized adverts or sales letters to each household based on their likelihood of responding to specific advertisements.

Example 2: Streaming Services

Clustering analysis is frequently used by streaming services to find viewers with similar activities. For example, a streaming service could collect the following information about users:

- Minutes per day spent watching
- Total weekly watching sessions
- Per month, the number of unique shows viewed

A streaming service can utilize these parameters to cluster analysis and identify high- and low-usage consumers. So they can see who is paying for their ads.

Example 3: Sports Science

Data scientists use the clustering technique for sports teams to find connected players. Professional basketball teams, for example, might collect the following information about their players:

- Points per game
- Rebounds per game
- Assists per game
- Steals per game

They could then feed these factors into a clustering algorithm to find players related to one another, allowing them to practice alongside one another and complete specific drills based on their skills and shortcomings.

Example 4: Email Marketing

Many organizations use cluster analysis to locate consumers who are related to one another so that they may modify the emails they send to customers to optimise their earnings. For example, a company might collect the following information about its customers:

- The percentage of emails that are opened
- The number of clicks per email
- The amount of time spent examining emails

Using these metrics, a business may perform cluster analysis to find clients who use email in similar ways and adjust the types and frequency of emails they send to different clusters of customers.

Example 5: Health Insurance

Actuaries commonly used cluster analysis at health insurance companies to identify "clusters" of consumers who used their health insurance in specific ways. An actuary, for example, might collect the following information about households:

- The total number of doctor visits each year
- The total size of the household
- Per-household total number of chronic conditions
- The average age of the members of the household

These characteristics might then be fed into a clustering algorithm by an actuary to find similar homes. The health insurance firm might then calculate monthly premiums based on how often households in specific clusters are expected to use their insurance.

Example 6: Social Network Analysis (SNA)

It is the procedure of analyzing quantitative and qualitative social formation using Graph Theory and networks. The structure of social networks is mapped here in terms of nodes (unique personalities, persons, or other network entities) and edges or connections (relationships, communication, or connection) that connect them.

In this study, clustering algorithms are required to map and analyse the relationships and conflicts among persons, groups, computer networks, businesses, and other linked data entities. Clustering analysis could provide a visual and statistical assessment of such relationships, as well as a summary of social networks.

For example, assessing the position and grouping of actors in a network is necessary for comprehending it and its contributors. Individuals, professional groups, departments, organizations, or any large system-level unit could be the actors.

By a clustering technique, SNA could envisage the communication amid contributors and get intelligence regarding many groupings and roles at the network, for example, experts, bridges, and connectors, who have quarantined actors and many related data. It also displays where clusters exist, who is in them, at the network's core, and on the network's periphery.

Example 7: Recommendation engines

The recommendation system is a broadly utilized technique for presenting automatic personalized recommendations regarding services, products, and data, whereas collaborative filtering is a well-known recommendation technique.

In this technique, clustering presented a design of like-minded users. Thus, the calculation as data presented through numerous users is a boost to enhance collaborative filtering techniques' performance. And this could be executed by providing recommendations for diversified appliances.

For instance, the recommendation system is generally utilized on Flipkart, and Amazon, to suggest products and on Youtube to recommend songs of a similar genre.

Although dealing with large data clustering is a good first step in restricting the selection of fundamentally suitable neighbours in collaborative filtering algorithms, increasing the performance of sophisticated recommendation engines.

Fundamentally, all clusters would be allocated to particular desires based on consumers' selections that belong to the cluster. After that, consumers will obtain suggestions guessed at the cluster level within each cluster.

Example 8: Analyzing Biological Data, Analysis and Detection of Cancer Cells and Analysis of Medical Imaging

Biological data analysis is one way to combine analytical techniques with biological material. It allows for a deep and thorough knowledge of the correlations revealed as they relate to experimental results. Moreover, biological data is organized in networks or sequences, making clustering algorithms essential for detecting significant commonalities.

On the other hand, from the precedent few years, the utilization of investigation completed on the human genome with the increasing ability to accumulate various kinds of gene expression information guide to developing biological information study fast.

Clustering assists in extracting helpful facts from enormous datasets gathered in biology and different areas of life sciences, such as medicine, with the basic goal of presenting forecasts and descriptions of the structure of data.

Cancerous datasets can be found; a mixture of datasets containing both cancerous and non-cancerous data could be studied using clustering algorithms to comprehend the various qualities in the dataset, relying on the algorithms used to create the clusters.

4.3 Novel Clustering Aggregation (NCA) Algorithm for Clustering Fall Detection Data

The availability of a massive collection of clustering algorithms in the literature may confuse experts attempting to choose the suitable algorithm for a given dataset (Rosato, Altilio, and Panella, 2021). Also, no clustering algorithm could universally solve all problems, for example, cluster shape, noise or density (Bindra, and Mishra, 2019). To address this issue, this work proposed a novel clustering aggregation (NCA) using the combination of three clustering algorithms, namely K-Means, Expectation-Maximization (EM) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN). In the first step, these three clustering algorithms cluster the common dataset separately. The final clusters are extracted through a voting process among the data instances in the next step. It means data instance assigned to majority voted cluster by these three clustering algorithms. Thus, it enhances the accuracy of clustering and further decreases the clustering time compared to the unique clustering algorithms of the ensemble. Figure 4.2 demonstrates the flow diagram of the NCA algorithm.

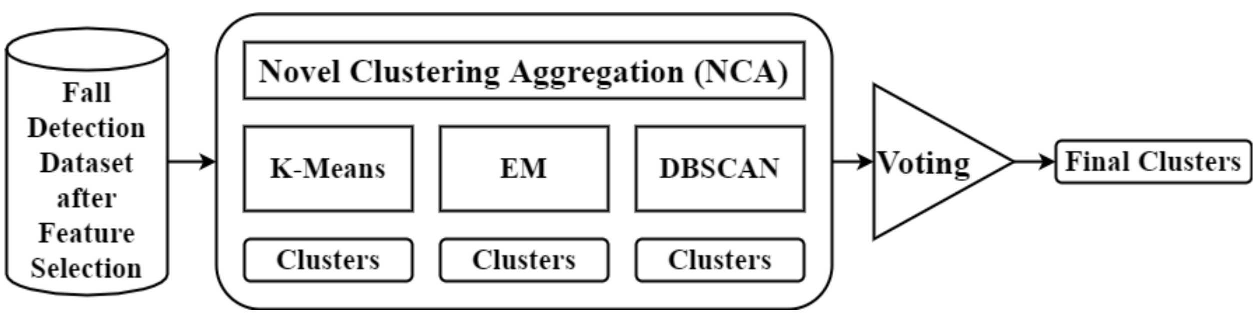

Figure 4. 2: Flow diagram of the NCA algorithm

Algorithm 4.1 shows Novel Clustering Aggregation (NCA) algorithm.

Algorithm 4.1: Novel Clustering Aggregation (NCA)

Input	**:**	Fall Detection Dataset (FDD)
Output	**:**	Assign each data instance to Majority Voted Cluster (MVC)
Step 1	**:**	Load Fall Detection Dataset with selected features
Step 2	**:**	Apply K-Means Clustering for FDD
Step 3	**:**	Apply EM Clustering for FDD
Step 4	**:**	Apply DBSCAN Clustering for FDD
Step 5	**:**	For each data instance DI from FDD
Step 6	**:**	Result1 = Get the K-Means cluster result for DI
Step 7	**:**	Result2 = Get the EM cluster result for DI
Step 8	**:**	Result3 = Get the DBSCAN cluster result for DI
Step 9	**:**	If(Result1 is equal to Result2), Then
Step 10	**:**	MVC = Result2
Step 11	**:**	Else If(Result1 is equal to Result3), Then
Step 12	**:**	MVC = Result3
Step 13	**:**	End if
Step 14	**:**	End For

4.3.1 K-Means Clustering

K-means Clustering is the most familiar algorithm of partitioning clustering, a type of clustering that partitions the data into non-hierarchical groups (Sinaga and Yang, 2020). It is alias the centroid based clustering. The dataset is separated into K sets in this algorithm, where K describes the number of pre-defined groups. The cluster's centre is created so that the distance between the data points of one group is small compared to another cluster centroid (Yuan and Yang, 2019). Figure 4.3 shows K-means clustering.

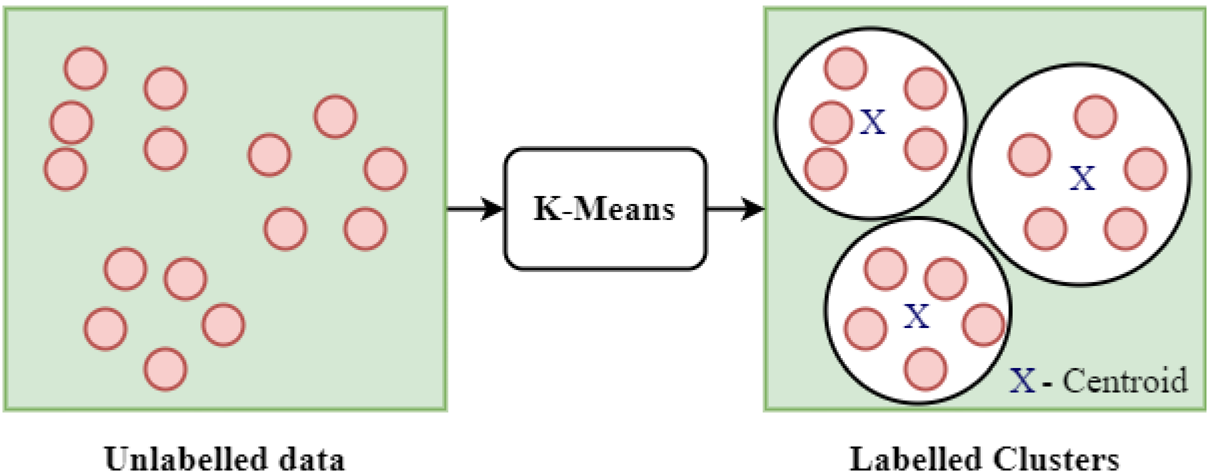

Figure 4. 3: K-Means clustering

K-Means clustering aims to divide n data instances into k clusters in which each data instance belongs to the group with the adjacent mean (Chakraborty, Paul, Das, and Xu, 2020). This algorithm creates precisely k different clusters. The goal of K-Means clustering is to reduce the entire intra-cluster difference, or the squared error function is shown in Eq. (4.1):

$$P = \sum_{j=1}^{k}\sum_{i=1}^{n} |xij - cj|^2 \tag{4.1}$$

Here P – objective function, n – instances count, k – clusters count and $|xij - cj|^2$ - Euclidean distance.

4.3.2 Expectation-Maximization (EM) Clustering

The most well-known Distribution Model-Based Clustering algorithm is the Expectation-Maximization Algorithm or EM algorithm for short. It is a method for maximal likelihood assessment in the existence of latent variables (Sammaknejad, Zhao, and Huang, 2019). The EM algorithm (Şimşek and Topaloglu, 2018) is an iterative method for those cycles among two modes. The estimation step or E-step is the first mode to assess missing or latent variables. The second mode, known as the maximization-step or M-step, optimises the model's parameters to describe the data.

- **E-Step:** Estimate the dataset's missing variables.
- **M-Step:** In the existence of data, maximize the model's parameters.

The EM algorithm has a wide range of applications, but it is probably best recognized in machine learning for its usage in unsupervised learning tasks like clustering and density estimation. Figure 4.4 shows EM Clustering.

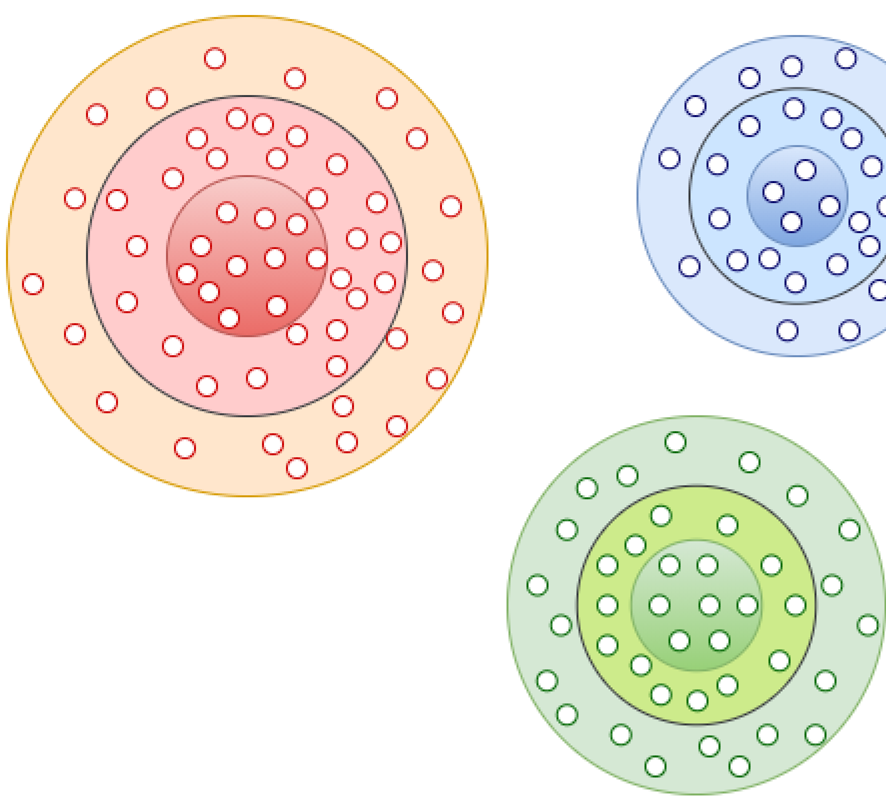

Figure 4. 4: EM clustering

4.3.3 Density-Based Spatial Clustering of Applications with Noise (DBSCAN) Clustering

The DBSCAN clustering links the highly-dense regions into clusters; the arbitrarily shaped distributions are created as long as the crowded area can be connected (Galán, 2019). This algorithm discovers various groups in the dataset and connects enormous densities' regions. Thus, the overcrowded areas of data space are divided from each other through sparser regions. In a density-based clustering algorithm, points are categorized as reachable points, core points and outliers, as follows:

- A point y is accessible from x if there is a route x1... xn with x1 = x and xn = y, where each xi+1 is straightly accessible from xi.
- A point x is a core point if minimum points are within distance ε (ε is the greatest radius of the neighbourhood from x). Those points are straightly reachable from x. By definition, no points are straightly reachable from a non-core point.
- All points that cannot reach from any other point are outliers.

Figure 4.5 shows DBSCAN clustering.

