IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 January 2024, accepted 24 January 2024, date of publication 1 February 2024, date of current version 7 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3360861

RESEARCH ARTICLE

Comparison of Feature Selection and Supervised Methods for Classifying Gait Disorders

MOHSEN SHAYESTEGAN[1], **JAN KOHOUT**[2], **LUDMILA VEREŠPEJOVÁ**[3], **MARTIN CHOVANEC**[3], **AND JAN MAREŠ**[1,2]

[1] Faculty of Electrical Engineering and Informatics, University of Pardubice, 530 02 Pardubice, Czech Republic
[2] Department of Mathematics, Informatics and Cybernetics, University of Chemistry and Technology Prague, 166 28 Prague, Czech Republic
[3] Department of Otorhinolaryngology, University Hospital Kralovske Vinohrady, Third Faculty of Medicine, Charles University, 100 34 Prague, Czech Republic

Corresponding author: Jan Mareš (jan.mares@vscht.cz)

The work of Jan Kohout and Jan Mareš was supported by the Ministry of Education, Youth, and Sports through the Development of Advanced Computational Algorithms for Evaluating Post-Surgery Rehabilitation under Grant LTAIN19007. The work of Martin Chovanec and Katerina Trnkova was supported by a Research Project of Charles University Cooperation—Surgical Disciplines, Third Faculty of Medicine.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Ethics Committee of the University Hospital Královské Vinohrady Prague under Application No. EK-VP/4310120.

ABSTRACT Recently, systems for classifying gait disorders have been of great interest. However, quantifying the progress of these disorders has been highly dependent on a physician's judgement in classifying sick and healthy subjects. We examine the effects of gait stability analysis on gait dysfunction problems, which are impacted by the patient's dynamic balance. The dataset in this study was collected and labelled based on the opinions of physicians at Prague Hospital; it included 84 measurements of 37 patients. A keypoint detector was applied to detect the skeletal keypoints of patients. We have prepared two different datasets from the detection and tracking results. For the proposed feature selection method, we have used statistical measurements such as the x and y coordinates for each keypoint, the distance, and the angle between two selected keypoints. Using these statistical measurements, we have prepared different subgroups with different numbers of features to examine. We have also applied ten different feature selection algorithms to obtain data from different numbers of features automatically. Then, these datasets with high-level features were used to train well-known networks, such as the long short-term memory (LSTM), gated recurrent unit (GRU), and multiple layer perceptron (MLP) networks. The study results showed that the 30 features selected by the analysis of variance (ANOVA) algorithm and used to train the GRU network ranked among the best features and resulted in a classification F-score of 85%. The results also prove that the data generated by the detector method are more effective than the data generated by the tracking method due to the format of the exercises in our dataset, which were designed by physicians. Moreover, the best feature selection approaches have considerably improved the classification F-score compared to manual feature generation.

INDEX TERMS Classification, deep learning, feature selection, gait analysis, GRU, LSTM, MLP, pattern recognition.

ABBREVIATIONS

Adam	Adaptive Moment Estimation.
ANOVA	Analysis of Variance.
CMI	Conditional Mutual Information.
CMIM	Conditional Mutual Information Maximisation.
CNNs	Convolutional Neural Networks.
DISR	Double Input Symmetrical Relevance.
FJMI	Fuzzy Joint Mutual Information.
GPU	Graphics Processing Unit.
GRFs	Ground Reaction Forces.
GRU	Gated Recurrent Unit.

The associate editor coordinating the review of this manuscript and approving it for publication was Utku Kose.

LSTM	Long Short-Term Memory.
MCRMCR	Minimum Conditional Relevance-Minimum Conditional Redundancy.
MLP	Multilayer Perceptron.
MI	Mutual Information.
MIFS	Mutual Information-based Feature Selection.
MS-COCO	Microsoft Common Objects in Context.
mRMR	minimal-redundancy-maximal-relevance.
RBEFF	Random Multi-subspace Based ReliefF.
RCNN	Region-based Convolutional Neural Networks.
ReLU	Rectified Linear Unit.
ResNet	Residual Network.
RNN	Recurrent Neural Network.
SRCFS	multi-Subspace Randomization and Collaboration Feature Selection.
SURF	Speeded Up Robust Features.
SVM	Support Vector Machine.

I. INTRODUCTION

Visually based human motion analysis is a general method of analysing and understanding people's movements, as captured by a camera. It includes pattern recognition, biomechanics, machine vision, and artificial intelligence. It is a challenging field with significant applications for businesses, education, and society [1], [2]. Human movements are conceptually classified by complexity into actions, gestures, interactions, and activities. The method of recognition involves tracking the human body in a video. The general methods of recognition include 2D kinematics, 3D kinematics, and image models. Human motion recognition with kinematic methods corresponds to human characteristics such as the number of joints and the lengths of the limbs [1]. The recognition of human actions in a video has recently become an interesting research area, and pose estimation, including the detection of body parts, has received research attention. Most studies focus on the motion analysis of the human skeleton [3]. The analysis of human motion with vision sensors is a new area of research, and much effort has been expended on analysing human motion using a Kinect camera [4]. Even after the Kinect camera was introduced, many applications and studies still required specialists to analyse the results. In this scenario, intelligent systems can play an essential role by quickly and objectively analysing human motion [4]. Methods for selecting features are applied in many intelligent systems and applications, such as machine learning, natural language processing, and bioinformatics. The selection of features is usually performed during data pre-processing, before the classifier is trained [5]. Feature selection not only identifies the essential features and improves the quality of the dataset but also eliminates undesirable features that impair the dataset's quality. The feature selection process selects the most significant features for advanced processing [6]. In most feature selection methods, a statistical measurement (the correlation) is used to determine the relationships between the characteristics. In other words, if one feature's systemic change affects another feature, then these two features are highly related. Feature selection methods, such as mutual information (MI), conditional mutual information (CMI), conditional mutual information maximisation (CMIM), and double input symmetrical relevance (DISR), can be used to predict the relationship between linear and nonlinear features [6].

In [7], the analysis of variance (ANOVA) is used to remove irrelevant and unwanted characteristics from data before using the feature selection method. The study used a hybrid model that combines ANOVA and whale optimisation to improve the results of classifying different heart disease datasets. First, ANOVA is used to select the relevant feature sets, and during this period, the whale optimisation discovers the best collection of features from the previous collection of features [7]. Moreover, the authors of [8] used the ANOVA feature selection method to reduce the number of features to provide a better separation between COVID-19 features. In [9], the authors propose an improved methodology for classifying Arabic text using the selection of chi-square features to improve the classification performance. They presented improved methods for selecting the chi-square feature method to minimise data and produce a greater classification accuracy. The study claimed that the chi-square method was very effective but still had some limitations, such as the number of attributes that affect the classification accuracy [9]. The authors of [10] proposed combining the chi-square method with the long short-term memory (LSTM), Bi-LSTM, and gated recurrent unit (GRU) models for sentiment analysis and compared the metrics of two benchmark datasets, YELP and American Airlines. They stated that the combination of LSTM networks and the feature selection method improved the accuracy of sentiment analysis. In [11], chi-square feature selection and an ensemble of classifiers, such as LPBoost, modified naive Bayes (MNB), and the support vector machine (SVM), are applied to improve an intrusion detection model. In experimental evaluations, this method exhibited a high accuracy in comparison with base classifiers [11]. The study presented in [12] designed a simple and very efficient feature selection method based on CMI. The authors showed that this feature selection approach outperforms other conventional techniques [12]. The authors of [13] also designed and enhanced the MI method. They introduced the normalised MI by eliminating a user-defined parameter. The authors of [14] compared efficient feature selection methods such as ReliefF, minimal-redundancy-maximal-relevance (mRMR), Mutual Information-based Feature Selection (MIFS), minimum conditional relevance-minimum conditional redundancy (MCRMCR), and CMIM on 15 public biological datasets and two artificial datasets [14]. The study presented in [5] introduces a nonlinear feature selection method that uses MI and the maximum of the minimum principle to mitigate the problem of overrating the feature significance. The Double Input Symmetrical Relevance (DISR) was introduced by [15]; it is promising in high feature-to-sample ratio classification tasks, such as those that involve gene expression microarray datasets. The study presented in [16] applied the efficient Kruskal-Wallis technique to select the most prominent face features. The algorithm eliminated redundant features and selected the most discriminative face features to recognise face images. In [17], a distributed version of the ReliefF algorithm was presented using the emerging Apache Spark programming model. In [18], a novel method called Random Multi-subspace Based ReliefF (RBEFF) is proposed and compared with popular methods such as the ReliefF, MI, Fuzzy Joint Mutual Information (FJMI), and multi-Subspace Randomization and Collaboration Feature Selection (SRCFS) algorithms on 28 real datasets. In [19], MultiSurf algorithms were used to carry out extensive experiments to obtain precise features of structural information and made it possible to better interpret the interaction between two- and three-way genes using the average nearest neighbours. The authors of [20] implemented a Relief-based algorithm training environment. They implemented the ReliefF, Speeded Up Robust Features (SURF), SURF*, MultiSURF*, and MultiSURF techniques. The above-mentioned algorithms are simple and very efficient feature selection methods, and they have shown that this type of feature selection approach outperforms conventional techniques. In this paper, we have used some of these methods to perform feature selection among all the gait features created to investigate the performance of the network on different feature size inputs.

A. BIOMEDICAL BACKGROUND

Human balance relies on a multisensory system that includes proprioception, the vestibular system, and vision to maintain body and limb positions in space. Dysfunction in these systems can lead to specific postural and gait issues.

Postural stability is vital for both static and dynamic coordination during movement. Gait involves a sequence of involuntary movements, which are analysed using spatiotemporal, kinematic, and kinetic features. Parameters like the walking speed, step frequency, and stride length offer insights into gait patterns [21].

A systematic gait analysis approach uncovers subtle variations that are often missed when the focus is on major features. Gait tracking diagrams record changes during specific phases – the standing, swing, and limb phases. These phases entail fulfilling unique demands for walking. The step width measures the horizontal distance between foot positions in an event, while the foot progression angle quantifies the angle between the foot's longitudinal axis and the direction of progression.

Kinetic analysis studies motion-affecting forces, including ground reaction forces on the hip, knee, and ankle joints. Kinematic analysis describes movements independently of forces, covering various planes and joints.