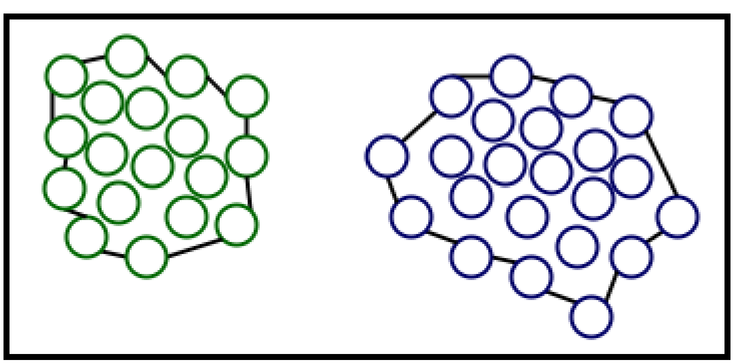

Figure 4. 5: DBSCAN clustering

4.3.4 Majority Voting Technique

A majority vote is more than half of the votes cast. The plan behind the majority vote is that the verdict of a committee is higher than the verdict of individuals. The voting-based clustering technique is that each data instance in a specified dataset votes for the cluster it belongs to and its equivalent collection in each other clustering outcome. The highest of these values denotes the best group for the data instance. It means that each data instance should cluster along with the opinion of the majority of the algorithms. Figure 4.6 is an illustration of the majority voting technique.

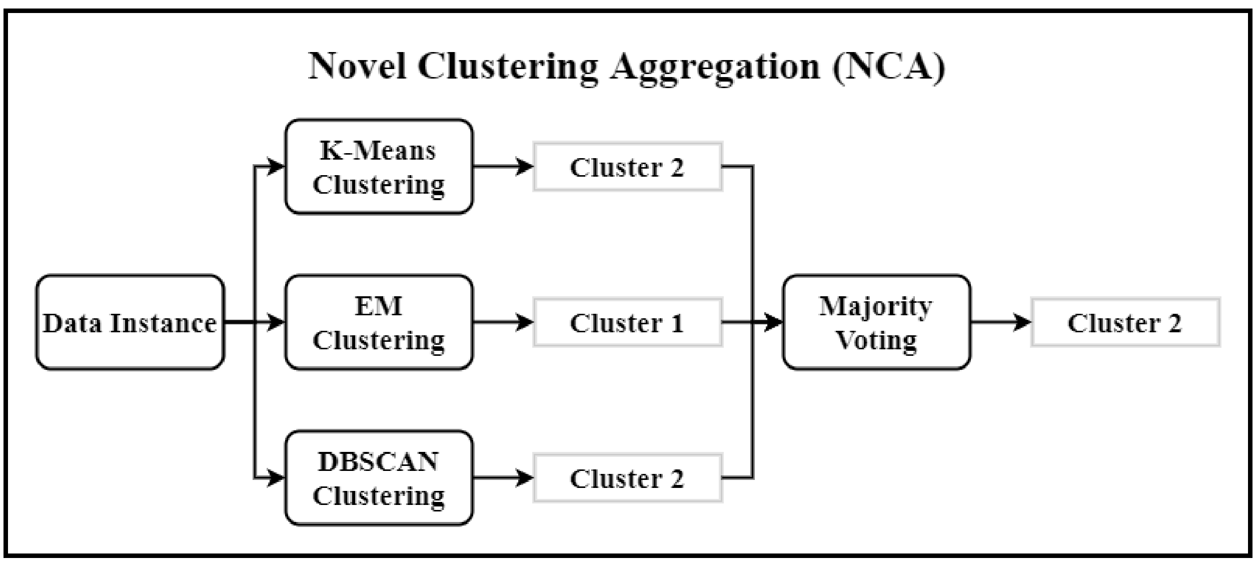

Figure 4. 6: Example of majority voting based clustering

In Figure 4.6, most of the three types of clustering agreed that the result is Cluster 2. Therefore, that data instance is assigned to cluster 2. In this way, the NCA algorithm provides better clustering results.

4.4 Process Description of Clustering

Table 4.1 shows the process description of clustering techniques.

Table 4. 1: Process Description of Clustering

Process	Descriptions
Load Fall Detection Dataset after Feature Selection	After MCFS algorithm, the dataset is extracted using the selected 5 features. This dataset given as input this proposed framework.
K-Means Clustering	In this process, the K-Means clustering algorithm is used to partition n observations into k clusters, with each observation belonging to the cluster with the closest mean, which serves as the cluster prototype.
Expectation-Maximization (EM) Clustering	In this process, the feature reducted dataset is clustered using the EM algorithm. It could be utilized to learn hidden variables probabilistic models. It accomplishes soft clustering, like the k-means algorithm, but with instances

	that are probabilistically in classes, when combined with a naive Bayes classifier
DBSCAN Clustering	In this process, the feature reducted dataset is clustered using the DBSCAN algorithm. It uses the premise that a cluster in data space is a continuous region of pinnacle density separated from other similar clusters through continuous regions of low point density to find unique groups/clusters in the data.
Novel Clustering Aggregation (NCA)	In this process, the feature reducted dataset is clustered using the aggregation of the above clustering techniques. It clusters the whole feature reducted dataset into two clusters.

4.5 Metrics for Clustering

Accuracy

To evaluate the performance of the NCA algorithm, clustering evaluation using classes mode is used. In this mode, clustering is executed in two steps to calculate accuracy. The first step ignores the class attribute and generates the clustering for a dataset. The next step induces the clustering for a dataset with a class attribute. If both cluster results are the same for a data instance, we conclude the data instance clustered correctly. Otherwise are incorrectly clustered. The formula of accuracy computation for clustering showed in Eq. (4.2).

$$Accuracy = \frac{Number\ of\ Correctly\ Clustered\ Instances}{Total\ number\ of\ instances} * 100 \tag{4.2}$$

Execution Time

Execution time is the time taken for clustering a dataset.

4.6 Results and Discussions

This section explains how to evaluate the performance of the NCA algorithm for clustering. Java has been used to generate this research experiment (version 1.8). For the experiment, a real-world dataset was obtained from the Internet. Clusters are generated using the NCA algorithm based on the selected features.

4.6.1 Clustering Results

This section analyses clustering results. The objective of clustering is to cluster the data instances into subsets to cluster related instances jointly while different instances belong to different groups.

Table 4.2 shows the clustering algorithms accuracy comparison for the fall detection dataset.

Table 4. 2: Clustering algorithms accuracy comparison for the fall detection dataset

Algorithm	Accuracy (in %)
K-Means Clustering	75.037
EM Clustering	75.184

DBSCAN Clustering	77.734
NCA	85.336

Figure 4.7 demonstrates the comparison of accuracy for the fall detection dataset. Compared with other types of clustering proposed, NCA accuracy is high.

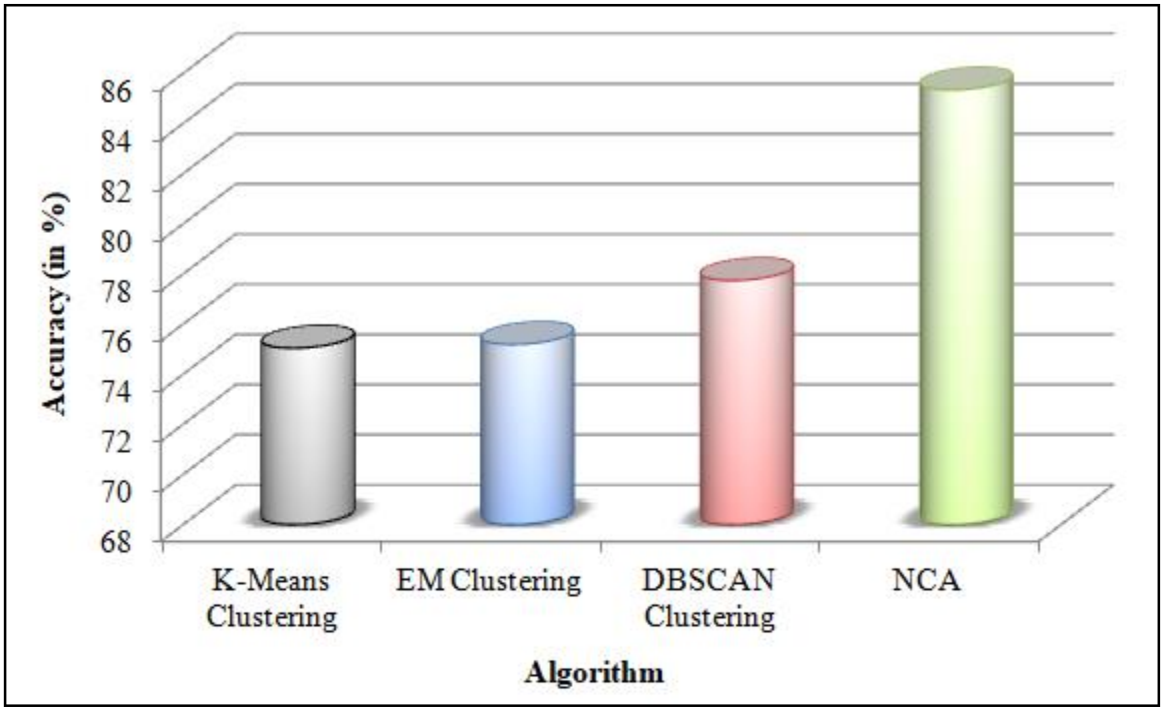

Figure 4. 7: Clustering Accuracy Comparison

Furthermore, Table 4.3 shows the execution time comparison for the fall detection dataset.

Table 4. 3: Execution time comparison for fall detection dataset

Algorithm	Execution Time (in ms)

K-Means Clustering	1162
EM Clustering	682
DBSCAN Clustering	252
NCA	273

Figure 4.8 demonstrates the comparison of execution time for the fall detection dataset. Compared with other types of clustering proposed, NCA takes less time for clustering.

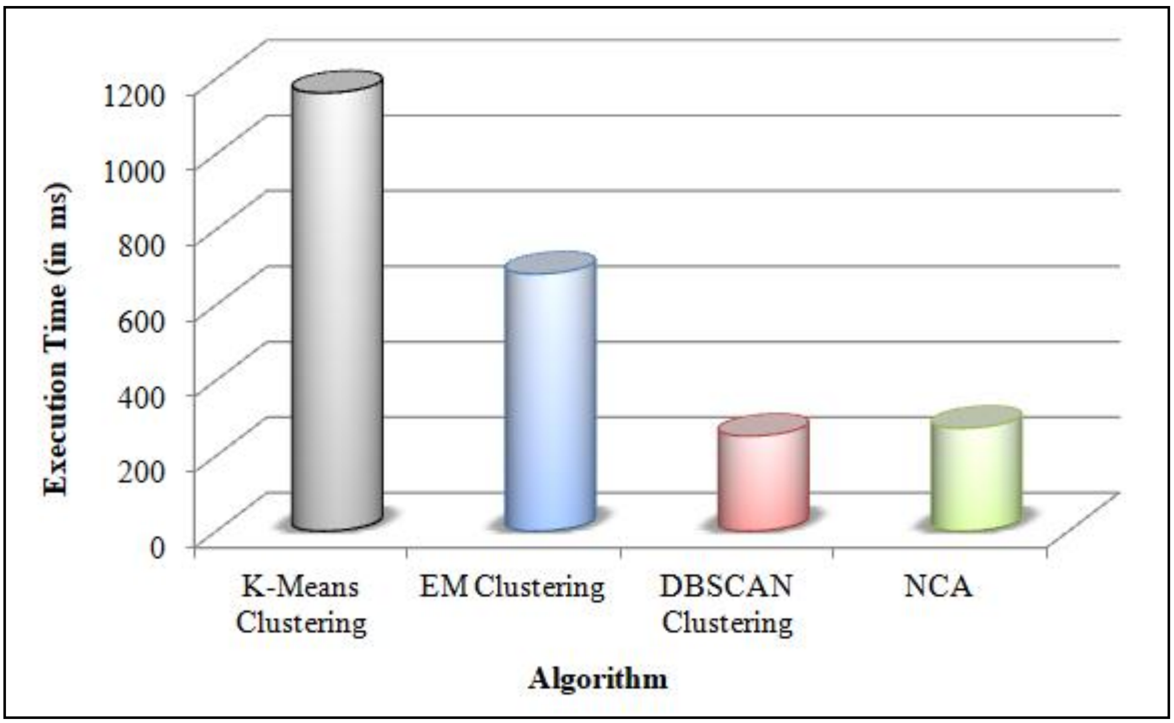

Figure 4. 8: Comparison of Execution Time for fall detection dataset

4.7 Hypothesis Justification

Table 4. 4: Hypothesis Justification

Hypothesis	Justification
H4	Novel Clustering Aggregation (NCA) algorithm provided efficient and accurate clustered datasets. These datasets are ready to improve the efficiency and performance of NSCP.

4.8 Conclusion

This chapter proposed a novel clustering aggregation (NCA) based on three clustering algorithms, including K-Means, EM and DBSCAN. In the first step, these three clustering algorithms cluster the common dataset separately. The final clusters are extracted through a voting process among the data instances in the next step.

CHAPTER 5

NSCP: Novel Stacking Classification and Prediction Algorithm for Fall Detection Data

5.1 Introduction

Identifying human fall plays a vital role in creating sensory-based alarm systems that help not only physical therapists minimize the after-effects of a fall but also save human lives. In general, the elderly suffer from various diseases, and fall activity is the most frequent condition for them at this time. By the way, this chapter reflects architecture for classifying fall events from the indoor natural functions of humans. Therefore, this chapter proposed a Novel Stacking Classification and Prediction (NSCP) algorithm to recognize the function of the elderly. A stacking classifier is an ensemble technique where the result of several classification algorithms is given as input to a meta-classifier for final classification. The stacking classification technique is the most efficient way to implement multiple classification problems. Complete individual classification techniques, generally referred to as base learning, can be integrated by creating a meta-classifier for the outcome prediction task. It can be done by stacking the results collectively from each classification technique and sending them as input to the meta-classifier. In the NSCP algorithm, the Repeated Incremental Pruning to Produce Error Reduction (RIPPER) classifier, Multinomial Logistic Regression (MLR) classifier and Dl4jMlpClassifier are used as base learning techniques, and the Naïve Bayes classifier is utilized as a meta-classifier.

5.2 Classification

Classification is the procedure of classifying a given data set; it can be done on structured and unstructured data (Charbuty and Abdulazeez, 2021). The procedure begins with calculating the class of data points given. Classes are frequently known as targets or labels. Estimating the mapping function from input variables to discrete output variables is called classification predictive modelling (Lai, Chen, Chen, Tang, and Lin, 2017). The primary aim is to determine which category or class the novel data belongs to. We would try to comprehend this with the example demonstrated in Figure 5.1.

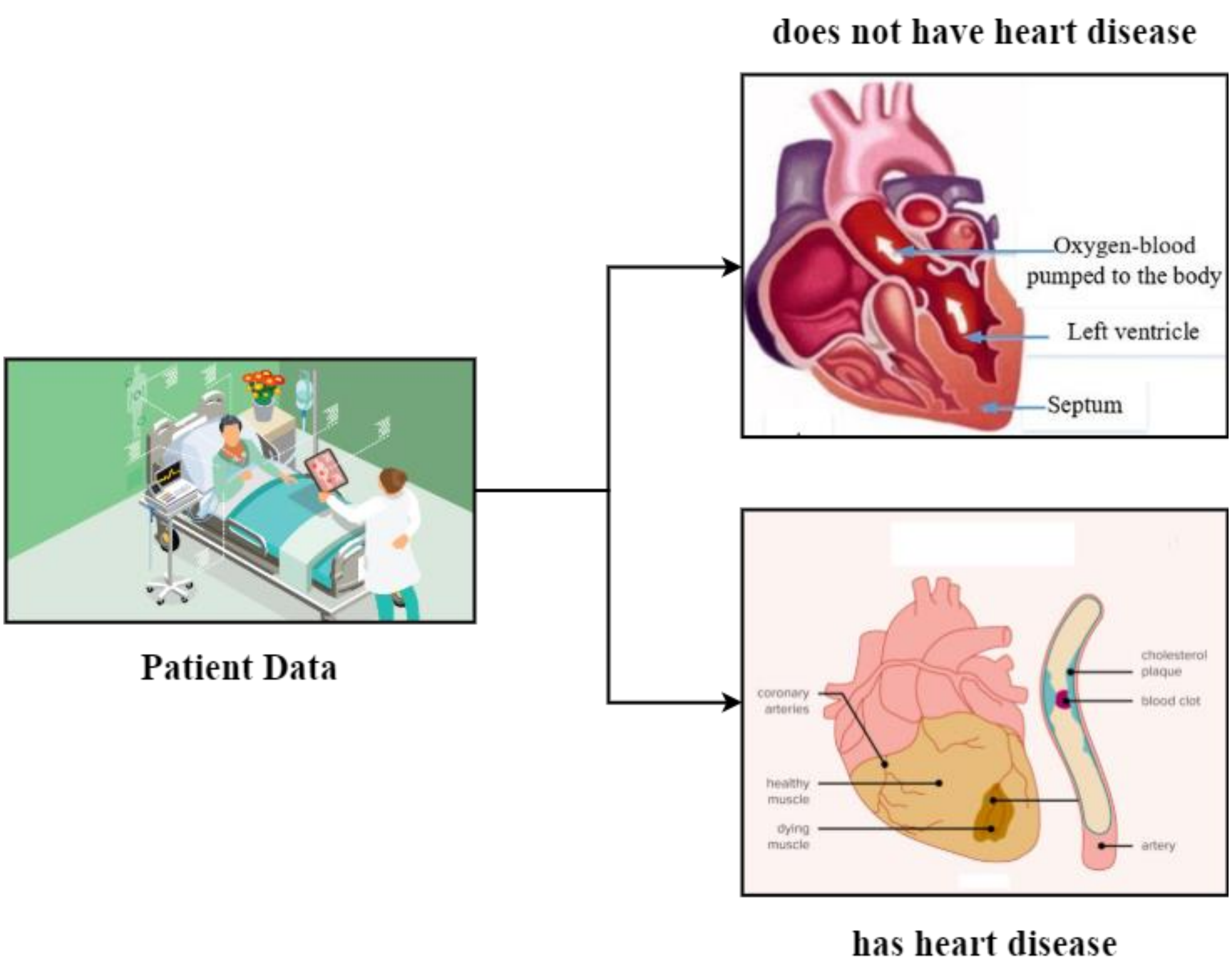

Figure 5. 1: Classification

Diagnosis of heart disease could be discovered like a classification issue or a binary classification because there could only be two classes: heart disease or no heart disease (Swathy and Saruladha, 2021). In this case, the classification algorithm requires training data to comprehend how the provided input variables are associated with the class. In addition, once the classification algorithm is trained precisely, it could be utilized to diagnose whether a particular patient has heart disease. Because classification is a kind of supervised learning, classes are further presented with input data. Let's look at how machine learning terminology is classified.

5.2.1 Machine Learning Terminologies for Classification

Classifier is a method utilized to map input data for a particular group.

Classification Model - The model forecasts or decides the input data specified for the training, predicting the class or type of information.

Feature - A feature is the unique scalable possessions of the observed event.

Binary classification is a kind of classification that has two effects, for example - false or true.

Multiple Class Classification - There are more than two classes in this classification; each sample in the multiple class taxonomy will be assigned to a single label.

Multiple Label Classification - This is a category of classification in which each sample is allocated to a set of labels.

Initialize - It must determine which classifier will be utilised.

Train Classifier - The fit (X, y) method is used by each classifier in the sci-kit to learn to fit the training model to train X and train label y.

Predict the label – The predict (X) method returns an anticipated label y for an unlabeled observation X.

Evaluate – This essentially means evaluating the model, classification report, accuracy score, etc.

5.2.2 Learner Types in Classification

Lazy Learners – Lazy learners merely save the training data and wait for the testing data (Panigrahi and Borah, 2020). Then, the classification is carried out utilizing the most closely related data from the training data that has been saved. Therefore, compared to eager learners, they have a larger prediction time—for instance, k-nearest neighbour and case-based reasoning.

Eager Learners – Eager learners make a classification method using the prearranged training dataset before receiving data for forecasting (Bafna and Saini, 2020). It should be capable of committing to a solitary hypothesis that would work for the whole space. Because of this, they get numerous times in training and fewer times for a forecast. E.g., Naive Bayes and Decision Tree.

5.3 Examples of Classification in Real-life Scenarios

5.3.1 Optical character recognition

The optical character recognition (OCR) issue, which identifies character codes from their images, is an instance of a classification issue (Bharath and Rani, 2017). It is an instance where there are many classes; there are many characters that we want to identify. Fascinating is the case when the characters are handwritten. People have a variety of handwriting styles; the characters can be written small or big, italic, with a pencil or pen, and can have multiple images associated with the same letter.

5.3.2 Face recognition

In the case of face recognition, the input is an image, classes need to be recognized, and the learning program must learn to relate face images to recognize (Ma, Liu, and Ren, 2019). This issue is more challenging than OCR since there are more classes, the input image is more significant, and the face is three-dimensional. In addition, the pose and lighting differences cause essential changes in the image.

5.3.3 Speech recognition

In speech recognition, input is audio, and classes are words that can be pronounced (Salazar, Kirchhoff, and Huang, 2019).

5.3.4 Medical diagnosis

In clinical diagnosis, the inputs are the related data regarding the patient, and the classes are diseases (Yadav and Jadhav, 2019). Entries include patient age, gender, precedent medical history and present symptoms. However, a few tests may not be applied to the patient, and therefore these inputs will miss.

5.3.5 Knowledge extraction

Classification rules could further be utilized to extract knowledge (Vidyarthi and Jain, 2020). The rule is a straightforward model that describes the data and looks at this model, and there is a detail of the data-based process.

5.3.6 Compression

Classification rules could be utilized for compression (Azar, Makhoul, Couturier, and Demerjian, 2020). By fitting a rule to the data, we obtain more exact details than the data, which needs small storage memory and small processing calculations.

5.3.7 More examples

Here are a few more instances of classification issues

a) The Intensive Care Unit (ICU) at the hospital assesses 17 variables such as blood pressure (BP), age, and recently admitted patients. Then, decide whether to put the patient in the ICU. Because of the high price of ICUs, high priority is prearranged merely for patients who can stay alive for a month or more. Such patients are labelled “low-risk patients” and others as “high-risk patients”. The issue is to make it a rule to classify a patient as a “high-risk patient” or a “low-risk patient” (Norton, Reddy, Subedi, Fabrizio, Wimmer, and Urrutia, 2021).

b) A credit card company gets hundreds of thousands of applications for new cards. Applications include data on various features such as annual salary, age, etc. The issue is that the rule of thumb is to classify applicants as eligible for credit, ineligible for credit, or in need of additional investigation (Purba, 2020)).

c) Astronomers list far-away objects in the sky utilizing digital images formed by special devices. Objects must be labelled as nebula, galaxy, star, etc. Unfortunately, the data is too noisy and too blurry. The issue is using a rule that can correctly name a far-away object (Jiménez, Torres, John, Triguero, 2020).

5.4 Novel Stacking Classification and Prediction Algorithm for Fall Detection Data

Stacking classification is an ensemble technique that merges multiple classifiers through a meta-classifier. The ensemble technique utilizes many classification algorithms to attain superior predictive performance than a single classification algorithm. Therefore, this paper proposed a novel stacking classification and Prediction algorithm (NSCP) for fall detection data. In NSCP, we used a stacking algorithm like a classifier to categorize the meta-features to attain the final class. Classifiers return the probability of belonging to a class (meta-features) in the first layer (RIPPER + MLR + Dl4jMlpClassifier). These meta-features are the input to the meta-classifier (Naive Bayes classifier) in the second layer. Ultimately, the classifier's output can be 0 (No fall, Happy Walking!) or 1 (Take a break, you tripped!) or 2 (Definite fall, Help is on the way). The NSCP algorithm's flow diagram is shown in Figure 5.2.

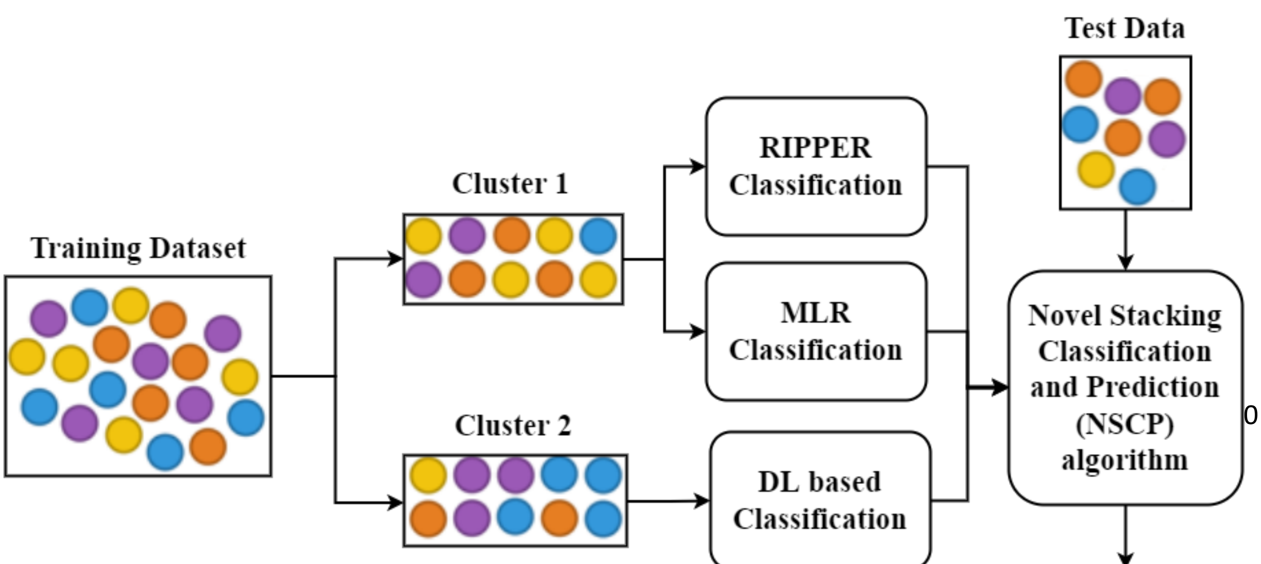

Figure 5. 2: Flow diagram of the NSCP algorithm

Algorithm 5.1 shows a novel stacking classification and Prediction algorithm (NSCP).

Algorithm 5.1: Novel Stacking Classification and Prediction algorithm (NSCP)

Input	**:**	Fall Detection Training Dataset Cluster 1 (FTC1), Fall Detection Training Dataset Cluster 2 (FTC2), Fall Detection Testing Dataset (FDTD)
Output	**:**	Fall Detection Predicted Result (FDPR)
Step 1	**:**	Classify FTC1 based on RIPPER classifier using weka
Step 2	**:**	Classify FTC1 using MLR classifier with weka
Step 3	**:**	Classify FTC2 using Dl4jMlpClassifier classifier with weka
Step 4	**:**	For each data instance DI from FDTD
Step 5	**:**	P1 = Predict DI using RIPPER classifier
Step 6	**:**	P2 = Predict DI using MLR classifier
Step 7	**:**	P3 = Predict DI using Dl4jMlpClassifier classifier

Step 8 **:** FDPR = Predict DI using Stacking classifier (RIPPER + MLR + Dl4jMlpClassifier as base classifier and Naïve Bayes as Meta-Classifier)

Step 9 **:** End For

5.4.1 RIPPER classifier

It refers to Repeated Incremental Pruning to Produce Error Reduction. The RIPPER algorithm is one of the classification algorithms based on the rule. It also uses the training dataset to generate a set of rules. It is the most broadly utilized rule induction algorithm (He, Zhang, Wang, Ren, Liu, Zhao, and Cheng, 2019).

RIPPER Algorithm Usage

1. It performs efficiently with unbalanced class partitions in data. If we possess many instances in data, most of them correspond to a specific class. If the remainder of the data belongs to various classes, we say the data has an uneven class distribution.
2. It performs better with noisy data as it utilizes a verification set to avert the model over-fitting.

Functioning of RIPPER Algorithm

Case 1: Training records be owned by merely two classes

It finds the majority class (which appears to be the most) in the given records and sets it as the default class. Consider the following example: If there are 50 entries and 35 belong to Class A with

15 to Class B, Class A becomes the default class. In the case of the other class, it tries to learn/get other rules to discover that class.

Case 2: There are more than two classes in the training records (numerous classes)

Regard all the available classes and arrange them in a specific order based on their frequency.

Regard the classes are prearranged as –

C1, C2, C3,......, Cn

C1 - minimum frequency

Cn - maximum frequency

The most frequent (Cn) class is taken as the default class.

How the rule evolved

In the primary case, it attempts to get the rules for C1 class records. C1 entries will be viewed as positive examples (+ ve), whereas entries from other classes will be treated as negative examples (-ve). A sequential covering algorithm is used to build rules that distinguish between the +ve and -ve instances. After that, the RIPPER algorithm attempts to differentiate the rules for C2 from other classes at this meeting. With Cn (default class) remaining, this procedure is continued until the stop criteria are met. Finally, the Ripper algorithm separates the rules from the minority class to the majority class.

The growing rule in the RIPPER algorithm

- The RIPPER algorithm utilizes the standard to a particular plan of rising rules. It initiates with an empty rule and continues to add the finest conjunct to the precursor of the rule.
- Metric was selected for the evaluation of links is FOIL's information gain. Therefore, utilising this ideal link is selected.
- When the rule starts to cover the negative (-ve) examples, the rule stops adding conjuncts.
- The novel rule is shortened based on its effectiveness in the validation set.

Rule Pruning Using RIPPER Algorithm

We require discovering whether or not a specific rule must be pruned. To decide this metric is utilized, which is –

(P-N) / (P+N)

P = the number of +ve examples in the rule's validation set.

N = the number of -ve examples in the rule's validation set.

- We compute the value of the above measurement for the original rule (before removing /adding) and the novel rule (after removing/adding) every time conjunction is removed or added.
- We could add/remove the conjunct if the novel rule's value is greater than the creative rule's. Otherwise, the conjunct will not be removed/inserted.
- Pruning is completed initiating from the rightmost end. For instance: Regard a rule – PQRS ---> Z, where P, Q, R, S are conjuncts, and Z is the class.

It will first remove the conjunct S and calculate the metric value. The conjunct S is removed if the metric's quality is improved. If the quality does not improve, the pruning is checked for RS, QRS, and other factors.

Creating a Rule Set in the RIPPER Algorithm

- Once a rule is obtained, all +ve and -ve instances covered through the rule are removed.
- The rule will then be attached to the rule set until the termination condition is violated. The stopping criterion which we could utilize are –
    A. Minimum Descriptive Length Policy: To transfer data from one node to another, you need the least number of bits. We desire the rule to be specified utilizing the least number of bits. If the novel rule raises the total descriptive length of the ruleset through d bits (by default, d is 64 bits), RIPPER ends by adding rules to the set of rules.
    B. Error Ratio - We would review the rule and compute its error rate (incorrect classification) in the validation set. The error ratio of a specific rule must not be surpassed 50%.