Clinical practice employs the rapid gait test, matching movement patterns to stereotypes, although this requires expertise. A pathological gait arises from orthopedic and neurological diseases. Neurology, orthopedics, and neurootology evaluations address postural and gait issues, complementing equilibrium assessments. Questionnaires gauge the impact of balance and gait on daily life and the quality of life. Clinical tests, like the Timed Up and Go Test, Berg Balance Scale, or Six-Minute Walk Test, assess dynamic stability but have limitations [22]. They offer quick assessments but lack quantitative processing and may be subjective. Currently, there is a gap in objectively evaluating the overall dynamic stability in clinical examinations.

B. GAIT DISORDER ANALYSIS

Gait analysis is an effective medical tool that is used for many applications involving the evaluation of progress during rehabilitation, neurological disorders, and the risk of falling. An effective gait evaluation tool that automatically tracks patients' skeletal keypoints can provide significant information to medical professionals to evaluate rehabilitation and carry out preventive analyses. Gait detection is a crucial research problem because by identifying abnormal and imbalanced gaits, weaknesses in given functions can be discovered in the human body [23]. In abnormal gait recognition based on skeletons, the use of original skeleton data reduces the performance of the recognition because it contains irrelevant information and noise, so features that are extracted and selected from the skeleton data are usually used [23]. A gait disorder is one indicator of neurological disorders. Studies have shown that machine learning methods can be applied to the classification of neurological disorders based on gait data [24].

In several studies, some movement parameters are used to distinguish between diseases. Unfortunately, the high variability of motion caused by patients with a body function deficiency increases the complexity of the classification. Methods of feature selection that have a major impact on enhancing the classification result have recently been investigated [25]. Indeed, a particular subset of all extracted features may be more suitable for determining a particular disease. For instance, spatiotemporal variables, such as the walking speed and step length, are the major indicators of the severity of the deficit and functional ability. From this point of view, combining the spatial and temporal data of movement into a limited set of variables that distinguish healthy and disabled subjects may be effective at informing medical professionals outside the movement analysis laboratory [25].

The authors of [24] investigated one construction technique and three feature selection methods to analyse neurodegenerative gaits. To evaluate the classification power of these feature combinations, they developed an SVM-based classifier.

The authors of [23] mentioned that it is hard to extract meaningful patterns using the existing feature extraction methods. They developed two recurrent neural network (RNN)-based autoencoder methods to solve this problem.

Their results indicate that the features extracted by their methods were more easily classified than the original skeleton-based features, and they yielded more accurate classifications.

The authors of [26] used the Kinect system to collect 3D skeleton data for one normal gait and five pathological gaits. They developed a GRU-based classifier to classify the pathological gaits. They achieved promising results with the GRU; the classification accuracy was around 90%. They recommended that their method be used to support experimental and examination decisions.

The authors of [27] used Kinect v2 to analyse three-dimensional motion. They extracted the kinematics and spatiotemporal parameters of ten healthy people while they walked on a treadmill. Their results showed that the Kinect sensor can be an effective assessment tool during the gait cycle.

The authors of [28] used ground reaction forces (GRFs) and the Gutenberg and GaitRec databases to assist physicians in classifying gait patterns. In all, these datasets contain data from 2645 gait disorder patients and around 560 healthy control patients. They applied three feature selection algorithms for feature extraction and for removing highly correlated features. They found that time-domain and wavelet features are the most significant features for gait classification.

With the development of Kinect cameras and skeleton detection algorithms, many skeleton-based methods for classifying gait disorders have recently been proposed. These methods performed well on simple gait patterns but were unable to classify complicated gait patterns. In other words, there is still a lack of a detailed analysis of the feature extraction and selection problem and a lack of accurate gait disorder recognition. In this study, we classify one normal gait and two disorders by developing three well-known deep neural networks: the LSTM, GRU, and multiple layer perceptron (MLP). These three classifiers are developed to classify gait disorders, and we compare their performances. We collected skeleton data as a base feature dataset by using a Kinect system. Furthermore, we extract meaningful features from skeleton-based features and then consider various joint groups from the total number of extracted features; we also apply ten feature selection algorithms to collect different joint groups to identify the essential features for gait disorder classification. The main objective of the feature selection technique is to reveal the most essential features that can provide a superior classification performance. On the one hand, selecting a few significant features may not yield sufficient information to build an efficient model; on the other hand, selecting a large number of features may also reduce the classification accuracy, as it may involve unnecessary features. Thus, it is necessary to select the best number of features with sufficient meaningful information. For this purpose, it is essential to develop a good feature selection model that searches and evaluates all subgroups of the features and identifies the important features by eliminating inappropriate features.

C. MAIN AIMS

The goal is to generate and analyse a variety of datasets, considering the number of features, and to develop detection and tracking methods for the classification of walking disorders to enable hospital and clinic doctors to visualise the effects of selecting a specific number of features. The development of the proposed framework involves the pre-processing of image data logged by Kinect cameras. Compared to existing evaluation methods, the proposed framework can be used to create and select important features that can improve classification results. These functions can be generated by extracting skeleton keypoints from patients in each image, including the angle and distance between specific skeleton keypoints and the 2D pixel coordinates of these points. The main contributions of the proposed framework are as follows:

1) We generated different groups of features with skeletal keypoints from a clinician's perspective.
2) We collected two different datasets from detection and tracking methods.
3) We proposed the ten best feature selection algorithms, which selected features from the total number of features that we generated, to analyse different ranges for the number of features.
4) We compared the performances of different well-known deep learning algorithms, such as the LSTM, GRU, and MLP algorithms.

II. MATERIALS AND METHODS

A. MEASUREMENT SCHEME

The whole process was approved by the Ethics Committee of the University Hospital Královské Vinohrady Prague (EK-VP/4310120), where the measurements were made. Each patient signed an informed consent form that described the research conditions. The process was described in [29].

B. DATASET

The dataset contains 84 successful rehabilitation exercises performed by 37 patients (Table 1). All patients (23 males and 14 females who were 21–77 years old) needed surgery for vestibular schwannoma (24 left side / 13 right side; Koos classification: 10 grade 1/10 grade 2/6 grade 3/11 grade 4a tumours; International classification: 10 grade T0/9 grade T1/8 grade T2/9 grade 3/0 grade 4/1 grade 5 tumours).

TABLE 1. Dataset overview.

Start date	16-01-2018
End date	13-12-2020
Number of patients	37
Number of sessions	84
Number of men	23
Number of women	14
Average age	56.6 years

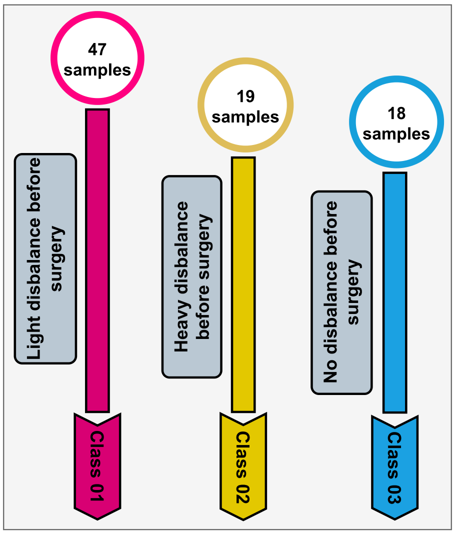
47 samples
19 samples
18 samples
Light disbalance before surgery
Heavy disbalance before surgery
No disbalance before surgery
Class 01
Class 02
Class 03

FIGURE 1. Distribution of classes in our dataset.

All patients had undergone standard neurootological tests to quantitatively assess the brain deficit [30].

Figure 1 illustrates the distribution of labels (classes) in our dataset. There are a total of 84 measurements (samples). Each sample contains three exercises (the patients performed three exercises by walking along a straight line in a hallway during the examination). As can be seen from Figure 1, 47 samples are labelled as class 1 (slight disbalance before surgery), which is split into training and testing sets (34 samples for the training set and 13 samples for the testing set). For classes 2 (heavy disbalance before surgery) and 3 (no disbalance before surgery), there are 19 samples (13 samples for the the training set and 6 for the testing set) and 18 samples (11 for the training set and 7 for the testing set), respectively. As can be seen in Figure 1, the distribution of labels (classes) in our dataset is unbalanced: there are more patients in class 1 (light disbalance) than in the other classes (heavy imbalance and no imbalance). To overcome this problem, we applied the optimum weights technique to each class in the training process, which is recommended in [31].

The patients were scanned during the exercises with a Kinect v2 camera, which was installed on a mobile robotic platform that was developed at UCT Prague [29]. Due to the inability of Kinect v2 to create skeletons correctly in real time, we employed the keypoint-RCNN detector from the PyTorch library [30].

C. DATA PRE-PROCESSING

The most essential step in classification problems is to obtain valuable data that can be used to train deep neural networks. The selection of data for the analysis of gait disorders is based on expert knowledge of the patient's movements during the evaluation. Before data are provided to the network, the selection of important features helps to train the network to distinguish between normal and imbalanced walking during the evaluation and enables a better classification accuracy.

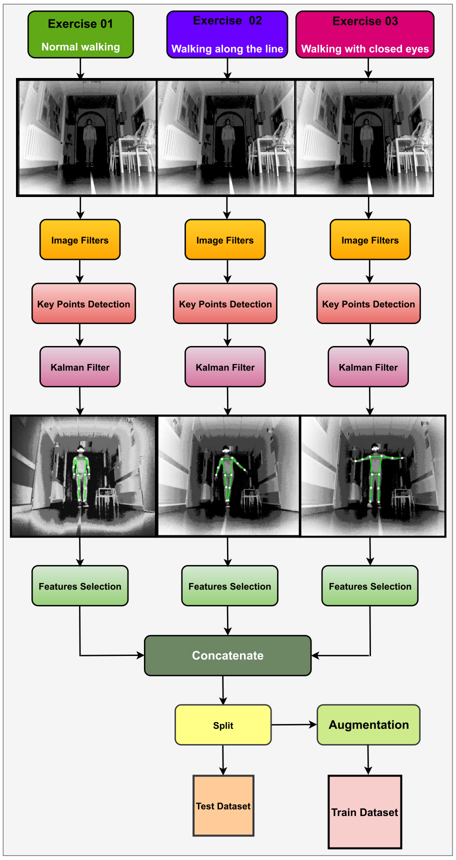

FIGURE 2. Data processing method.