5.4.2 MLR classifier

Multinomial logistic regression (MLR) is a classification algorithm similar to the logistic regression for binary classification (Kayabol, 2019). In the logistic regression for binary classification, the classification task is to forecast the target class belonging to the binary type. Like Yes / No, 0/1, male / female. When it comes to MLR, the idea is to utilize logistic regression algorithms to forecast the target class (over two target classes).

As long as the probabilities for each target are calculated, the underlining technique will be the same as the logistic regression for the binary classification. Once the probabilities have been computed, convert them to a hot encoding and calculate the exact optimal weight using cross-entropy techniques during the training procedure.

Multinomial Logistic Regression Example

Using MLR, we can solve various kinds of classification issues. The trained model is utilized to forecast the target class from more than two target classes. Below are some examples to comprehend what type of issues we can solve using MLR

- Forecasting the type of Iris flower species

  Targets: a variety of species
- Assessing the acceptability of the car utilizing the given attributes

  Targets: Very Good, Good, Bad, Very Bad
- Forecasting the animal category utilizing the given animal attributes

  Targets: Camel, Horse, Cow, Deer

Advantages

- MLR is easy to execute and more proficient at interpreting and training.
- There is no assumption about the distribution of classes in the feature space.
- It not only presents a measure of how relevant a prediction (coefficient scale) is but further provides the direction (+ve or -ve) of its connection.
- It is more rapid in categorizing unknown records.

- It works well for numerous easy data sets with good accuracy and when the dataset is linearly divisible.
- It could be interpreted as sample coefficients as indicators of feature significance.

5.4.3 Dl4jMlpClassifier

Dl4jMlpClassifier is one of the DL classification algorithms that arbitrarily create deep feedforward neural networks, including convolutional neural networks (Lang, Bravo-Marquez, Beckham, Hall, Frank, 2019). Dl4jMlpClassifier is the core technology of WekaDeeplearning4j, which is built into a Weka package, thus creating Deeplearning4j methods available throughout the Weka environment. Dl4jMlpCassifier can be used for regression and classification by selecting the appropriate loss functions. The convolutional neural network (CNN) is a deep neural network used in deep learning. They are called space-inverting or shift-inverting artificial neural networks (SIANNs). They use the wide-weight configuration of convolution kernels or filters to slide with input, and present equivalent responses called hypothetical maps. Anti-intuitive, most CNNs are simply equivalent, unchanging, and contrary to translation. CNN regulates versions of multi-layer perceptrons (MLPs). Fully linked networks, or MLPs, are where each neuron in one layer is linked to genuine neurons in the subsequent layer. The “full connection” of these networks creates opportunities for data overload. Standard methods to avoid formalization or over-fitting contain: penalising parameters by training or trimming attachment. CNN adopts various regularisation techniques: taking advantage of the hierarchical pattern in information and increasing the difficulty of using minimal and easy patterns embedded in their filters. Therefore, even at the level of the connection problem, CNNs are inferior.

5.4.4 Stacking Classifier

A stacking classifier is a group technique in which the output of several classifiers is sent as input to a meta-classifier for final classification (Khraisat, Gondal, Vamplew, Kamruzzaman, and Alazab, 2020). The stacking classifier technique is the most efficient way to implement multiple classification problems. Complete individual classification techniques, commonly known as basic learning techniques, can be integrated by creating a meta-classifier for the final result prediction task. It can be accomplished by stacking the results collectively from each classification algorithm and sending them as input to the meta-classifier. For example, in the NSCP algorithm, the Naive Bayes classifier is used as a meta-classifier. Figure 5.3 shows the NSCP algorithm using the stacking classifier technique.

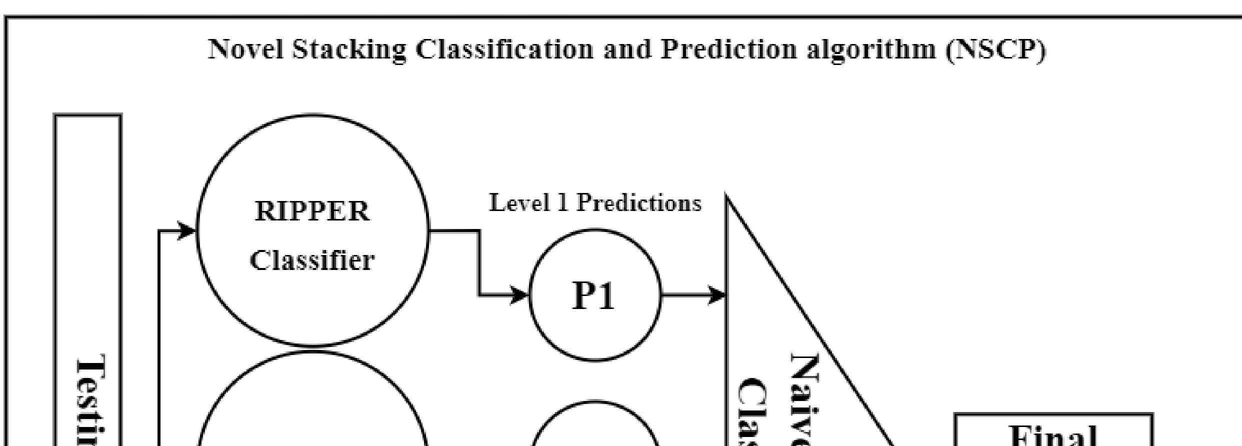

Figure 5. 3: NSCP algorithm using the stacking classifier technique

The Naïve Bayes algorithm is a supervised learning algorithm using the Bayes theorem that is also used to solve classification problems (Salmi, and Rustam, 2019). The Naïve Bayes classifier is one of the most accessible and efficient classifiers to help build rapid ML techniques to create rapid forecasting. It is a probability classifier, i.e. it forecasts the basis of an object's probabilities. The Naïve Bayes algorithm consists of two terms, Naïve and Bayes, which could be explained as:

Naïve: It is known as Naïve since a particular attribute is excluded from the occurrence of other features. For example, if the fruit is recognized based on shape, colour and taste, the red, spherical and sweet fruits are identified as apples. Therefore, each attribute helps to recognize which is an apple devoid of trusting each other.

Bayes: It is known as Bays since it relies on the principle of the Bayes theorem.

The Bayes theorem, often known as Bayes' law, is a mathematical formula for calculating the probability of a hypothesis based on past knowledge. The condition's probability determines it.

5.5 Process Descriptions of NSCP algorithm

Table 5.1 shows the process description of NSCP algorithm.

Table 5. 1: Process Descriptions of NSCP algorithm

Process	Descriptions
Load Training Dataset	After NCA algorithm implementation, 2 clusters are generated. From these 2 clusters, 75% data taken to generate 2 training datasets (It means 75% of data from 1st clster used as 1st training dataset and 75% of data from 2nd cluster used as 2nd training dataset) and remaining 25% data (obtained from both clusters) taken as testing dataset. In this process, 2 training datasets given as input to the NSCP algorithm.
Load Testing Dataset	In this process, a testing dataset is given as input to the NSCP algorithm. It is necessary for measure the performance of all classifier algorithms
Machine Learning based Classification and Prediction	In this process, Machine Learning classification algorithm especially RIPPER and MLR classifier used for classification with 1st training dataset and prediction of elderly fall in testing dataset.

Deep Learning based Classification and Prediction	In this process, Deep Learning classification algorithm especially Dl4jMlpClassifier used for classification with 2nd training dataset and and prediction of elderly fall in testing dataset.
Novel Stacking Classification and Prediction (NSCP) algorithm	In this process, a novel stacking classification algorithm using the combination of RIPPER, MLR, Dl4jMlpClassifier with Naive Bayes classifier used for classification with (both training datasets combined as one training dataset) and prediction of elderly fall in testing dataset.
Elderly Patients Fall Prediction for single person	In this process, the NSCP algorithm predicts elderly fall for new data especially for single person. If a user wants to predict fall for single person, he or she give any values like distance, pressure, HRV and SPO2, the NSCP algorithm predicts and provide elderly patients fall prediction results efficiently.
Predict fall for multiple persons	In this process, the NSCP algorithm predicts elderly fall for new dataset especially for multiple persons. If a user wants to predict fall

	for multiple persons, he or she give any datasets including like distance, pressure, HRV and SPO2 features, the NSCP algorithm predicts and provide elderly patients fall prediction results efficiently.

5.6 Metrics for Classification

Accuracy

Accuracy is the fraction of the whole quantity of accurate predictions and the whole quantity of forecasts. Accuracy denoted the % of each correctly forecasted data point, shown in Eq. (5.1).

$$Accuracy = \frac{(\mathrm{TP} + \mathrm{TN})}{(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})} \tag{5.1}$$

Precision

Precision is the ratio between the True Positives and all the Positives. Precision is a measure of exactness or quality or positive predictive value, shown in Eq. (5.2).

$$\mathrm{Precision} = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FP})} \tag{5.2}$$

Recall

Recall is the measure of accurate identification of true positives. The recall is further called sensitivity, shown in Eq. (5.3).

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{5.3}$$

F-Measure

The harmonic mean of precision and recall is the F-measure shown in Eq. (5.4).

$$\text{F} - \text{Measure} = 2.\frac{(\text{Precision} \ * \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{5.4}$$

5.7 Results and Discussions

This section explains how to evaluate the performance of the NSCP algorithm for fall detection and prediction. Java has been used to generate this research experiment (version 1.8). For the experiment, a real-world dataset was obtained from the Internet. The NSCP algorithm is utilized to discover and predict the fall.

5.7.1 Classification Results

This section analyses classification results. Classification is the procedure of classifying a given set of data; it could be done on structured and unstructured data. Table 5.2 shows the accuracy comparison of RIPPER, MLR, and Dl4jMlpClassifier and NSCP algorithms.

Table 5. 2: Accuracy comparison of RIPPER, MLR, Dl4jMlpClassifier and NSCP algorithms

Algorithm	Accuracy (in %)
RIPPER	80.0
MLR	88.0

Dl4jMlpClassifier	93.0
NSCP	99.0

Figure 5.4 demonstrates the accuracy comparison for the fall detection dataset. Compared with other classification algorithms, the accuracy of the NSCP algorithm is highest. It is because the NSCP algorithm is used the ensemble and stacking classifier approach. This approach boosts the performance of the NSCP algorithm. Therefore, the NSCP algorithm provides the highest accuracy.

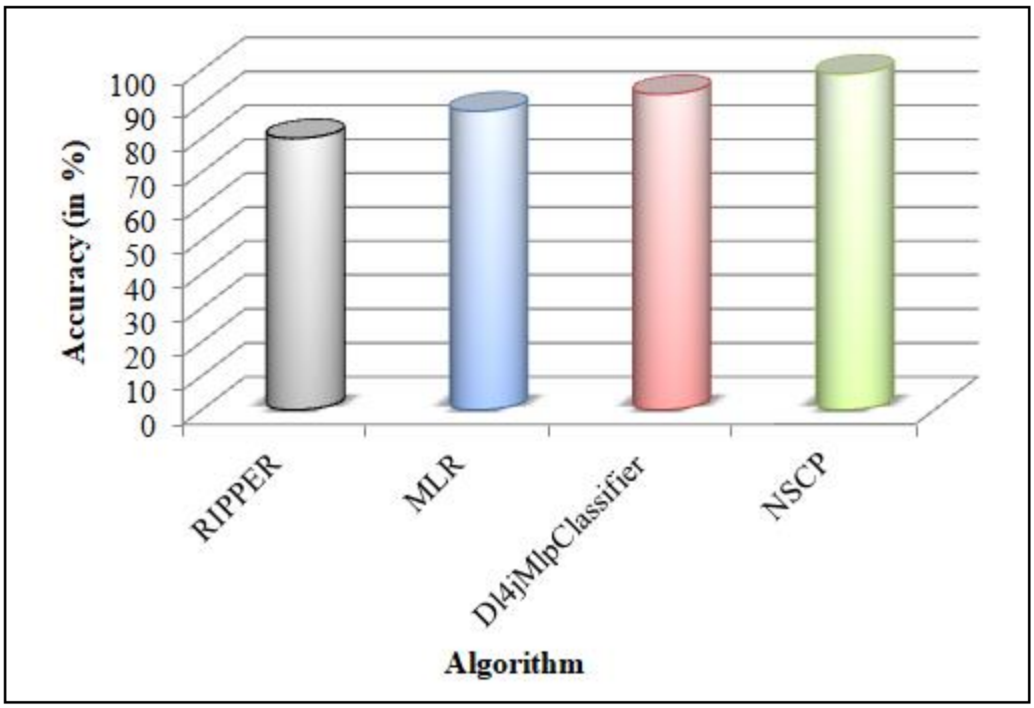

Figure 5. 4: Accuracy comparisons of RIPPER, MLR, and Dl4jMlpClassifier and NSCP algorithms

Table 5.3 demonstrates the Precision comparison of the RIPPER, MLR, and Dl4jMlpClassifier and NSCP algorithms.

Table 5. 3: Precision comparison of RIPPER, MLR, and Dl4jMlpClassifier and NSCP algorithms

Algorithm	Precision (in %)
RIPPER	80.50
MLR	81.60
Dl4jMlpClassifier	91.60
NSCP	97.69

Figure 5.5 demonstrates the comparison of precision for the fall detection dataset. Compared with other classification algorithms precision of the proposed NSCP algorithm is enormous. Since the NSCP algorithm is used, both ML (RIPPER, MLR, Naïve Bayes) and DL (Dl4jMlpClassifier) algorithms. Here, DL provides high accuracy. But ML gives lesser accuracy than DL. Also, DL requires enormous data. But ML can train on more minor data. By this, we can know ML solves the defect of DL and DL solves the defect of ML. The NSCP algorithm uses both techniques, which provide the highest precision.

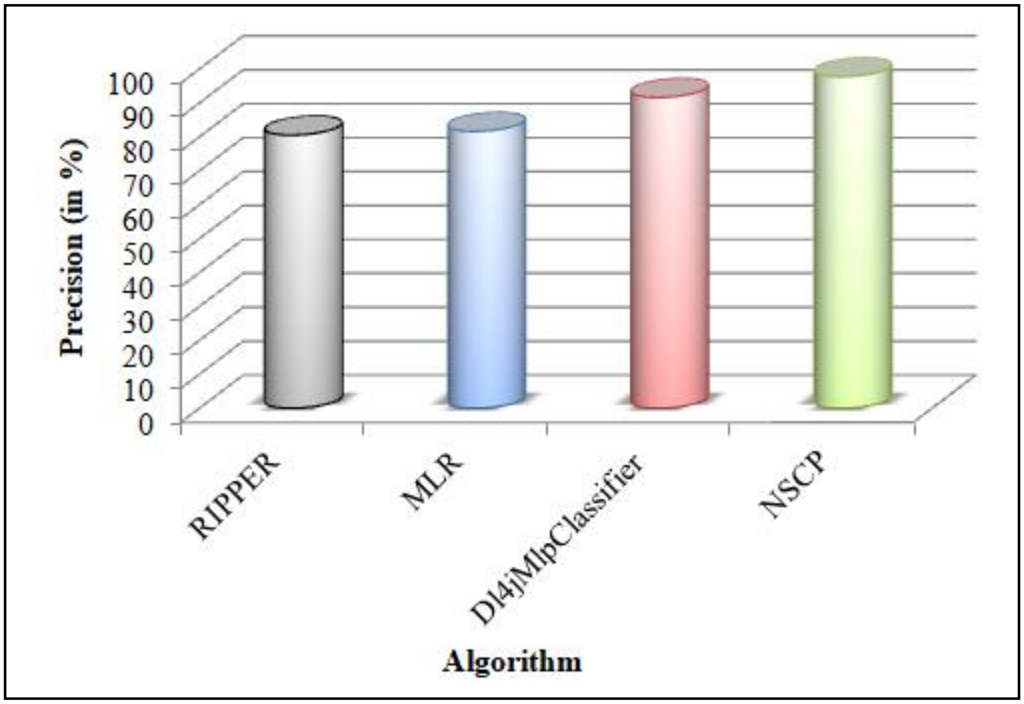

Figure 5. 5: Precision comparisons of RIPPER, MLR, and Dl4jMlpClassifier and NSCP algorithms

Table 5.4 shows the Recall comparison of RIPPER, MLR, and Dl4jMlpClassifier and NSCP algorithms.

Table 5. 4: Recall comparison of RIPPER, MLR, and Dl4jMlpClassifier and NSCP algorithms

Algorithm	Recall (in %)
RIPPER	89.64

MLR	89.97
Dl4jMlpClassifier	94.98
NSCP	98.10

Figure 5.6 demonstrates the comparison of recall for the fall detection dataset. Compared with RIPPER, MLR, and Dl4jMlpClassifier algorithms, recall of the NSCP algorithm is high.

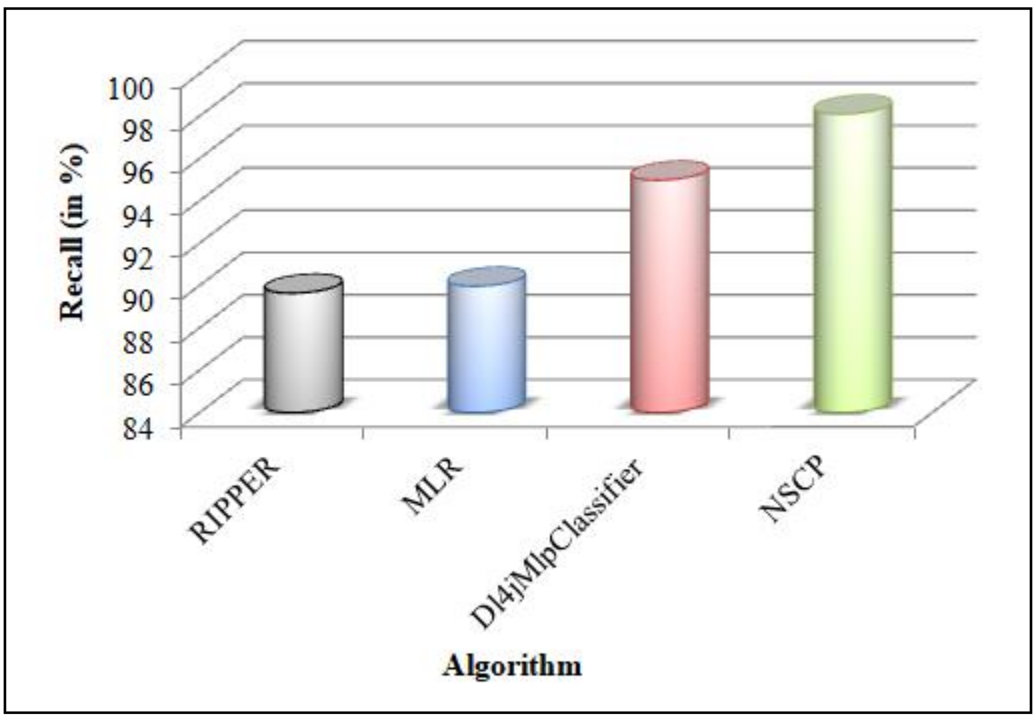

Figure 5. 6: Recall comparisons of RIPPER, MLR, and Dl4jMlpClassifier and NSCP algorithms

Furthermore, Table 5.5 demonstrates the F-Measure comparison of RIPPER, MLR, and Dl4jMlpClassifier and NSCP algorithms.

Table 5. 5: F-Measure comparison of RIPPER, MLR, and Dl4jMlpClassifier and NSCP algorithms

Algorithm	F-Measure (in %)
RIPPER	84.82
MLR	85.58
Dl4jMlpClassifier	93.26
NSCP	97.90

Figure 5.7 demonstrates the comparison of F-Measure for the fall detection dataset. Compared with other classification algorithms F-Measure of the proposed NSCP algorithm is high.

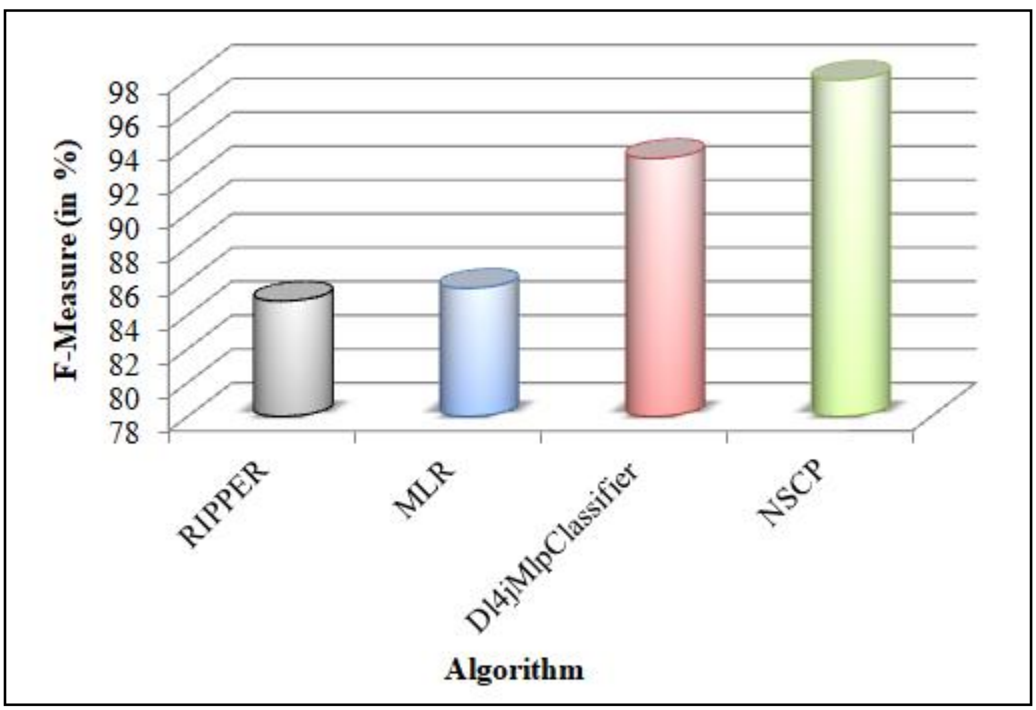

Figure 5. 7: F-Measure comparisons of RIPPER, MLR, and Dl4jMlpClassifier and NSCP algorithms

Table 5.6 shows the accuracy comparison of existing classification algorithms namely, CNN (Fakhrulddin, Fei, and Li, 2017), DNN (Pires, Garcia, Pombo, Flórez-Revuelta, Spinsante, and Teixeira, 2018), CNN-LSTM (Xu, He, and Zhang, 2019) (using MobiAct dataset for fall detection) with proposed NSCP algorithms (using cStick dataset for fall detection).

Table 5. 6: Accuracy comparison of existing CNN, DNN, CNN-LSTM with proposed NSCP algorithm

Algorithm	Accuracy (in %)
CNN	92.3
DNN	89.51
CNN-LSTM	98.98
NSCP	99.0

Figure 5.8 demonstrates the accuracy comparison of the existing CNN, DNN, CNN-LSTM algorithms with proposed NSCP algorithm. Compared with existing classification algorithms, the

accuracy of the NSCP algorithm is highest. It is because the NSCP algorithm is used the ensemble and stacking classifier approach. This approach boosts the performance of the NSCP algorithm. Therefore, the NSCP algorithm provides the highest accuracy.

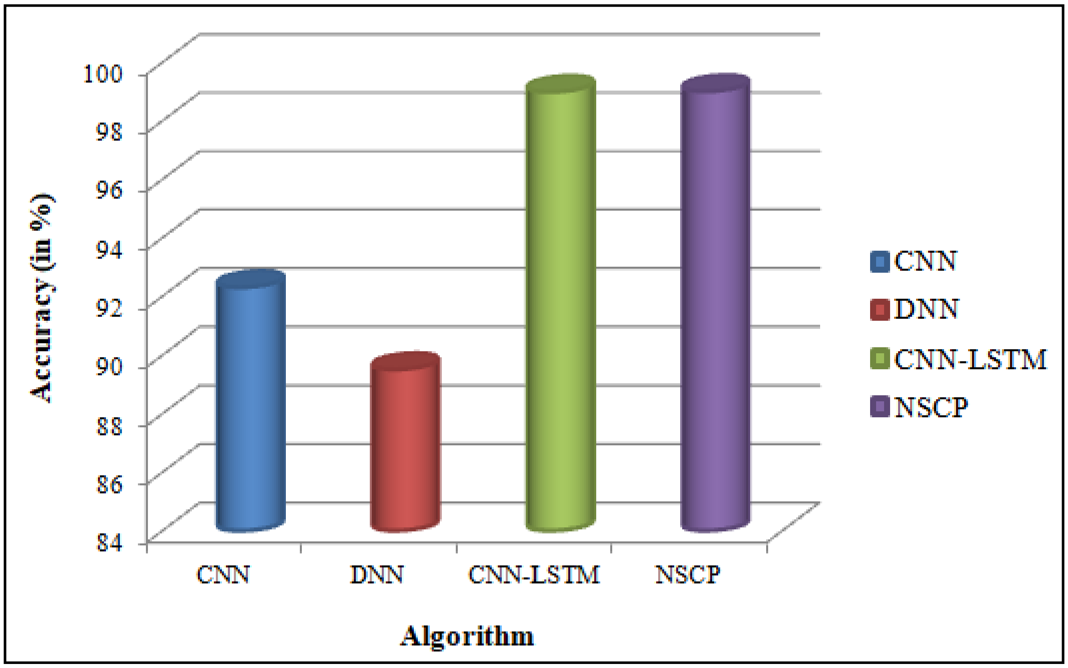

Figure 5. 8: Accuracy comparison of existing CNN, DNN, CNN-LSTM with proposed NSCP algorithm

5.8 Testing

Validation Testing

Validation testing is the procedure of determining whether the developed and tested software meets the requirements of the client or user. The reasoning or scenarios for the business requirements must be thoroughly tested. An application's critical functionality has been thoroughly tested.

The screenshot of validation testing is shown in Figure 5.8. It verifies whether or not the training dataset has been loaded.

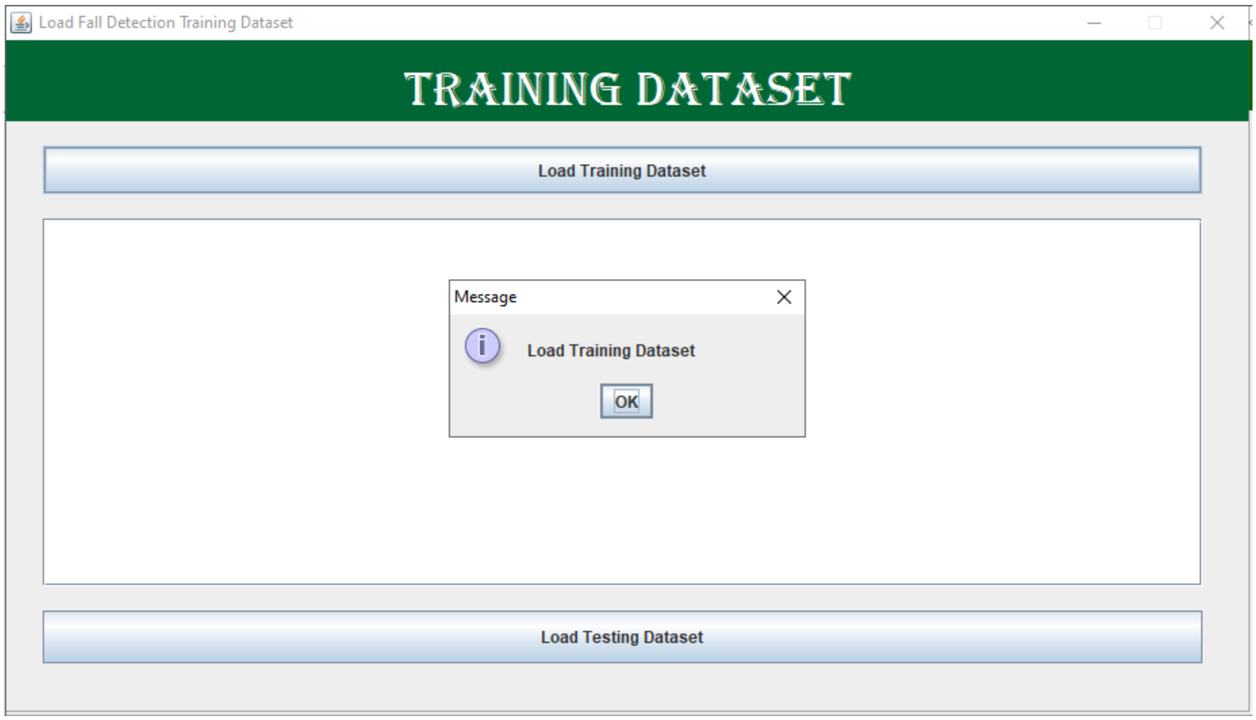

Figure 5. 9: Screen - 1 for Validation Testing

Another screenshot of validation testing is shown in Figure 5.9. Again, it checks whether the testing dataset has been loaded.

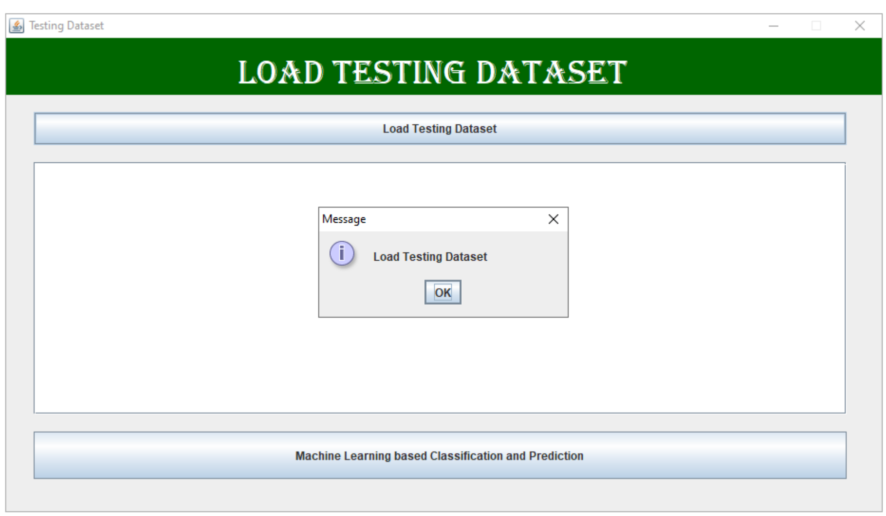

Figure 5. 10: Screen - 2 for Validation Testing

5.9 Hypothesis Justification

Table 5. 7: Hypothesis Justification

Hypothesis	Justification
H1	It has been predicted that the elderly fall by implementing a novel stacking classification and prediction (NSCP) algorithm.
H2	Using the training dataset, the NSCP algorithm was best used to diagnose how best to map incoming data events to specific class labels.
H5	The guidance of this method increased the accuracy of the fall prediction.