Figure 2 demonstrates the construction of the proposed data processing step before the model is trained. As can be seen, we have images of three exercises in our illustration of the data processing step: the reason for this is that the clinicians made measurements and labelled the category by observing these three exercises, so we have the exact measurements for all three exercises in each class. First, different image filters are applied to get a better image for detection. The Equalhist, GaussianBlur, and applyColorMap filters from the OpenCV library have been used to adjust the contrast of the image, remove unwanted noise, and improve the visual quality of the image to obtain better detection results. Then, these filtered images are used as the inputs for detecting the keypoints to extract the skeletal keypoints from the patient's images. To detect the skeletal keypoints, a keypoint-RCNN detector using the Torchvision library has used. The model is built on top of the ResNet-50 FPN backbone [32]; in [33], the authors extended the model's ability to detect human skeletal keypoints. The keypoint-RCNN is trained on the MS-COCO dataset [34]. If a patient is detected, then we obtain 17 keypoints from the detector's output [33], as shown in Figure 1. In general, the detector model predicts 2D confidence maps for the position of the body and a set of related 2D vectors [33]; then, each confidence map returns the keypoints of the body part [35]. Through the observation and visualisation of the performance of the proposed detector on

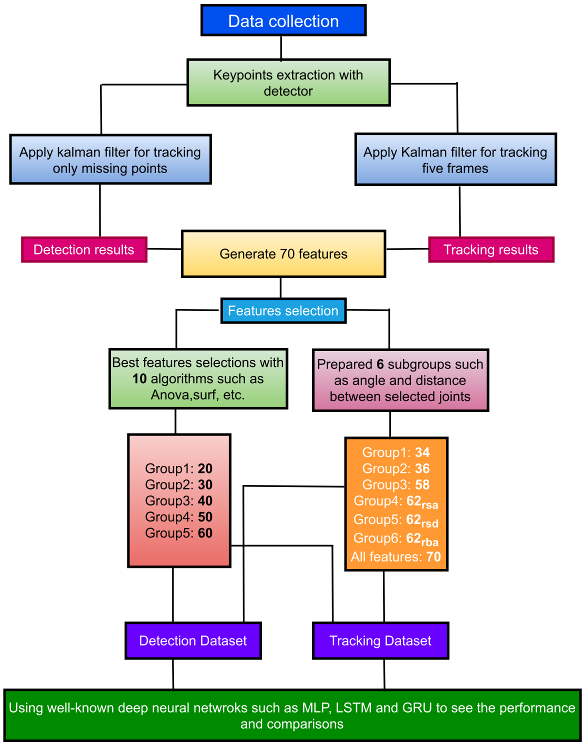
Data collection
Keypoints extraction with detector
Apply kalman filter for tracking only missing points
Apply Kalman filter for tracking five frames
Detection results
Generate 70 features
Tracking results
Features selection
Best features selections with 10 algorithms such as Anova,surf, etc.
Prepared 6 subgroups such as angle and distance between selected joints
Group1: 20
Group2: 30
Group3: 40
Group4: 50
Group5: 60
Group1: 34
Group2: 36
Group3: 58
Group4: 62rsa
Group5: 62rsd
Group6: 62rba
All features: 70
Detection Dataset
Tracking Dataset
Using well-known deep neural netwroks such as MLP, LSTM and GRU to see the performance and comparisons

FIGURE 3. Data collection process.

our data, we chose a window size of 100 sequences to collect the extracted keypoints.

As we are going to investigate the detection and tracking results, the Kalman filter [36], [37] is used to track only the points missing from the detection results for the detection dataset; this is needed because the resulting frames may suffer from missing points. However, in the tracking dataset, the Kalman filter is used to track and update five frames of the patients' movements, and the resulting frames may suffer from a high coordinate variance due to even small inaccuracies in the detection algorithm. A Kalman filter [36], [38] is a regularly used estimation method: by using past estimates of all the keypoints, it can predict the future state of all the keypoints, which smooths the sequence of the coordinates [38].

After this step, the extracted features are subjected to the feature selection and feature engineering methods in each exercise image and are then concatenated. The features are concatenated to obtain more sequences of features. As can be seen from Figure 2, the type of walking is slightly different in different exercises (the patients walk with their eyes closed, walk normally, and walk along a line). Therefore, combining these data helps to provide a long sequence of motion patterns in the training data. In the end, the datasets are split into training and testing sets. Due to the smallness of our datasets and in order to avoid overfitting, we use data augmentation methods, as shown in Table 2, which is recommended in [31]. Figure 3 exhibits the overall structure of our research project.

TABLE 2. Data augmentation methods.

Method	Description
Jittering	One of the most effective data augmentation methods
Rotation	Increases accuracy when combined with other augmentation methods
Scaling	Can change the global intensity of a time series
Magnitude warping	Warps the magnitude of a signal using a smoothed curve
Slicing	Slices time steps off the ends of the pattern
Time warping	Perturbs a pattern in the temporal dimension using a smooth warping

1) MANUAL FEATURE GENERATION

In the classification of gait disorders, all body-related features are valuable. However, some features are more significant than others [39]. The study presented in [40] found that the swing of the arms is the most valuable feature. Researchers or experts can find complex patterns even if they do not exist. Thus, we can use feature creation to derive new features from existing ones. Regarding feature creation, we have applied the concepts of clinicians to categorise gait disorders, as they focus mainly on the patient's hands and legs during exercises.

Patients were required to carry out three exercises at a leisurely pace with the feeling of balance and confidence in a 5-meter hospital hallway. Also, patients trained for the particular exercises before actually performing the recording. Different exercises have different types of walking; for example, in exercise 3, patients walk with closed eyes and cannot walk straight, or they can balance their movements with their hands. Analyzing these different types of movements requires measuring several features, which can be defined as kinematic and spatial-temporal features. The spatial-temporal features are mainly associated with measuring the distance between various body parts, whereas kinematic features refer to the angular movement in the body's joints during walking. These variations, such as posture instability, are very important to assess the evolution of gait disorders. Due to these reasons, we have created more features from skeletal points by measuring the distance of the most effective parts of the body as well as the body joint angle. Actually, these feature creations are the same as physicians' evaluation metrics during the observation of these exercises.

In this study, we have created 36 features to add to 34 base features (x and y from 17 keypoints) to reach 70 features in total. This total number of features is limited to the 17 keypoints that we have. In other words, we have tried to find the most meaningful features that can be found from the skeleton keypoints, and all of these measurements are confirmed by clinicians before they are tested with practical machine learning models. Figure 4 demonstrates the

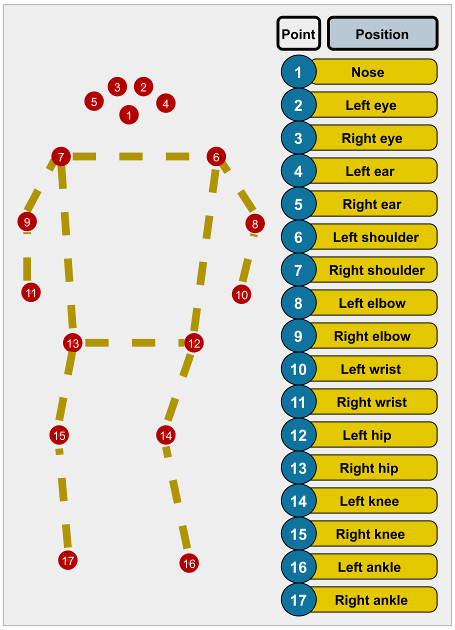

FIGURE 4. Skeleton keypoint information.

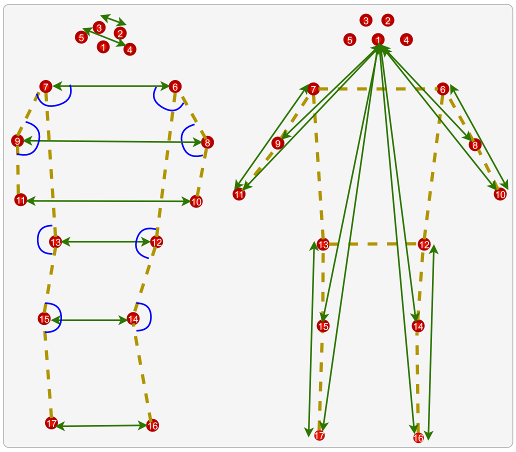

FIGURE 5. Concept of feature extraction.

numbering and keypoint marks. By observing the skeletal keypoints from the patient walking in the hallway, we have generated valuable features that can help to obtain the patterns of gait disorders from the sequences of data that have been collected. Figure 5 shows the skeletal keypoints and the selected points for generating valuable features for training the model. The blue curve shows the angle between two selected bones, and the green arrows illustrate the distance between two selected points.

These spatial features are determined from pixel coordinates and are calculated from four statistical measurements (i.e. x and y for each point, the distance, and the angle between two selected points) as follows [38]:

$$d_i = \sqrt{\Delta x_i^2 + \Delta y_i^2}, \tag{1}$$

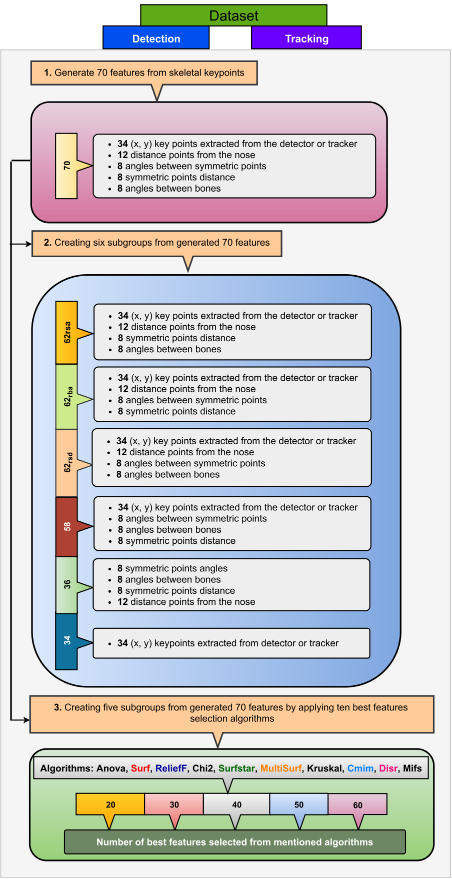
Dataset
Detection
Tracking
1. Generate 70 features from skeletal keypoints
70
• 34 (x, y) key points extracted from the detector or tracker
• 12 distance points from the nose
• 8 angles between symmetric points
• 8 symmetric points distance
• 8 angles between bones
2. Creating six subgroups from generated 70 features
62rsa
• 34 (x, y) key points extracted from the detector or tracker
• 12 distance points from the nose
• 8 symmetric points distance
• 8 angles between bones
62rba
• 34 (x, y) key points extracted from the detector or tracker
• 12 distance points from the nose
• 8 angles between symmetric points
• 8 symmetric points distance
62rsd
• 34 (x, y) key points extracted from the detector or tracker
• 12 distance points from the nose
• 8 angles between symmetric points
• 8 angles between bones
58
• 34 (x, y) key points extracted from the detector or tracker
• 8 angles between symmetric points
• 8 angles between bones
• 8 symmetric points distance
36
• 8 symmetric points angles
• 8 angles between bones
• 8 symmetric points distance
• 12 distance points from the nose
34
• 34 (x, y) keypoints extracted from detector or tracker
3. Creating five subgroups from generated 70 features by applying ten best features selection algorithms
Algorithms: Anova, Surf, ReliefF, Chi2, Surfstar, MultiSurf, Kruskal, Cmim, Disr, Mifs
20
30
40
50
60
Number of best features selected from mentioned algorithms