5.10 Conclusion

Fall is a moderately general incidence in older people, which could have dramatic health effects. Falls could cause physical injuries, containing fractures, head injuries, or severe decay. The impact of the falls will be dramatic on specific populations, prompting the search for enhanced ways to mitigate and respond to falls. Therefore, this chapter proposed a Novel Stacking Classification and Prediction (NSCP) algorithm to recognize the activities of older people. Here, RIPPER, MLR and Dl4jMlpClassifier algorithms are used as a base classifier, and the Naive Bayes classification algorithm is used as Meta-classifier in the NSCP algorithm.

CHAPTER 6

Conclusion and Future Enhancement Works

6.1 Introduction

The results and achievements of the present thesis are quickly summarised in this chapter, which also serves as the study's last chapter. The limitations of the proposed work are described, and probable future works are also discussed.

6.2 Summary of the Research

This thesis makes a significant contribution by proposing a novel fall detection and prediction method. This work has three primary phases: feature selection, clustering, and classification.

A feature selection algorithm is applied to the dataset in the first phase. Then, the Multi-strategy Combination-based Feature Selection technique is used to choose the best features from the dataset.

Clustering is the second phase, and it utilizes a novel clustering aggregation algorithm. The clusters are created using a newly created dataset after feature selection. New training and testing datasets are generated for fall detection and prediction based on these clusters.

The last phase is classification. In this phase, the novel training dataset is trained to utilize a Novel Stacking Classification and Prediction algorithm, and also it detects the fall events of the testing dataset. Furthermore, given new data are predicted as definite fall, no fall detected, person slipped or tripped using an NSCP algorithm.

Precision, recall, f-measure, and accuracy have all been used to assess the proposed algorithms. Experiments and findings demonstrate that the proposed work accurately predicts fall events.

6.3 Findings

The following results can be drawn from the experiments conducted in Chapter 6: The cStick fall prediction dataset was used in the tests.

- The performance of the NSCP algorithm is superior to other classification algorithms like RIPPER, MLR, and Dl4jMlpClassifier.
- Furthermore, the performance of the NCA algorithm is superior to other clustering algorithms like K-Means, EM and DBSCAN.
- In addition, the performance of the MCFS algorithm is superior to other feature selection algorithms like IG-Based, FS-Based, MMN-Based, CC-Based, and MAD-Based feature selection algorithms.
- Also, The NSCP algorithm's accuracy is better by having more training samples.

6.4 Proposed work's limitations

The following are the limits of the proposed work:

- The proposed MCFS algorithm takes more time than Fisher Score-based feature selection algorithm to select optimal features. Therefore, instead of MCFS, use any alternative feature selection method based on optimization.
- The prediction results of the NSCP algorithm are solely using the selected features. Therefore, it is a key issue for this algorithm.

6.5 Recommendations for future works

In comparison to existing methods, validation and testing revealed that the proposed method efficiently identified and predicted falls. The algorithm, however, can yet be enhanced. The recommendations for enhancing fall detection and prediction effectiveness are listed below.

- To expand the dataset size
- To choose features using a meta-heuristic algorithm
- To detect and predict falls using a Novel Classification algorithm

6.6 Summary

Fall detection and prediction algorithm are presented in this research. A combination of machine learning and deep learning algorithms is used for fall detection. The cStick dataset is used to test the proposed work's effectiveness and efficiency. The proposed work was assessed using a set of evaluation criteria. Several current algorithms were compared to the proposed research endeavour.

The proposed NSCP algorithm is compared to the classification algorithms RIPPER, MLR, and Dl4jMlpClassifier. The results demonstrate that RIPPER has 80 per cent detection accuracy, MLR has 88 per cent detection accuracy, Dl4jMlpClassifier has 93 per cent detection accuracy, and the suggested NSCP method has 99 per cent cent cent detection accuracy. The experiment demonstrates that the NSCP classifier can accurately predict fall events.

By analysing the summary of the proposed algorithm and its outcomes, it is possible to conclude that the thesis's goals have been met. A few feasible future works were given in this chapter for further improvements in the fall detection and prediction issue.

REFERENCES

Abd Mutalib, S. S. S., Satari, S. Z., & Yusoff, W. N. S. W. (2021, July). Comparison of Robust Estimators for Detecting Outliers in Multivariate Datasets. In Journal of Physics: Conference Series (Vol. 1988, No. 1, p. 012095). IOP Publishing.

Abdul Hamid, N. H. A., & Mustopa, N. N. (2020). PRESIMS: Requirements and design for preschool management system using usability theory. ESTEEM Academic Journal, 16, 51-64.

Abdullah, S., Ismail, M., & Ahmed, A. N. (2019). Multi-layer perceptron model for air quality prediction. Malaysian Journal of Mathematical Sciences, 13, 85-95.

Aksu, D., Üstebay, S., Aydin, M. A., & Atmaca, T. (2018, September). Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm. In International symposium on computer and information sciences (pp. 141-149). Springer, Cham.

Alam, F., Faulkner, N., & Parr, B. (2020). Device-free localization: A review of non-RF techniques for unobtrusive indoor positioning. IEEE Internet of Things Journal, 8(6), 4228-4249.

Albert, M. V., Kording, K., Herrmann, M., & Jayaraman, A. (2012). Fall classification by machine learning using mobile phones. PloS one, 7(5), e36556.

AL-OBAIDI, S. A. L. A. H. (2020). Privacy aware human action recognition: an exploration of temporal salience modelling and neuromorphic vision sensing (Doctoral dissertation, University of Sheffield).

Alzubi, J., Nayyar, A., & Kumar, A. (2018, November). Machine learning from theory to algorithms: an overview. In Journal of physics: conference series (Vol. 1142, No. 1, p. 012012). IOP Publishing.

Apple Watch. (2022). Helping your patients identify early warning signs. Retrieved March 15, 2022, from https://www.apple.com/in/healthcare/apple-watch/

Ayyadevara, V. K. (2018). Random forest. In Pro Machine Learning Algorithms (pp. 105-116). Apress, Berkeley, CA.

Azar, J., Makhoul, A., Couturier, R., & Demerjian, J. (2020). Robust IoT time series classification with data compression and deep learning. Neurocomputing, 398, 222-234.

Bafna, P. B., & Saini, J. R. (2020). On Exhaustive Evaluation of Eager Machine Learning Algorithms for Classification of Hindi Verses. International Journal of Advanced Computer Science and Applications, 11(2).

Bagala, F., Becker, C., Cappello, A., Chiari, L., Aminian, K., Hausdorff, J. M., ... & Klenk, J. (2012). Evaluation of accelerometer-based fall detection algorithms on real-world falls. PloS one, 7(5), e37062.

Baishya, K., & Samalia, H. V. (2020). Extending unified theory of acceptance and use of technology with perceived monetary value for smartphone adoption at the bottom of the pyramid. International Journal of Information Management, 51, 102036.

Balli, S., Sağbaş, E. A., & Peker, M. (2019). Human activity recognition from smart watch sensor data using a hybrid of principal component analysis and random forest algorithm. Measurement and Control, 52(1-2), 37-45.

Batista, E., Moncusi, M. A., López-Aguilar, P., Martínez-Ballesté, A., & Solanas, A. (2021). Sensors for Context-Aware Smart Healthcare: A Security Perspective. Sensors, 21(20), 6886.

Beddiar, D. R., Nini, B., Sabokrou, M., & Hadid, A. (2020). Vision-based human activity recognition: a survey. Multimedia Tools and Applications, 79(41), 30509-30555.

Béjot, Y., & Yaffe, K. (2019). Ageing Population: A Neurological Challenge. Neuroepidemiology, 52(1-2), 76-78.

Berger, S., Häckel, B., & Häfner, L. (2021). Organizing self-organizing systems: A terminology, taxonomy, and reference model for entities in cyber-physical production systems. Information Systems Frontiers, 23(2), 391-414.

Bharath, V., & Rani, N. S. (2017, June). A font style classification system for English OCR. In 2017 International Conference on Intelligent Computing and Control (I2C2) (pp. 1-5). IEEE.

Bhatti, A., Siyal, A. A., Mehdi, A., Shah, H., Kumar, H., & Bohyo, M. A. (2018, February). Development of cost-effective tele-monitoring system for remote area patients. In 2018 International Conference on Engineering and Emerging Technologies (ICEET) (pp. 1-7). IEEE.

Binda, J., Yuan, C. W., Cope, N., Park, H., Choe, E. K., & Carroll, J. M. (2018, May). Supporting effective sharing of health information among intergenerational family members. In Proceedings of the 12th eai international conference on pervasive computing technologies for healthcare (pp. 148-157).

Bindra, K., & Mishra, A. (2019). Effective data clustering algorithms. In Soft Computing: Theories and Applications (pp. 419-432). Springer, Singapore.

Cahoolessur, D. K., & Rajkumarsingh, B. (2020). Fall Detection System using XGBoost and IoT. R&D Journal, 36, 8-18.

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. Neurocomputing, 300, 70-79.

Cai, W., Qiu, L., Li, W., Yu, J., & Wang, L. (2019, August). Practical Fall Detection Algorithm based on Adaboost. In Proceedings of the 2019 4th International Conference on Biomedical Signal and Image Processing (ICBIP 2019) (pp. 117-121).

Carver, L. F., Beamish, R., Phillips, S. P., & Villeneuve, M. (2018). A scoping review: Social participation as a cornerstone of successful aging in place among rural older adults. Geriatrics, 3(4), 75.

Casals, E., Cordero, J. A., Dauwalder, S., Fernández, J. M., Solà, M., Vargiu, E., & Miralles, F. (2018). Ambient Intelligence by ATML Rules in BackHome. In Emerging Ideas on Information Filtering and Retrieval (pp. 31-54). Springer, Cham.

Cavoukian, A., Mihailidis, A., & Boger, J. (2010). Sensors and in-home collection of health data: A privacy by design approach. Information and Privacy Commissioner, Tech. Rep.

Chakraborty, S., Paul, D., Das, S., & Xu, J. (2020, June). Entropy weighted power k-means clustering. In International Conference on Artificial Intelligence and Statistics (pp. 691-701). PMLR.

Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, 2(01), 20-28.

Chavhan, P. G., Chavhan, G. H., Shafi, P. M., & Mahalle, P. N. (2021). Context Awareness Computing System—A Survey. In Proceedings of International Conference on Communication and Artificial Intelligence (pp. 423-429). Springer, Singapore.

Chelli, A., & Pätzold, M. (2019). A machine learning approach for fall detection and daily living activity recognition. IEEE Access, 7, 38670-38687.

Chua, S. L., Foo, L. K., & Guesgen, H. W. (2020). Predicting Activities of Daily Living with Spatio-Temporal Information. Future Internet, 12(12), 214.

Chung, H., & Shin, K. S. (2018). Genetic algorithm-optimized long short-term memory network for stock market prediction. Sustainability, 10(10), 3765.

Cinini, A., Cutugno, P., Ferraris, C., Ferretti, M., Marconi, L., Morgavi, G., & Nerino, R. (2021). Final results of the NINFA project: impact of new technologies in the daily life of elderly people. Aging Clinical and Experimental Research, 33(5), 1213-1222.

Clemente, J., Song, W., Valero, M., Li, F., & Liy, X. (2019, June). Indoor person identification and fall detection through non-intrusive floor seismic sensing. In 2019 IEEE International Conference on Smart Computing (SMARTCOMP) (pp. 417-424). IEEE.

Costa, S. D., Barcellos, M. P., de Almeida Falbo, R., Conte, T., & de Oliveira, K. M. (2022). A core ontology on the Human-Computer Interaction phenomenon. Data & Knowledge Engineering, 101977.

Cui, J., Ding, Z., Fan, P., & Al-Dhahir, N. (2018). Unsupervised machine learning-based user clustering in millimeter-wave-NOMA systems. IEEE Transactions on Wireless Communications, 17(11), 7425-7440.

Dang, L. M., Min, K., Wang, H., Piran, M. J., Lee, C. H., & Moon, H. (2020). Sensor-based and vision-based human activity recognition: A comprehensive survey. Pattern Recognition, 108, 107561.

Dawar, N., & Kehtarnavaz, N. (2018, June). A convolutional neural network-based sensor fusion system for monitoring transition movements in healthcare applications. In 2018 IEEE 14th International Conference on Control and Automation (ICCA) (pp. 482-485). IEEE.

de Quadros, T., Lazzaretti, A. E., & Schneider, F. K. (2018). A movement decomposition and machine learning-based fall detection system using wrist wearable device. IEEE Sensors Journal, 18(12), 5082-5089.

de Quadros, T., Lazzaretti, A. E., & Schneider, F. K. (2018). A movement decomposition and machine learning-based fall detection system using wrist wearable device. IEEE Sensors Journal, 18(12), 5082-5089.

Dhole, S. R., Kashyap, A., Dangwal, A. N., & Mohan, R. (2019). A novel helmet design and implementation for drowsiness and fall detection of workers on-site using EEG and Random-Forest Classifier. Procedia Computer Science, 151, 947-952.

Diao, E., Ding, J., & Tarokh, V. (2019, December). Restricted recurrent neural networks. In 2019 IEEE International Conference on Big Data (Big Data) (pp. 56-63). IEEE.

Dubois, A., Mouthon, A., Sivagnanaselvam, R. S., & Bresciani, J. P. (2019). Fast and automatic assessment of fall risk by coupling machine learning algorithms with a depth camera to monitor simple balance tasks. Journal of neuroengineering and rehabilitation, 16(1), 1-10.

Duke University: Duke Launches Autism Research App with Global Reach. (2022). Retrieved March 15, 2022, from https://autismandbeyond.researchkit.duke.edu/ch

Dunne, R., Morris, T., & Harper, S. (2021). A survey of ambient intelligence. ACM Computing Surveys (CSUR), 54(4), 1-27.

Dwivedi, Y. K., Rana, N. P., Tamilmani, K., & Raman, R. (2020). A meta-analysis based modified unified theory of acceptance and use of technology (meta-UTAUT): a review of emerging literature. Current opinion in psychology, 36, 13-18.

ElHady, N. E., & Provost, J. (2018). A systematic survey on sensor failure detection and fault-tolerance in ambient assisted living. Sensors, 18(7), 1991.

Elhayatmy, G., Dey, N., & Ashour, A. S. (2018). Internet of Things based wireless body area network in healthcare. In Internet of things and big data analytics toward next-generation intelligence (pp. 3-20). Springer, Cham.