FIGURE 6. Feature generation and selection steps.

$$\alpha_{1_i} = \arctan 2(\Delta y_i, \Delta x_i), \tag{2}$$

$$\alpha_{2i} = \arccos\left(\frac{v_{1i} \cdot v_{2i}}{|v_{1i}| \times |v_{2i}|}\right), \tag{3}$$

where d_i is the distance based on two keypoints, α_{1i} is the angle between two symmetric keypoints, and α_{2i} is the angle between two bones. The number of features generated in each frame from these measurements is 70. This includes 34 features from the coordinates of 17 pixels, eight angle variables from the horizontal green arrows that indicate pairs of symmetric keypoints, eight angles from the blue curves between selected bones, eight distance variables from the horizontal green arrows that indicate pairs of symmetric keypoints, and 12 distance variables between keypoints connected by vertical green arrows. To analyse and investigate each feature's effect on the classification results, these 70 features are also divided into six subgroups, as shown in Figure 6. In group 1, there are 34 features that represent only the values of the x and y coordinates of the keypoints. In group 2, there are 36 features; in this case, the x and y coordinates were removed from the total of 70 features. Group 3 includes 58 features; 12 vertical distance features were removed from a total of 70 features. Groups 4, 5, and 6 include 62 features from the total of 70 features: eight horizontal distance features are removed in group 4, eight

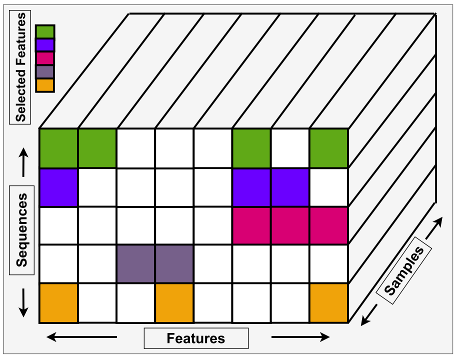

FIGURE 7. An example of the selection of the best features in a data file.

angles of symmetric keypoints are removed in group 5, and eight angles between selected bones are removed in group 6. Each data sample includes data from the three exercises and the labelled class of the data. From our observations, we chose a window size of 100 for each exercise to collect only 300 frames (100 × 3) for each labelled sample, so that the file for each sample has 300 rows (sequences) and 70 columns (features).

2) BEST FEATURE SELECTION ALGORITHMS

In the first step, we have already created 70 features from kinematic and spatial-temporal features as base features but may have a strong correlation between these patterns; due to this reason, we also have applied some of the best feature selection techniques to see the impact of reducing features size to select features with less correlation. Most of the feature selection methods in this section can be found in the open-access repository of scikit-learn and scikit-rebate. These methods have been selected because of their computational efficiency and their popularity in the literature. In addition, the selected methods are employed to order the features according to their significance for the specific goal of this paper [20], [41]. These methods select the best features from the total of 70 generated features that we discussed in the previous subsection, as shown in Figure 7. Figure 6 shows all of the feature generation and selection steps for our datasets. A detailed description of the methods used in this study is provided below.

a: ANOVA

ANOVA is a feature selection method used to reduce the number of features. ANOVA computes the importance of each feature for analysing experimental data under various conditions and then ranks the features. In other words, it is used to decide whether a feature shows a significant difference between two or more classes [42]. ANOVA has become a statistical method that can be used to compare class means using a specific feature [7]. This method ranks the features by determining the ratio of the variance within groups [8]. Algorithm 1 shows the ANOVA algorithm [7].

Algorithm 1 ANOVA Algorithm for Feature Selection

```
for each fi feature set do
    $i = 1, 2, 3, \ldots, G$
    calculate the value of MB
    evaluate the value of MW
    evaluate the $F$-statistic ($Fi$)
    evaluate the $p$-value ($pi$) for each $Fi$
    if $pi < 0.001$ then then
        select the fi feature
        append fi to a feature matrix GM
    else
        fi feature is discarded
    end if
    sort the feature set in ascending order of $p$
    if size $GM$ > defined-feature-size then
        select only these top feature sets
    else
        keep the GM feature matrix as it is
    end if
end for
return the GM feature matrix
```

b: CHI-SQUARE

The chi-square feature selection algorithm has been successfully used in many recent problems for classification and prediction [43], [44]. This method is used to minimise the amount of data and generate a higher classification accuracy [9]. The authors of [45] used chi-square feature selection and a multi-class SVM to improve the training, testing time, and performance of the classifier. The purpose of selecting this method is to identify the optimal subset of attributes based on statistical significance tests and select the features that depend on class labels [11]. When there is a high correlation between the features, overfitting will occur. To overcome this problem, the chi-square method can be used. The chi-square value is sensitive to the sample size. The chi-square index shows a weaker fitting when the correlation between the independent and dependent variables increases [10]. The chi-square equation [43] is defined below:

$$X^2 = \sum_{i=1}^{2} \sum_{j=1}^{k} \frac{(A_{i_j} - E_{i_j})^2}{E_{i_j}}, \tag{4}$$

where k is the number of classes, A_{i_j} is the number of patterns in the i_{th} interval and j_{th} class, and E_{i_j} is the expected frequency of A_{i_j}.

c: CMIM

CMIM is another method that can select the optimal subset of features [14]. This method ensures a good trade-off between the independent and dependent variables. This method does not select features similar to the ones already selected, even if a feature is individually powerful, because it does not provide additional information about the expected class. This estimation takes into account families M composed of unique features that have already been selected [12]. The CMIM equation [15] is

$$X_{CMIM} = \arg\max_{X_i \in X_{-s}} \{\min_{X_j \in X_s} I(X_i; Y \| X_j)\}, \tag{5}$$

where Y is the output, X_i is the input, and X_j is a selected feature.

d: DISR

The authors of [15] introduced a new effective feature selection approach called DISR for classification with a large number of input variables. This method has shown that the information returned by a set of variables is greater than the sum of the information of its individual variables. The proposed method is competitive with existing methods. The authors of [5] mentioned that the goal of DISR is to alternate joint mutual information with symmetrical relevance. The DISR equation [15] is

$$X_{DISR} = \arg\max_{X_i \in X_{-s}} \{\sum_{X_j \in X_s} SR(X_{i_j}, Y)\}, \tag{6}$$

$$SR(X, Y) = \frac{I(X_{i_j}; Y)}{H(X_{i_j}; Y)}, \tag{7}$$

where $SR(X, Y)$ is the symmetrical relevance of the two random variables X and Y.

e: KRUSKAL

The authors of [16] mentioned that the Kruskal method is a significant feature selection approach, and it is simple and inexpensive to apply. They used it to reduce the data dimensions to select the most important features. They mentioned that in their algorithm, if the p-value is close to zero, features are selected, as they have discriminative information; otherwise, they are not selected. The Kruskal equation [46] is

$$X_{Kruskal} = \frac{12}{n(n+1)} \sum_{i=1}^{n} \frac{T_i^2}{n_i} - 3(n+1), \tag{8}$$

where n is the total number of observations, while T and n_i are the sum of the ranks and the number of observations within class i, respectively.

f: MIFS

Mutual information-based feature selection (MIFS) is a powerful statistical technique used to identify the relationships between sets of features. It is also called the elimination method, and it is used to extract the mutual relations between features and reduce the input size; however, it does not change the most significant features of classification problems. The feature selection method must select only independent features that indicate no correlation, and this can be done by determining the dependence between random features, which must always be non-negative and symmetrical [47], [48]. This method is fast and efficient and is used to select the best features that do not exist in current individuals. However, its performance decreases in the case of a group of features that is relevant but does not include the individual features that result in the group [13]. The MIFS equation [48] is

$$X_{MIFS} = I(C; X_i) - \beta \sum_{X_s \in S_{i-1}} I(X_s; X_i), \tag{9}$$

where X_i is any non-selected feature, S_{i-1} is the set of features selected in the previous steps, and β is a parameter that can be manually tuned.

3) RELIEF-BASED METHODS

a: RELIEFF

ReliefF is an extension of the Relief method, which is designed to deal with multi-class data by using $k > 1$ neighbours [18], [49]. It is also capable of dealing with incomplete and noisy datasets [17]. This method is more accurate with smaller feature sets [50]. By defining a suitable threshold and assigning weights to each feature, it can select the quality features above the threshold [17]. Since it remains small but not too small, it will be robust in terms of the number of nearest neighbours [51]. Algorithm 2 shows the ReliefF algorithm [17].

Algorithm 2 ReliefF Algorithm for Feature Selection

```
calculate the prior probabilities $P(C)$ for all classes
set all weights $W[A] := 0.0$
for $i = 1$ to $m$ do
    randomly select an instance $R_i$
    find $k$ nearest hits $H_j$
    for all classes $C \neq cl(R_i)$ do
        from class $C$, find $k$ nearest misses $M_j(C)$
    end for
    for $A := 1$ to $a$ do
      $\mathbf{H := -\sum_{j=1}^{k} diff(A, R_i, H_j)/k}$
      $\mathbf{M := \sum_{C \neq cl(R_i)} [\frac{P(C}{1-P(cl(R_i))}}$
          $\mathbf{\sum_{j=1}^{k} diff(A, R_i, M_j(C))]/k}$
      $\mathbf{W[A] := W[A] + (H + M)/m}$
    end for
end for
return $W$
```

b: SURF

The SURF approach keeps most of the ReliefF method. Unlike ReliefF, SURF eliminates the user parameters k and uses distance thresholds T to determine which samples are considered neighbouring samples. The thresholds have the same magnitude for each target sample and are defined by the average distance between all sample pairs in the data [20].

c: MULTISURF

MultiSURF is the newest Relief-based method: it determines the target sample-centric neighbourhood to estimate the features [19], [52]. The authors of [19] mentioned that a large number of redundant features in the dataset reduce the performance of this method for selecting the relevant features. This method uses a threshold or adaptive radius to verify which samples are considered neighbours [53].

d: SURF*

The SURF* method gets the most out of the SURF approach. Unlike SURF, SURF* considers the concept of samples that are near versus far from the target samples. Using the same threshold from SURF, any sample within the threshold is considered near, and those outside the threshold are considered far [20]. Figure 8 shows illustrations of the feature selection processes of ReliefF-based algorithms [20].

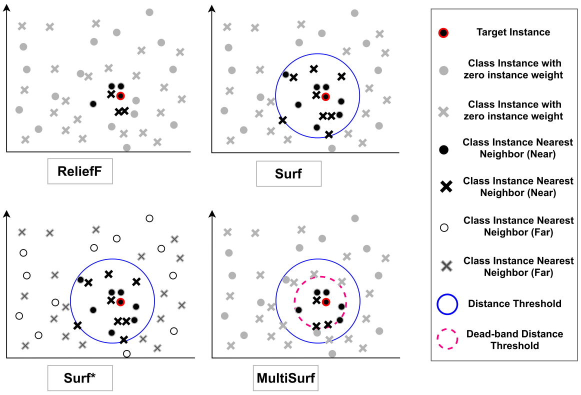
ReliefF
Surf
Surf*
MultiSurf
Target Instance
Class Instance with zero instance weight
Class Instance with zero instance weight
Class Instance Nearest Neighbor (Near)
Class Instance Nearest Neighbor (Near)
Class Instance Nearest Neighbor (Far)
Class Instance Nearest Neighbor (Far)
Distance Threshold
Dead-band Distance Threshold

FIGURE 8. Illustrations of ReliefF-based algorithms' feature selection methods.

D. CLASSIFICATION METHODOLOGY

1) MLP

The MLP is the most frequently used neural network. It has the ability to represent nonlinear functions [54]. The universal MLP consists of one input layer, one hidden layer, and one output layer. An MLP with more than one hidden layer is considered a deep neural network. Each hidden layer is composed of a certain number of neurons. The output of an MLP, as it is a classifier, defines the classes belonging to the input data. Figure 9 shows the proposed MLP used in this study; it is made of three hidden layers with the ReLU activation function. Given a sequence of vectors $v_1, \ldots, v_n$, each including input features $(x_1, x_2, \ldots, x_n)$, the computation of an MLP sublayer on any v_i is defined as

$$Output_j = ReLU(\sum_{i=1}^{n} w_{i_j} x_i + b_j), \tag{10}$$

where W_{i_j} and b_j are the weight parameter from the input to the hidden layer and the bias parameter, respectively, and n is the number of features in the input vector feature.

2) LSTM

The LSTM network is a modified version of the RNN that adds memory cells and gate units to perform better in finding the temporal dependency over a long period of time [30]. Figure 11 shows the proposed LSTM algorithm that was used in this study. As shown in Figure 11, before input data x_t were fed into the LSTM network, a linear layer with a ReLU activation function from time frame t was added; this layer is given as follows:

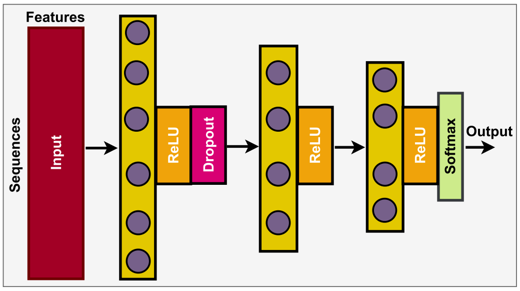

FIGURE 9. Proposed MLP network.

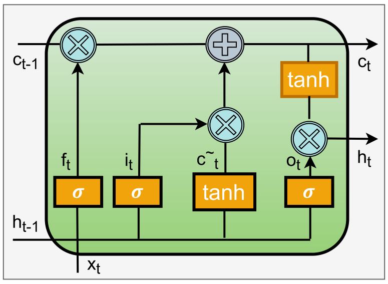

FIGURE 10. LSTM cell architecture.

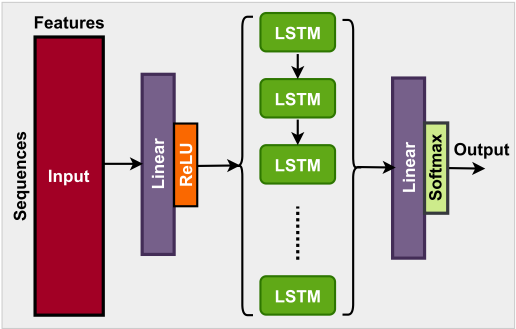

FIGURE 11. Proposed LSTM network.

$$a_t = ReLU(w_a x_t + b_a), \tag{11}$$

where w_a and b_a are weights and biases, respectively. The structure of each LSTM cell [30], [55] in Figure 11 is shown in Figure 10. The outputs of the LSTM cell at time t are determined [30] as follows:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \tag{12}$$

$$h_t = o_t \odot \tanh(c_t), \tag{13}$$

where i_t, o_t, c_t, and h_t are the input gate, output gate, state of the memory cell, and hidden layer value, respectively. Finally,

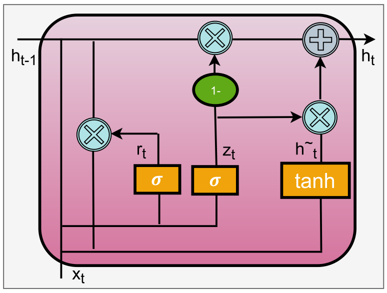

FIGURE 12. GRU cell architecture.

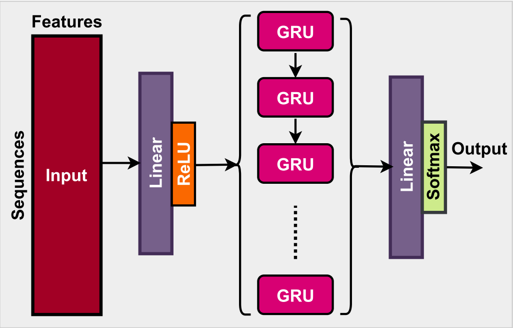

FIGURE 13. Proposed GRU network.

the predicted output vector is given by

$$output = Softmax(w_f c \cdot h_t + b_f c), \tag{14}$$

where w_{fc} and b_{fc} denote the weights and biases of the sublayers of the network.

3) GRU

The GRU is another RNN architecture that is useful for handling data over a long period of time. It, like the LSTM network, can deal with the problem of a vanishing gradient in an RNN [26]. Like the proposed LSTM algorithm, in the proposed GRU algorithm, the input data x_t are fed into a linear layer containing the ReLU activation function. The structure of each GRU cell [55] in Figure 13 is shown in Figure 12. The outputs of the GRU cells at time t are determined [26], [55] as follows:

$$\begin{aligned} c_t &= (1 - z_t) \odot c_{t-1} + z_t \odot \\ &\quad \tanh(W_c \cdot x_t + W_h \odot (r_t \odot h_{t-1}) + b_c), \qquad (15) \\ h_t &= c_t, \qquad (16) \end{aligned}$$

where z_t and r_t are the update gate and reset gate, respectively. When the last hidden cell is determined, it is fed into a linear layer with a SoftMax active function layer for classification [26]. Finally, the predicted output vector is given by

$$output = Softmax(w_f h \cdot h_t + b_f h), \tag{17}$$

where w_{fh} and b_{fh} denote the weights and biases, respectively, of the sublayers of the network.

III. RESULTS

The selected and modified deep learning classifiers (MLP, LSTM, GRU) were trained to classify three labelled gait disorders in our dataset. One hundred frames have been selected for each exercise because of environmental noise in some frames. For recorded exercises with less than 100 frames of detection results, zero padding was added to make the time steps the same for all data. Twenty-six of the 84 samples were randomly selected for the testing set, whereas the remaining 58 were kept for the training set (10% were randomly selected to be used for tuning the hyperparameters in the validation process). Due to the fact that the data are imbalanced, more samples in class 1 were selected in both the training and testing sets. The 5-fold cross-validation method is used for the training process to prevent overfitting and validate unseen data. PyTorch [56], a free open-source framework based on the Torch library written in Python, is used for implementing deep neural network algorithms. The system used to train the models consists of an Intel(R) Core i7–CPU@2.60 GHz, 16 GB of RAM, and an NVIDIA GeForce GTX 1660Ti. To obtain a fixed initialisation for the network parameters, the random seed was set to 21. Moreover, to accelerate the training process and optimise the trained models, the Adamw optimiser [57] was used in the training process. Furthermore, to validate the performance and achieve the optimised validation accuracy, the regularisation value $1e - 5$ was added for weight decay.

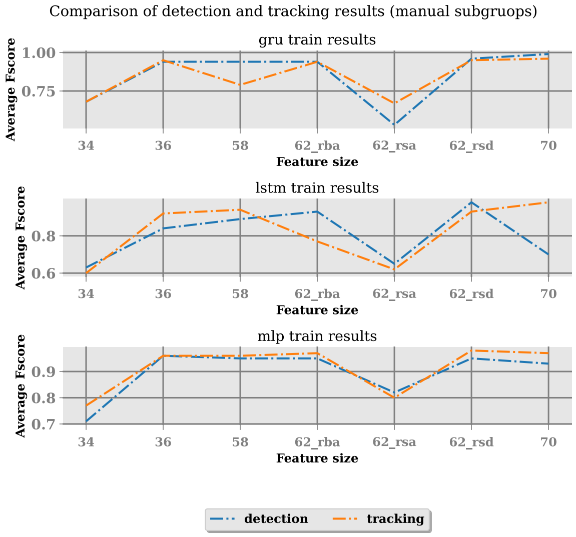

FIGURE 14. Comparison of results on detection and tracking datasets: training results.