Elshwemy, F. A., Elbasiony, R., & Saidahmed, M. T. (2020). A new approach for thermal vision based fall detection using residual autoencoder. Int. J. Intell. Eng. Syst, 13(2).

Espinosa, R., Ponce, H., Gutiérrez, S., Martínez-Villaseñor, L., Brieva, J., & Moya-Albor, E. (2019). A vision-based approach for fall detection using multiple cameras and convolutional neural networks: A case study using the UP-Fall detection dataset. Computers in biology and medicine, 115, 103520.

Fakhrulddin, A. H., Fei, X., & Li, H. (2017, November). Convolutional neural networks (CNN) based human fall detection on body sensor networks (BSN) sensor data. In 2017 4th international conference on systems and informatics (ICSAI) (pp. 1461-1465). IEEE.

Galán, S. F. (2019). Comparative evaluation of region query strategies for DBSCAN clustering. Information Sciences, 502, 76-90.

Ganesan, B., Gowda, T., Al-Jumaily, A., Fong, K. N. K., Meena, S. K., & Tong, R. K. Y. (2019). Ambient assisted living technologies for older adults with cognitive and physical impairments: a review. European review for medical and pharmacological sciences.

Gao, S., Calhoun, V. D., & Sui, J. (2018). Machine learning in major depression: From classification to treatment outcome prediction. CNS neuroscience & therapeutics, 24(11), 1037-1052.

Genoud, D., Cuendet, V., & Torrent, J. (2016, March). Soft fall detection using machine learning in wearable devices. In 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA) (pp. 501-505). IEEE.

Ghayvat, H., Mukhopadhyay, S., Shenjie, B., Chouhan, A., & Chen, W. (2018, May). Smart home based ambient assisted living: Recognition of anomaly in the activity of daily living for an elderly living alone. In 2018 IEEE international instrumentation and measurement technology conference (I2MTC) (pp. 1-5). IEEE.

Google Trends (2022). Fall Detection. Available online at: https://www.google.com/trends

Granić, A., & Marangunić, N. (2019). Technology acceptance model in educational context: A systematic literature review. British Journal of Educational Technology, 50(5), 2572-2593.

Guo, Y., Zhu, H., & Dang, X. (2020). Tracking multiple acoustic sources by adaptive fusion of TDOAs across microphone pairs. Digital Signal Processing, 106, 102853.

Gupta, T. K., & Raza, K. (2019). Optimization of ANN architecture: a review on nature-inspired techniques. Machine learning in bio-signal analysis and diagnostic imaging, 159-182.

Guvensan, M. A., Kansiz, A. O., Camgoz, N. C., Turkmen, H., Yavuz, A. G., & Karsligil, M. E. (2017). An energy-efficient multi-tier architecture for fall detection on smartphones. Sensors, 17(7), 1487.

Hakim, A., Huq, M. S., Shanta, S., & Ibrahim, B. S. K. K. (2017). Smartphone based data mining for fall detection: Analysis and design. Procedia computer science, 105, 46-51.

Hakim, A., Huq, M. S., Shanta, S., & Ibrahim, B. S. K. K. (2017). Smartphone based data mining for fall detection: Analysis and design. Procedia computer science, 105, 46-51.

Hamidzadeh, J., & Moslemnejad, S. (2019). Identification of uncertainty and decision boundary for SVM classification training using belief function. Applied Intelligence, 49(6), 2030-2045.

He, H., Zhang, X., Wang, Q., Ren, J., Liu, J., Zhao, X., & Cheng, Y. (2019). Ensemble multi boost based on ripper classifier for prediction of imbalanced software defect data. IEEE Access, 7, 110333-110343.

He, J., Bai, S., & Wang, X. (2017). An unobtrusive fall detection and alerting system based on Kalman filter and Bayes network classifier. Sensors, 17(6), 1393.

Hemmatpour, M., Ferrero, R., Montrucchio, B., & Rebaudengo, M. (2019). A review on fall prediction and prevention system for personal devices: evaluation and experimental results. Advances in human-computer interaction, 2019.

Henderson, J., Condell, J., Connolly, J., Kelly, D., & Curran, K. (2021). Review of wearable sensor-based health monitoring glove devices for rheumatoid arthritis. Sensors, 21(5), 1576.

Hossain, F., Ali, M. L., Islam, M. Z., & Mustafa, H. (2016, December). A direction-sensitive fall detection system using single 3D accelerometer and learning classifier. In 2016 International Conference on Medical Engineering, Health Informatics and Technology (MediTec) (pp. 1-6). IEEE.

Hossain, M. Z., Akhtar, M. N., Ahmad, R. B., & Rahman, M. (2019). A dynamic K-means clustering for data mining. Indonesian Journal of Electrical engineering and computer science, 13(2), 521-526.

Hsieh, C. Y., Liu, K. C., Huang, C. N., Chu, W. C., & Chan, C. T. (2017). Novel hierarchical fall detection algorithm using a multiphase fall model. Sensors, 17(2), 307.

Hussain, F., Hussain, F., Ehatisham-ul-Haq, M., & Azam, M. A. (2019). Activity-aware fall detection and recognition based on wearable sensors. IEEE Sensors Journal, 19(12), 4528-4536.

Hussain, F., Umair, M. B., Ehatisham-ul-Haq, M., Pires, I. M., Valente, T., Garcia, N. M., & Pombo, N. (2019). An efficient machine learning-based elderly fall detection algorithm. arXiv preprint arXiv:1911.11976.

Igual, R., Medrano, C., & Plaza, I. (2013). Challenges, issues and trends in fall detection systems. Biomedical engineering online, 12(1), 1-24.

Ionel, L. (2022). Characterizing spatio-temporal aspects of chirped pulse amplification laser micro-cone target interaction. Laser Physics, 32(3), 035301.

Jadhav, S., He, H., & Jenkins, K. (2018). Information gain directed genetic algorithm wrapper feature selection for credit rating. Applied Soft Computing, 69, 541-553.

Jahanjoo, A., Tahan, M. N., & Rashti, M. J. (2017, April). Accurate fall detection using 3-axis accelerometer sensor and MLF algorithm. In 2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA) (pp. 90-95). IEEE.

Jefiza, A., Pramunanto, E., Boedinoegroho, H., & Purnomo, M. H. (2017, September). Fall detection based on accelerometer and gyroscope using back propagation. In 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI) (pp. 1-6). IEEE.

Jiménez, M., Torres, M. T., John, R., & Triguero, I. (2020). Galaxy image classification based on citizen science data: a comparative study. IEEE Access, 8, 47232-47246.

Jobanputra, C., Bavishi, J., & Doshi, N. (2019). Human activity recognition: A survey. Procedia Computer Science, 155, 698-703.

Johns Hopkins University: EpiWatch: App and research study. (2022). Retrieved March 15, 2022, from https://www.epiwatch.com/

Kang, Y. K., Kang, H. Y., & Kim, J. B. (2021, January). A Study of Fall Detection System Using Context Cognition Method. In 2021 21st ACIS International Winter Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD-Winter) (pp. 79-83). IEEE.

Kao, H. C., Hung, J. C., & Huang, C. P. (2017, May). GA-SVM applied to the fall detection system. In 2017 International Conference on Applied System Innovation (ICASI) (pp. 436-439). IEEE.

Karasu, S., Altan, A., Bekiros, S., & Ahmad, W. (2020). A new forecasting model with wrapper-based feature selection approaches using multi-objective optimization technique for chaotic crude oil time-series—energy, 212, 118750.

Kayabol, K. (2019). Approximate sparse multinomial logistic regression for classification. IEEE transactions on pattern analysis and machine intelligence, 42(2), 490-493.

Kerdjidj, O., Ramzan, N., Ghanem, K., Amira, A., & Chouireb, F. (2020). Fall detection and human activity classification using wearable sensors and compressed sensing. Journal of Ambient Intelligence and Humanized Computing, 11(1), 349-361.

Khair, U., Fahmi, H., Al Hakim, S., & Rahim, R. (2017, December). Forecasting error calculation with mean absolute deviation and mean absolute percentage error. In Journal of Physics: Conference Series (Vol. 930, No. 1, p. 012002). IOP Publishing.

Khan, M. A., Saboor, A., Kim, H. C., & Park, H. (2021). A Systematic Review of Location Aware Schemes in the Internet of Things. Sensors, 21(9), 3228.

Khan, S., Qamar, R., Zaheen, R., Al-Ali, A. R., Al Nabulsi, A., & Al-Nashash, H. (2019). Internet of things based multi-sensor patient fall detection system. Healthcare Technology Letters, 6(5), 132-137.

Khan, S., Rahmani, H., Shah, S. A. A., & Bennamoun, M. (2018). A guide to convolutional neural networks for computer vision. Synthesis Lectures on Computer Vision, 8(1), 1-207.

Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2020). Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one-class support vector machine. Electronics, 9(1), 173.

Kim, D., Han, H., Wang, W., & Kim, H. S. (2022). Improvement of Deep Learning Models for River Water Level Prediction Using Complex Network Method. Water, 14(3), 466.

Kim, T. H., Choi, A., Heo, H. M., Kim, K., Lee, K., & Mun, J. H. (2019). Machine learning-based pre-impact fall detection model to discriminate various types of fall. Journal of biomechanical engineering, 141(8).

Kozan, P., Phan, J., Thaker, A., Sedarat, A., & Issa, D. (2021). S3419 Return to Activities of Daily Living Following Outpatient Endoscopic Bariatric Therapy. Official journal of the American College of Gastroenterology| ACG, 116, S1405.

Kumar Basak, S., Wotto, M., & Belanger, P. (2018). E-learning, M-learning and D-learning: Conceptual definition and comparative analysis. E-learning and Digital Media, 15(4), 191-216.

Kumar, A., Pirogova, E., & Fang, J. Q. (2018, December). Classification of error-related potentials using linear discriminant analysis. In 2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES) (pp. 18-21). IEEE.

Kumar, V. C., Ha, S., Sawicki, G., & Liu, C. K. (2020, May). Learning a control policy for fall prevention on an assistive walking device. In 2020 IEEE International Conference on Robotics and Automation (ICRA) (pp. 4833-4840). IEEE.

Lai, H. Y., Chen, X. X., Chen, W., Tang, H., & Lin, H. (2017). Sequence-based predictive modelling to identify cancerlectins. Oncotarget, 8(17), 28169.

Lang, S., Bravo-Marquez, F., Beckham, C., Hall, M., & Frank, E. (2019). Wekadeeplearning4j: A deep learning package for weka based on deeplearning4j. Knowledge-Based Systems, 178, 48-50.

Lee, J. S., & Tseng, H. H. (2019). Development of an enhanced threshold-based fall detection system using smartphones with built-in accelerometers. IEEE Sensors Journal, 19(18), 8293-8302.

Li, H., Shrestha, A., Fioranelli, F., Le Kernec, J., Heidari, H., Pepa, M., ... & Spinsante, S. (2017, October). Multisensor data fusion for human activities classification and fall detection. In 2017 IEEE sensors (pp. 1-3). IEEE.

Li, Y., Zhang, L., Xu, Y., Yao, Y., Lau, R. Y. K., & Wu, Y. (2017). Enhancing binary classification by modeling uncertain boundary in three-way decisions. IEEE transactions on knowledge and data engineering, 29(7), 1438-1451.

Li, Z., Li, H., Jiang, X., Chen, B., Zhang, Y., & Du, G. (2018, November). Efficient fpga implementation of softmax function for DNN applications. In 2018 12th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID) (pp. 212-216). IEEE.

Liciotti, D., Bernardini, M., Romeo, L., & Frontoni, E. (2020). A sequential deep learning application for recognising human activities in smart homes. Neurocomputing, 396, 501-513.

Liu, K. C., Hsieh, C. Y., Hsu, S. J. P., & Chan, C. T. (2018). Impact of sampling rate on wearable-based fall detection systems based on machine learning models. IEEE Sensors Journal, 18(23), 9882-9890.

Lou, Z., Wang, L., Jiang, K., Wei, Z., & Shen, G. (2020). Reviews of wearable healthcare systems: Materials, devices and system integration. Materials Science and Engineering: R: Reports, 140, 100523.

Ma, J., & Gao, X. (2020). A filter-based feature construction and feature selection approach for classification using Genetic Programming: knowledge-Based Systems, 196, 105806.

Ma, Z., Liu, Y., Liu, X., Ma, J., & Ren, K. (2019). Lightweight privacy-preserving ensemble classification for face recognition. IEEE Internet of Things Journal, 6(3), 5778-5790.

Maetzler, W., & Pilotto, A. (2021). Digital assessment at home—mPower against Parkinson disease. Nature Reviews Neurology, 17(11), 661-662.

Manekar, R., Saurav, S., Maiti, S., Singh, S., Chaudhury, S., Kumar, R., & Chaudhary, K. (2020). Activity recognition for indoor fall detection in 360-degree videos using deep learning techniques. In Proceedings of 3rd International Conference on Computer Vision and Image Processing (pp. 417-429). Springer, Singapore.

Mauldin, T. R., Canby, M. E., Metsis, V., Ngu, A. H., & Rivera, C. C. (2018). SmartFall: A smartwatch-based fall detection system using deep learning. Sensors, 18(10), 3363.

Miall, R. C., Rosenthal, O., Ørstavik, K., Cole, J. D., & Sarlegna, F. R. (2019). Loss of haptic feedback impairs control of hand posture: a study in chronically deafferented individuals when grasping and lifting objects. Experimental brain research, 237(9), 2167-2184.

Miranda, G. M. D., Mendes, A. D. C. G., & Silva, A. L. A. D. (2016). Population aging in Brazil: current and future social challenges and consequences. Revista Brasileira de Geriatria e Gerontologia, 19, 507-519.

Mishra, P., Biancolillo, A., Roger, J. M., Marini, F., & Rutledge, D. N. (2020). New data preprocessing trends are based on an ensemble of multiple preprocessing techniques—TrAC Trends in Analytical Chemistry, 116045.

Mohammadi, S., Mirvaziri, H., Ghazizadeh-Ahsaee, M., & Karimipour, H. (2019). Cyber intrusion detection by the combined feature selection algorithm. Journal of information security and applications, 44, 80-88.

Moreira, M. W., Rodrigues, J. J., Korotaev, V., Al-Muhtadi, J., & Kumar, N. (2019). A comprehensive review on smart decision support systems for health care. IEEE Systems Journal, 13(3), 3536-3545.

Morrissey, E. C., Casey, M., Glynn, L. G., Walsh, J. C., & Molloy, G. J. (2018). Smartphone apps for improving medication adherence in hypertension: patients' perspectives. Patient preference and adherence, 12, 813.

Munkhdalai, L., Munkhdalai, T., Park, K. H., Lee, H. G., Li, M., & Ryu, K. H. (2019). A mixture of activation functions with extended min-max normalization for forex market prediction. IEEE Access, 7, 183680-183691.

Musci, M., De Martini, D., Blago, N., Facchinetti, T., & Piastra, M. (2020). Online fall detection using recurrent neural networks on smart wearable devices. IEEE Transactions on Emerging Topics in Computing, 9(3), 1276-1289.

Naja, S., Makhlouf, M. M. E. D., & Chehab, M. A. H. (2017). An ageing world of the 21st century: a literature review. Int J Community Med Public Health, 4(12), 4363-9.