A. MANUAL FEATURE EXTRACTION RESULTS

The number of data points in a single frame is 34 because the 2D coordinates of 17 joints make up the skeleton. We add 36 features so that we have a maximum number of features from our data of 70. Then, these 70 features are also divided into six subgroups according to the concepts with which they are generated. Figures 14 and 15 show a comparison

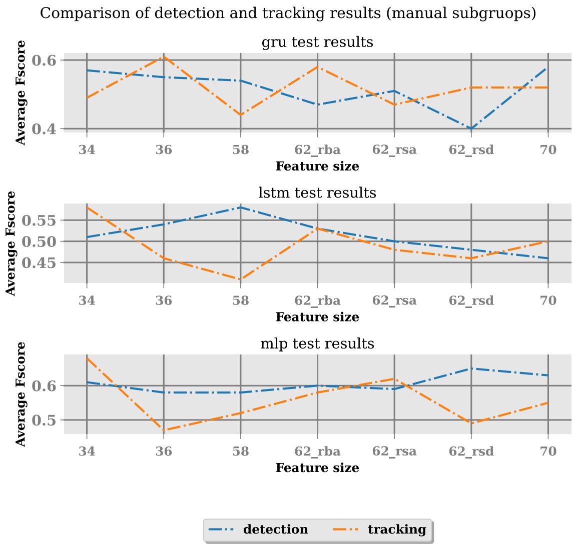

FIGURE 15. Comparison of results on detection and tracking datasets: testing results.

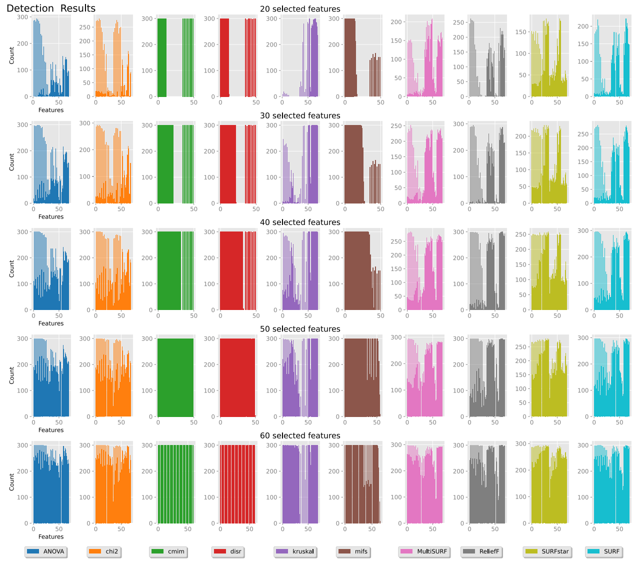

FIGURE 16. Different numbers of selected features: detection results.

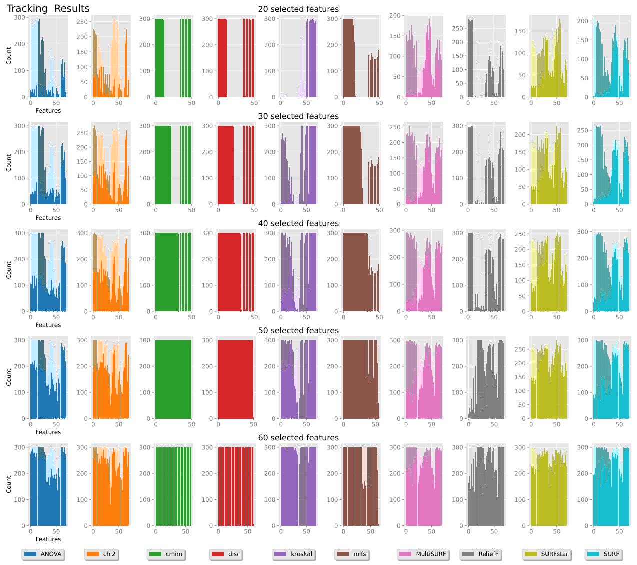

FIGURE 17. Different numbers of selected features: tracking results.

of the performance of the proposed classifier algorithms on both detection and tracking datasets. As can be seen from Figure 14, all classifiers were trained successfully on a high number of features, except for 62_{rsa} features, for which the model training F-score dropped by around 20%. Figure 15 shows the test results of the classifiers, where the MLP and GRU show a better performance than the LSTM network. In the GRU, the maximum F-score was achieved with 70 features for the detection dataset, whereas the maximum value for the tracking dataset was achieved with 36 features. The MLP classifier attains its maximum F-score with 34 features for the tracking dataset; the minimum value is achieved for the tracking dataset with 36 features. In contrast, the highest values for the detection dataset were achieved with 62_{rsd} and 70 features, respectively. In the LSTM results, both the minimum and maximum values were achieved with 58 features.

As is obvious from Figure 15, the MLP performs better than the other algorithms. Furthermore, generally, we can say that the proposed classifiers perform better on the detection dataset than on the tracking dataset.

B. ALGORITHM-BASED FEATURE EXTRACTION RESULTS

We also extracted and generated features using the ten best feature selection algorithms in a single frame, selecting from 20 to 60 of the total number of features (70) we had already generated. Figures 16 and 17 show the results of the ten best feature selection algorithms on the detection and tracking datasets, respectively. On both the detection and tracking datasets, the features selected by different algorithms have approximately similar information. As can be seen, the results from using 20 and 30 features show the differences between these algorithms clearly.

For instance, the results of the MIFS algorithm (shown in brown) show that most of the best features are selected from the first feature points in the feature vector, whereas the results of the Kruskal algorithm show that most features are selected from the last feature points. The results of the CMIM and DISR algorithms (shown in green and red) show that the selected features are from the first and last feature points in the feature vector. For the other algorithms, we can say that the selected features are distributed among the range of feature points in all feature vectors. In other words, these algorithms select different points in most frames.

Figure 18 shows the training classification F-score on the detection dataset as the number of extracted features changes. The results show that most algorithms worked well with the GRU network, except the Kruskal algorithm. The F-scores are mostly above 90% as the number of extracted features

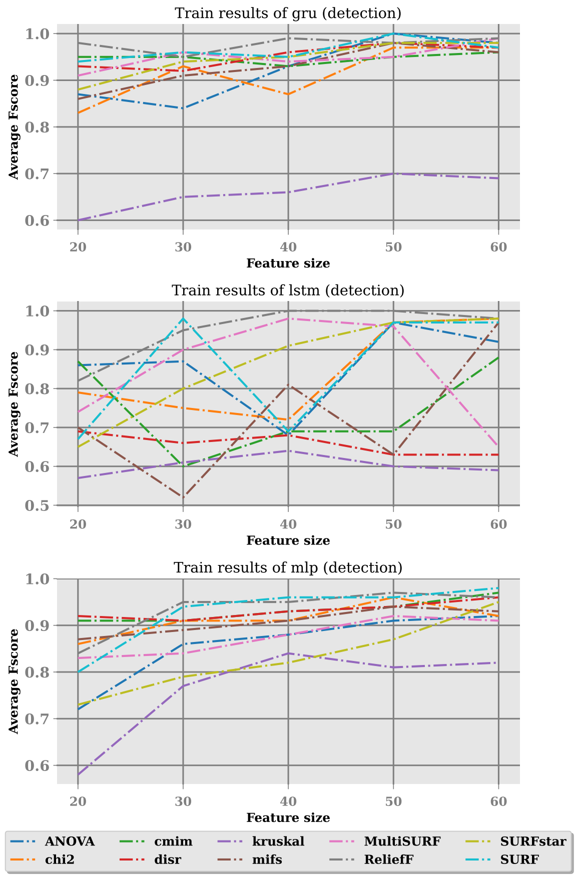

FIGURE 18. Results of three deep networks trained with best selected features for the detection dataset: training dataset.

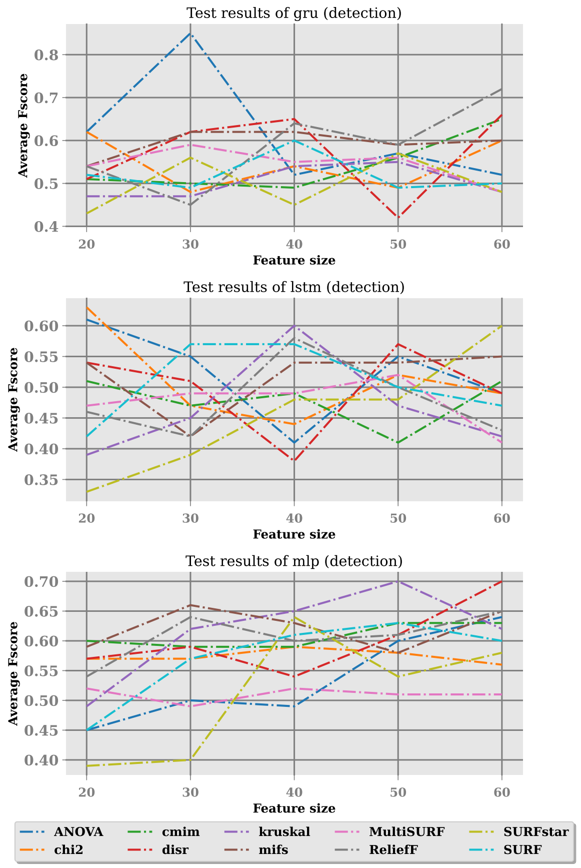

FIGURE 19. Results of three deep networks trained with best selected features for the detection dataset: test dataset.

changes. The GRU network is followed by the MLP network, where the number of features is 30 or more: approximately half of the algorithms obtained F-scores above 90%, and the others obtained F-scores above 80%. For the LSTM network, the results show fluctuations in the algorithm training results as the number of extracted features changes. Figure 19 shows the test classification F-score on the detection dataset as the number of extracted features changes. As can be seen, the extracted features of the ANOVA method combined with the GRU network show the highest classification F-score on the testing set when the number of features is 30. In other words, the results show that the features of the ANOVA method can reduce the nonlinearity caused by the sequential features of the gait data and can help the GRU network to classify the gait data more easily. This network obtained an F-score of around 85% during training but showed the best test result. The reason for this could be that the optimal training of the algorithm occurred rather than overfitting; the other algorithms obtained high classification F-score results in training but did not perform as well on the testing set. The second-highest classification F-score is achieved by the ANOVA and GRU combination; the F-score is above 70% when the number of features is 60. This is followed by the MLP network with the Kruskal and DSIR algorithms, which both achieved a classification F-score of around 70% with a number of features of 50 and 60, respectively. The test results of the LSTM network with these 10 algorithms showed worse results; the classification F-scores are mostly below 55%. Generally, the features selected by most algorithms provide a better performance; the classification F-score is mostly above 60% with the MLP network.