Nakamura, T., Bouazizi, M., Yamamoto, K., & Ohtsuki, T. (2020, December). Wi-fi-CSI-based fall detection by spectrogram analysis with CNN. In GLOBECOM 2020-2020 IEEE Global Communications Conference (pp. 1-6). IEEE.

Nejatian, S., Parvin, H., & Faraji, E. (2018). Using sub-sampling and ensemble clustering techniques to improve the performance of imbalanced classification. Neurocomputing, 276, 55-66.

Ngiam, K. Y., & Khor, W. (2019). Big data and machine learning algorithms for healthcare delivery. The Lancet Oncology, 20(5), e262-e273.

Ngu, A. H., Tseng, P. T., Paliwal, M., Carpenter, C., & Stipe, W. (2018). Smartwatch-based IoT fall detection application. Open Journal of Internet Of Things (OJIOT), 4(1), 87-98.

Nguyen, T. L., Le, T. A., & Pham, C. (2018, November). The internet-of-things based fall detection using fusion feature. In 2018 10th International Conference on Knowledge and Systems Engineering (KSE) (pp. 129-134). IEEE.

Niknejad, N., Ismail, W. B., Mardani, A., Liao, H., & Ghani, I. (2020). A comprehensive overview of smart wearables: The state of the art literature, recent advances, and future challenges. Engineering Applications of Artificial Intelligence, 90, 103529.

Ning, Y., Zhang, S., Nie, X., Li, G., & Zhao, G. (2019, September). Fall detection algorithm based on gradient boosting decision tree. In 2019 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC) (pp. 1-4). IEEE.

Nogas, J., Khan, S. S., & Mihailidis, A. (2018, July). Fall detection from thermal camera using convolutional lstm autoencoder. In Proceedings of the 2nd workshop on Aging, Rehabilitation and Independent Assisted Living, IJCAI Workshop.

Norton, J. M., Reddy, P. K., Subedi, K., Fabrizio, C. A., Wimmer, N. J., & Urrutia, L. E. (2021). Utilization of an ICU severity of illness scoring system to triage patients with ST-elevation myocardial infarction. Journal of Intensive Care Medicine, 36(8), 857-861.

Nowlan, S. J., & Hinton, G. E. (2018). Simplifying neural networks by soft weight sharing. In The Mathematics of Generalization (pp. 373-394). CRC Press.

Offermann-van Heek, J., & Ziefle, M. (2018). They don’t care about us! Care personnel’s perspectives on ambient assisted living technology usage: scenario-based survey study. JMIR rehabilitation and assistive technologies, 5(2), e10424.

Oguego, C. L., Augusto, J. C., Muñoz, A., & Springett, M. (2018). A survey on managing users' preferences in ambient intelligence. Universal Access in the Information Society, 17(1), 97-114.

Panigrahi, R., & Borah, S. (2020). A Statistical Analysis of Lazy Classifiers Using Canadian Institute of Cybersecurity Datasets. In Advances in Data Science and Management (pp. 215-222). Springer, Singapore.

Pires, I. M., Garcia, N. M., Pombo, N., Flórez-Revuelta, F., Spinsante, S., & Teixeira, M. C. (2018). Identification of activities of daily living through data fusion on motion and magnetic sensors embedded on mobile devices. Pervasive and Mobile Computing, 47, 78-93.

Purba, E. (2020). Classification of Feasibility of Credit for Candidate CS Finance Debtors Using Naïve Bayes Method. Jurnal Mantik, 4(3), 1885-1899.

Putra, I. P. E. S., Brusey, J., Gaura, E., & Vesilo, R. (2017). An event-triggered machine learning approach for accelerometer-based fall detection. Sensors, 18(1), 20.

Putra, I. P. E. S., Brusey, J., Gaura, E., & Vesilo, R. (2018). An event-triggered machine learning approach for accelerometer-based fall detection. Sensors, 18(1), 20.

Putra, I. P. E. S., Brusey, J., Gaura, E., & Vesilo, R. (2018). An event-triggered machine learning approach for accelerometer-based fall detection. Sensors, 18(1), 20.

Ramezani, R., Xiao, Y., & Naeim, A. (2018, March). Sensing-Fi: Wi-Fi CSI and accelerometer fusion system for fall detection. In 2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI) (pp. 402-405). IEEE.

Ramirez, H., Velastin, S. A., Fabregas, E., Meza, I., Makris, D., & Farias, G. (2021). Fall detection using human skeleton features.

Ranieri, C. M., MacLeod, S., Dragone, M., Vargas, P. A., & Romero, R. A. F. (2021). Activity recognition for ambient assisted living with videos, inertial units and ambient sensors. Sensors, 21(3), 768.

Ravi, S., Climent-Pérez, P., & Florez-Revuelta, F. (2021). A Review on Visual Privacy Preservation Techniques for Active and Assisted Living. arXiv preprint arXiv:2112.09422.

Remeseiro, B., & Bolon-Canedo, V. (2019). A review of feature selection methods in medical applications. Computers in biology and medicine, 112, 103375.

Researchkit. (2022). Retrieved March 15, 2022, from http://researchkit.org/

Rizk, H., Yamaguchi, H., Youssef, M., & Higashino, T. (2020, November). Gain without pain: Enabling fingerprinting-based indoor localization using tracking scanners. In Proceedings of the 28th International Conference on Advances in Geographic Information Systems (pp. 550-559).

Rodrigues, T. B., Salgado, D. P., Cordeiro, M. C., Osterwald, K. M., Teodiano Filho, F. B., de Lucena Jr, V. F., ... & Murray, N. (2018). Fall detection system by machine learning framework for public health. Procedia Computer Science, 141, 358-365.

Rosato, A., Altilio, R., & Panella, M. (2021). A decentralized algorithm for distributed ensemble clustering. Information Sciences, 578, 417-434.

Saha, J., Chowdhury, C., & Biswas, S. (2020). Review of machine learning and deep learning based recommender systems for health informatics. In Deep Learning Techniques for Biomedical and Health Informatics (pp. 101-126). Springer, Cham.

Salazar, J., Kirchhoff, K., & Huang, Z. (2019, May). Self-attention networks for connectionist temporal classification in speech recognition. In ICASSP 2019-2019 IEEE International

Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7115-7119). IEEE.

Saleh, M., & Jeannès, R. L. B. (2019). Elderly fall detection using wearable sensors: A low cost highly accurate algorithm. IEEE Sensors Journal, 19(8), 3156-3164.

Salleh, S. M., & Koh, C. Z. (2020). Neural Network Algorithm-based Fall Detection Modelling. International Journal of Integrated Engineering, 12(3), 138-150.

Salmi, N., & Rustam, Z. (2019, June). Naïve Bayes classifier models for predicting colon cancer. IOP Conference Series: Materials Science and Engineering (Vol. 546, No. 5, p. 052068). IOP Publishing.

Sammaknejad, N., Zhao, Y., & Huang, B. (2019). A review of the expectation-maximization algorithm in data-driven process identification. Journal of process control, 73, 123-136.

Sandeepa, C., Moremada, C., Dissanayaka, N., Gamage, T., & Liyanage, M. (2020, June). An emergency situation detection system for ambient assisted living. In 2020 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 1-6). IEEE.

Santos, G. L., Endo, P. T., Monteiro, K. H. D. C., Rocha, E. D. S., Silva, I., & Lynn, T. (2019). Accelerometer-based human fall detection using convolutional neural networks. Sensors, 19(7), 1644.

Sarabia-Jácome, D., Usach, R., Palau, C. E., & Esteve, M. (2020). Highly-efficient fog-based deep learning AAL fall detection system. Internet of Things, 11, 100185.

Sarabia-Jácome, D., Usach, R., Palau, C. E., & Esteve, M. (2020). Highly-efficient fog-based deep learning AAL fall detection system. Internet of Things, 11, 100185.

Shah, S. M. S., Shah, F. A., Hussain, S. A., & Batool, S. (2020). Support vector machines-based heart disease diagnosis using feature subset, wrapping selection and extraction methods. Computers & Electrical Engineering, 84, 106628.

Shahzad, A., & Kim, K. (2018). FallDroid: An automated smart-phone-based fall detection system using multiple kernel learning. IEEE Transactions on Industrial Informatics, 15(1), 35-44.

Shinde, P. P., & Shah, S. (2018, August). A review of machine learning and deep learning applications. In 2018 Fourth international conference on computing communication control and automation (ICCUBEA) (pp. 1-6). IEEE.

Şimşek, A. S., & Topaloglu, H. (2018). An expectation-maximization algorithm to estimate the parameters of the Markov chain choice model. Operations Research, 66(3), 748-760.

Sinaga, K. P., & Yang, M. S. (2020). Unsupervised K-means clustering algorithm. IEEE Access, 8, 80716-80727.

Singh, A. P., Luhach, A. K., Gao, X. Z., Kumar, S., & Roy, D. S. (2020). Evolution of wireless sensor network design from technology centric to user centric: an architectural perspective. International Journal of Distributed Sensor Networks, 16(8), 1550147720949138.

Sinha, M., Sachan, D. K., & Parthasarathi, R. (2021). Artificial Intelligence in Clinical Toxicology. In Artificial Intelligence in Medicine (pp. 1-15). Cham: Springer International Publishing.

Siva Shankar, G., Ashokkumar, P., Vinayakumar, R., Ghosh, U., Mansoor, W., & Alnumay, W. S. (2020). An Embedded-Based Weighted Feature Selection Algorithm for Classifying Web Document. Wireless Communications and Mobile Computing, 2020.

Soleimani, E., & Nazerfard, E. (2021). Cross-subject transfer learning in human activity recognition systems using generative adversarial networks. Neurocomputing, 426, 26-34.

Stevens, J. A., & Phelan, E. A. (2013). Development of STEADI: a fall prevention resource for health care providers. Health promotion practice, 14(5), 706-714.

Swathy, M., & Saruladha, K. (2021). A comparative study of classification and prediction of Cardio-Vascular Diseases (CVD) using Machine Learning and Deep Learning techniques. ICT Express.

Tahir, A., Ahmad, J., Morison, G., Larijani, H., Gibson, R. M., & Skelton, D. A. (2021). Hrnn4f: Hybrid deep random neural network for multi-channel fall activity detection. Probability in the Engineering and Informational Sciences, 35(1), 37-50.

Torti, E., Fontanella, A., Musci, M., Blago, N., Pau, D., Leporati, F., & Piastra, M. (2018, August). Embedded real-time fall detection with deep learning on wearable devices. In 2018 21st euromicro conference on digital system design (DSD) (pp. 405-412). IEEE.

Tsinganos, P., & Skodras, A. (2017, September). A smartphone-based fall detection system for the elderly. In Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis (pp. 53-58). IEEE.

Vallabh, P., Malekian, R., Ye, N., & Bogatinoska, D. C. (2016, September). Fall detection using machine learning algorithms. In 2016 24th international conference on software, telecommunications and computer networks (SoftCOM) (pp. 1-9). IEEE.

Vidyarthi, V. K., & Jain, A. (2020). Knowledge extraction from trained ANN drought classification model. Journal of Hydrology, 585, 124804.

Viswakumar, A., Rajagopalan, V., Ray, T., & Parimi, C. (2019, November). Human gait analysis using OpenPose. In 2019 fifth international conference on image information processing (ICIIP) (pp. 310-314). IEEE.

Wang, H., Li, M., Li, J., Cao, J., & Wang, Z. (2016, August). An improved fall detection approach for elderly people based on feature weight and Bayesian classification. In 2016 IEEE International Conference on Mechatronics and Automation (pp. 471-476). IEEE.

Wang, Y., Cang, S., & Yu, H. (2019). A survey on wearable sensor modality centred human activity recognition in health care. Expert Systems with Applications, 137, 167-190.

Ward, R. J., Jjunju, F. P. M., Griffith, E. J., Wuerger, S. M., & Marshall, A. (2020). Artificial Odour-Vision Syneasthesia via Olfactory Sensory Argumentation. IEEE Sensors Journal, 21(5), 6784-6792.

Waris, A., & Aggarwal, R. K. (2018, March). Acoustic modeling in automatic speech recognition-A survey. In 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 1408-1412). IEEE.

Weitschies, W., Müller, L., Grimm, M., & Koziolek, M. (2021). Ingestible devices for studying the gastrointestinal physiology and their application in oral biopharmaceutics. Advanced Drug Delivery Reviews, 176, 113853.

WHO (2018). Falls. Available online at: https://www.who.int/news-room/fact-sheets/detail/falls

Wibowo, A. H., & Oesman, T. I. (2020, February). The comparative analysis on the accuracy of k-NN, naive Bayes, and decision tree algorithms in predicting crimes and criminal actions in Sleman regency. In Journal of Physics: Conference Series (Vol. 1450, No. 1, p. 012076). IOP Publishing.

World Health Organization. (2022). Falls. World Health Organization. Retrieved March 14, 2022, from https://www.who.int/news-room/fact-sheets/detail/falls

Xiao, J., & Bulut, O. (2020). Evaluating the performances of missing data handling methods in ability estimation from sparse data. Educational and Psychological Measurement, 80(5), 932-954.

Xu, J., He, Z., & Zhang, Y. (2019, July). CNN-LSTM combined network for IoT enabled fall detection applications. In Journal of Physics: Conference Series (Vol. 1267, No. 1, p. 012044). IOP Publishing.

Xu, T., Zhou, Y., & Zhu, J. (2018). New advances and challenges of fall detection systems: A survey. Applied Sciences, 8(3), 418.

Yadav, S. S., & Jadhav, S. M. (2019). Deep convolutional neural network-based medical image classification for disease diagnosis. Journal of Big Data, 6(1), 1-18.

Yang, F. J. (2018, December). An implementation of naive bayes classifier. In 2018 International conference on computational science and computational intelligence (CSCI) (pp. 301-306). IEEE.

Yang, W., Si, Y., Wang, D., & Guo, B. (2018). Automatic recognition of arrhythmia based on principal component analysis network and linear support vector machine. Computers in biology and medicine, 101, 22-32.

Yu, S., Chen, H., & Brown, R. A. (2017). Hidden Markov model-based fall detection with motion sensor orientation calibration: A case for real-life home monitoring. IEEE journal of biomedical and health informatics, 22(6), 1847-1853.

Yuan, C., & Yang, H. (2019). Research on the K-value selection method of the K-means clustering algorithm. J, 2(2), 226-235.

Zhang, Y., Wei, S., Zhang, L., & Liu, C. (2019). Comparing the performance of random forest, SVM and their variants for ECG quality assessment combined with nonlinear features. Journal of medical and biological engineering, 39(3), 381-392.

Zhao, S., Li, W., Niu, W., Gravina, R., & Fortino, G. (2018, March). Recognition of human fall events based on single tri-axial gyroscope. In 2018 IEEE 15th International conference on networking, sensing and control (ICNSC) (pp. 1-6). IEEE.

Zhou, X., Qian, L. C., You, P. J., Ding, Z. G., & Han, Y. Q. (2018, July). Fall detection using convolutional neural network with multi-sensor fusion. In 2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) (pp. 1-5). IEEE.

Appendix A

Publications

Paper entitled "**Novel Stacking Classification and Prediction Algorithm Based Ambient Assisted Living for Elderly**" was accepted for publication in Wireless Communications and Mobile Computing-Hindawi Journal (Impact factor:1.8 , Indexed in SCE, Q2)