Figure 20 shows the training classification F-score for the tracking dataset as the number of extracted features changes. The results have a trend similar to that of the training performance of the three networks trained on the detection data. However, as can be seen from Figure 21, the test classification F-scores of these three networks for the tracking dataset are mostly within the range 45% to 60%, and there is no significant classification F-score result compared with the results on the detection dataset.

As can be seen from Figure 22, the average value of the proposed 10 algorithms for the detection results is better than that for the tracking results. This is because the format of the exercises in our dataset was designed by physicians. In other words, the movement of patients who suddenly change the positions of their hands or their legs results in the loss of the proper tracking results, as tracking updates are configured every five frames.

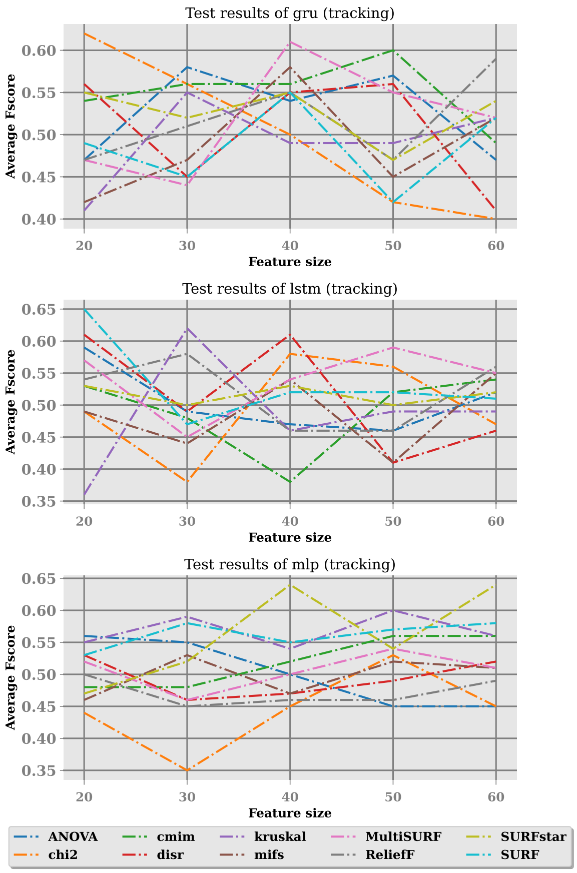

FIGURE 20. Results of three deep networks trained with best selected features for the tracking dataset: training dataset.

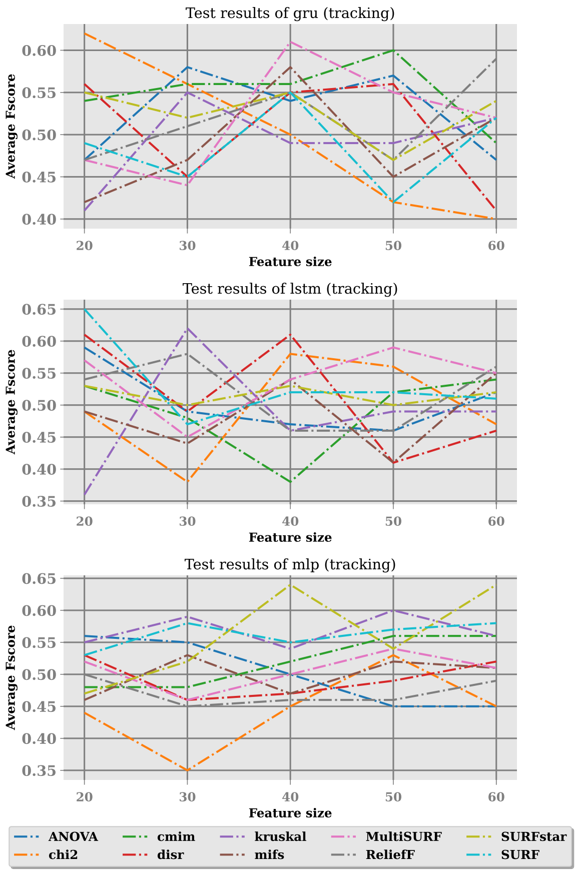

FIGURE 21. Results of three deep networks trained with best selected features for the tracking dataset: test dataset.

IV. DISCUSSION

In this section, we discuss the results of three classifiers (MLP, LSTM, and GRU) on only the detection dataset using the proposed manual feature selection method and also the best feature selection techniques. We only show the results on the detection dataset, as the above results on the tracking dataset have shown that the three classifiers have a low performance compared with the results on the detection dataset. We compare our manually extracted features with the best features selected by the algorithms in terms of the classification F-score. Moreover, we compare our manual feature extraction method with the automatically extracted features obtained by using the 10 best feature selection algorithms. In this study, as mentioned in the previous section, using ANOVA to extract 30 features achieves the highest performance. In the experiments, we have not individually set the learning configuration of each model, which consists of settings such as the number of training epochs and the learning rate, because there are many models that were trained sequentially using a single GPU. According to the test results, the features of the GRU network achieve the best performance in the classification of gait disorders. The highest F-score is achieved when the features of the ANOVA algorithm are fed into the GRU network. Feeding the ReliefF features into the GRU network achieves the second-best results; however, these two selection algorithms have quite a low performance when they are combined with the MLP and LSTM networks. In most cases, the MIFS and

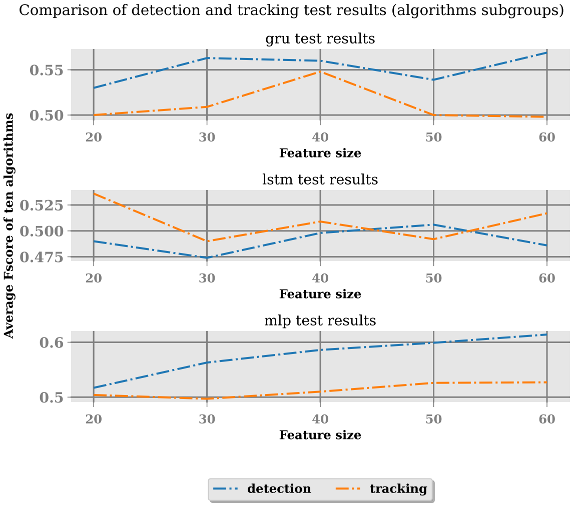

FIGURE 22. Comparison of detection and tracking results: average values for the 10 best feature selection algorithms.

TABLE 3. Detection results of LSTM classification for 10 methods.

		LSTM			
Method	Feat.	Accuracy	F-score	Precision	Recall
ANOVA	60	0.58	0.42	0.36	0.5
chi2	60	0.54	0.42	0.47	0.43
Kruskal	60	0.46	0.36	0.43	0.49
CMIM	60	0.5	0.48	0.48	0.48
DISR	60	0.58	0.42	0.36	0.5
MIFS	60	0.54	0.5	0.51	0.51
ReliefF	60	0.46	0.35	0.35	0.36
SURF	60	0.5	0.35	0.33	0.38
SURF*	60	**0.62**	**0.55**	**0.6**	**0.54**
MultiSURF	60	0.54	0.32	0.35	0.39
ANOVA	50	0.58	**0.51**	0.6	**0.57**
chi2	50	0.54	0.46	0.47	0.46
Kruskal	50	0.54	0.39	0.34	0.48
CMIM	50	0.5	0.22	0.17	0.33
DISR	50	**0.65**	0.48	0.43	**0.57**
MIFS	50	0.54	0.5	0.5	0.51
ReliefF	50	0.54	0.43	0.45	0.44
SURF	50	0.54	0.4	0.37	0.47
SURF*	50	0.5	0.45	0.46	0.5
MultiSURF	50	0.54	0.47	**0.62**	0.46
ANOVA	40	0.46	0.33	0.3	0.37
chi2	40	0.42	0.43	**0.58**	0.47
Kruskal	40	**0.69**	**0.52**	0.46	**0.62**
CMIM	40	0.5	0.43	0.42	0.43
DISR	40	0.5	0.22	0.17	0.33
MIFS	40	0.54	0.51	0.52	0.53
ReliefF	40	0.62	**0.52**	0.55	0.53
SURF	40	0.65	0.48	0.43	0.55
SURF*	40	0.54	0.41	0.37	0.47
MultiSURF	40	0.54	0.4	0.37	0.45
ANOVA	30	**0.58**	**0.51**	**0.52**	**0.55**
chi2	30	0.5	0.4	0.39	0.47
Kruskal	30	0.5	0.39	0.42	0.49
CMIM	30	0.54	0.4	0.35	0.48
DISR	30	0.5	0.49	0.49	0.5
MIFS	30	0.54	0.31	0.28	0.39
ReliefF	30	0.42	0.41	0.41	0.41
SURF	30	**0.58**	**0.51**	**0.52**	0.51
SURF*	30	0.46	0.33	0.29	0.4
MultiSURF	30	0.5	0.45	0.48	0.5
ANOVA	20	**0.62**	**0.56**	**0.58**	0.56
chi2	20	**0.62**	**0.56**	0.57	0.57
Kruskal	20	0.5	0.22	0.17	0.33
CMIM	20	0.5	0.48	0.53	0.5
DISR	20	0.58	0.47	0.54	0.48
MIFS	20	0.54	0.54	**0.58**	**0.59**
ReliefF	20	0.46	0.43	0.43	0.43
SURF	20	0.5	0.3	0.27	0.36
SURF*	20	0.5	0.22	0.17	0.33
MultiSURF	20	0.54	0.4	0.35	0.48

TABLE 4. Detection results of GRU classification for 10 methods.

		GRU			
Method	Feat.	Accuracy	F-score	Precision	Recall
ANOVA	60	0.58	0.43	0.41	0.47
chi2	60	0.62	0.55	0.6	0.54
Kruskal	60	0.5	0.41	0.39	0.44
CMIM	60	0.65	0.59	0.6	0.59
DISR	60	0.65	0.63	0.62	0.64
MIFS	60	0.58	0.53	0.54	0.53
ReliefF	60	**0.73**	**0.69**	**0.78**	**0.7**
SURF	60	0.58	0.43	0.38	0.49
SURF*	60	0.5	0.45	0.52	0.44
MultiSURF	60	0.5	0.41	0.61	0.41
ANOVA	50	0.58	0.51	0.52	0.51
chi2	50	0.5	0.41	0.48	0.41
Kruskal	50	0.62	0.46	0.41	0.54
CMIM	50	0.58	0.51	**0.55**	0.5
DISR	50	0.5	0.34	0.31	0.39
MIFS	50	0.58	**0.53**	0.53	0.53
ReliefF	50	0.62	0.51	0.52	0.51
SURF	50	0.5	0.44	0.46	0.44
SURF*	50	**0.65**	0.48	0.43	**0.55**
MultiSURF	50	0.58	0.5	0.5	0.5
ANOVA	40	0.54	0.46	0.49	0.46
chi2	40	0.54	0.46	0.47	0.46
Kruskal	40	0.58	0.44	0.42	0.52
CMIM	40	0.5	0.43	0.43	0.43
DISR	40	**0.65**	**0.6**	**0.61**	**0.61**
MIFS	40	0.62	0.55	0.56	0.56
ReliefF	40	**0.65**	**0.6**	**0.61**	0.59
SURF	40	0.65	0.49	0.48	0.59
SURF*	40	0.5	0.37	0.35	0.4
MultiSURF	40	**0.65**	0.48	0.44	0.56
ANOVA	30	**0.85**	**0.85**	**0.83**	**0.87**
chi2	30	0.54	0.39	0.37	0.42
Kruskal	30	0.58	0.37	0.36	0.43
CMIM	30	0.5	0.42	0.42	0.43
DISR	30	0.65	0.56	0.56	0.58
MIFS	30	0.62	0.58	0.59	0.6
ReliefF	30	0.46	0.42	0.46	0.41
SURF	30	0.5	0.45	0.45	0.45
SURF*	30	0.58	0.5	0.51	0.5
MultiSURF	30	0.62	0.54	0.62	0.59
ANOVA	20	0.62	**0.58**	**0.59**	**0.6**
chi2	20	**0.65**	0.54	0.54	0.58
Kruskal	20	0.58	0.37	0.36	0.43
CMIM	20	0.54	0.42	0.44	0.43
DISR	20	0.54	0.44	0.43	0.45
MIFS	20	0.54	0.5	0.5	0.51
ReliefF	20	0.54	0.45	0.45	0.45
SURF	20	0.54	0.49	0.48	0.5
SURF*	20	0.42	0.39	0.42	0.37
MultiSURF	20	0.54	0.49	0.51	0.48

ReliefF features achieve a better performance when they are fed into the GRU and MLP networks compared to the other features.

As shown in Table 3, the features of the ANOVA and chi2 algorithms show a much better performance than the other feature selection algorithms for the LSTM network. These algorithms achieved a classification accuracy of 62% and an F-score of 56% with only 20 features. They are followed by the Surf* algorithm with 60 features, which achieved an accuracy of 62% and an F-score of 55%. With other numbers of features, considering only the F-score results, the Kruskal and ReliefF algorithms obtained an F-score of 52%, and in two cases, ANOVA achieved an F-score of 51%. From the results, it is clear that in most cases, with different numbers of features, the features selected by ANOVA perform better than those selected by the other feature selection algorithms.

Table 4 shows the results of feature selection algorithms combined with the GRU network. The 30 features selected by ANOVA show a significantly better performance than those selected by the other algorithms, with a classification accuracy and F-score of 85%. The second-best F-score (69%) was achieved with 60 features selected by ReliefF, and the accuracy of this method was around 73%. This method is followed by ReliefF and DISR, which both achieved a classification F-score of 60% and a classification accuracy of 65%. The methods with other numbers of features obtained F-scores below 60%.

As shown in Table 5, for the MLP network, the maximum F-score value was achieved by the Kruskal algorithm when

TABLE 5. Detection results of MLP classification for 10 methods.

		MLP			
Method	Feat.	Accuracy	F-score	Precision	Recall
ANOVA	60	0.65	0.55	0.54	0.55
chi2	60	0.58	0.49	0.49	0.5
Kruskal	60	0.62	0.6	0.61	0.62
CMIM	60	0.65	0.57	**0.71**	0.57
DISR	60	**0.69**	**0.65**	0.67	**0.64**
MIFS	60	0.65	0.58	0.58	0.58
ReliefF	60	0.65	0.57	0.61	0.56
SURF	60	0.62	0.54	0.54	0.54
SURF*	60	0.58	0.51	0.58	0.52
MultiSURF	60	0.54	0.4	0.37	0.45
ANOVA	50	0.62	0.52	0.52	0.53
chi2	50	0.58	0.51	0.52	0.51
Kruskal	50	**0.69**	**0.7**	**0.72**	**0.71**
CMIM	50	0.65	0.55	0.56	0.55
DISR	50	0.62	0.52	0.52	0.53
MIFS	50	0.58	0.5	0.5	0.5
ReliefF	50	0.62	0.54	0.56	0.54
SURF	50	0.65	0.56	0.59	0.55
SURF*	50	0.58	0.48	0.51	0.48
MultiSURF	50	0.5	0.43	0.44	0.43
ANOVA	40	0.46	0.42	0.46	0.43
chi2	40	0.58	0.53	0.54	0.53
Kruskal	40	0.65	0.65	0.64	**0.67**
CMIM	40	0.58	0.52	0.53	0.51
DISR	40	0.58	0.45	0.47	0.46
MIFS	40	0.62	0.52	0.53	0.53
ReliefF	40	0.62	0.52	0.51	0.53
SURF	40	0.62	0.56	0.56	0.56
SURF*	40	**0.65**	**0.6**	**0.65**	0.61
MultiSURF	40	0.54	0.45	0.46	0.45
ANOVA	30	0.5	0.43	0.43	0.43
chi2	30	0.58	0.45	0.43	0.47
Kruskal	30	0.62	**0.61**	**0.61**	**0.62**
CMIM	30	0.62	0.5	0.52	0.51
DISR	30	0.62	0.51	0.55	0.51
MIFS	30	**0.65**	0.59	**0.61**	0.59
ReliefF	30	**0.65**	0.56	0.58	0.56
SURF	30	0.58	0.48	0.48	0.48
SURF*	30	0.46	0.35	0.31	0.43
MultiSURF	30	0.5	0.44	0.44	0.45
ANOVA	20	0.42	0.39	0.42	0.39
chi2	20	0.58	0.48	0.48	0.48
Kruskal	20	0.58	0.4	0.51	0.44
CMIM	20	**0.62**	**0.53**	**0.53**	**0.53**
DISR	20	0.58	0.49	0.49	0.5
MIFS	20	**0.62**	0.5	0.5	0.51
ReliefF	20	0.54	0.45	0.45	0.45
SURF	20	0.42	0.37	0.39	0.36
SURF*	20	0.42	0.34	0.35	0.43
MultiSURF	20	0.5	0.45	0.47	0.45

50 features were selected, and its accuracy was around 60%. The second-best F-score, 65%, was achieved by the DSIR algorithm with 60 features. It is clear that the top two F-scores were obtained by higher numbers of features, whereas other methods obtained F-scores below 61%. From the three tables mentioned above, the top two precision and recall values were obtained by the GRU network with 30 and 60 features selected by ANOVA and ReliefF, respectively.

Table 6 shows the results of using manually extracted features to train the above-mentioned three deep learning algorithms. The best F-score was obtained by the MLP network trained with 62 features, where the symmetric distance feature was removed from the total of 70 features. This method obtained a classification accuracy of 58% and an F-score of 65%. The best result achieved by manual feature extraction is far from the results achieved with the best selection algorithms, as described above.

The corresponding confusion matrices of the three best deep learning networks are shown in Table 7. The overall classification F-score is higher for the GRU trained with the 30 features selected by ANOVA. This method has the best results. It obtains 10 correct predictions from 13 samples for class 1 and five correct predictions from six samples for class 2, and it correctly predicts 100% of the samples in class 3. Overall, the GRU was effective for class 1 and class 3 for all feature selection algorithms. It shows a poor performance for class 2, except when it is combined with the ANOVA method; in that case, there is only one mistake among six samples. The second-best result is obtained by the MLP network, which also shows better prediction results for classes 1 and 3. The LSTM results represent the worse case: there are zero correct predictions in classes 1 and 3 for some feature selection algorithms, such as the Kruskal and Surf* algorithms. In other words, they show the highest misclassification results, where samples from all classes are predicted to be in class 1. In general, as can be seen, class 2 has the lowest accuracy; class 2 samples are misclassified as class 1 samples in most cases. Actually, both classes represent abnormal cases, so they may have similar patterns.

To summarise, this study aims to compare the effects of different feature sizes on the proposed networks. To prepare the datasets, ten feature selection methods were used to select five different numbers of features, and seven manual features were added, which resulted in a total of around 114 (57 × 2) different datasets, including detection and tracking data, being generated to train three deep network algorithms. In other words, a total of around 342 (114 × 3) models were trained. We have designed and modified three deep neural network models, and all datasets were used to train and test these networks. Therefore, the number of features affects the training and test results, as the network structure is fixed. In other words, the same network with different datasets can illustrate the optimal, underfitting, or overfitting process. Due to this, the limited number of input features, and our tiny dataset, we observed overfitting and underfitting phenomena for some models. As can be seen from the results, both the quantities and qualities of selected features played an essential role in the performance of the deep networks. The datasets with proper features show a good performance for the training and test results. The underfitted models might be enhanced by increasing the number of layers and neurons in the proposed networks. Furthermore, the overfitted models might also be enhanced by decreasing the number of neurons and layers. However, this is outside of the objectives and contributions of this study. We used a system with only one GPU, and for this reason, we limited our contributions to applying different datasets to fixed-structure networks that were trained sequentially.

Regarding the training algorithms, all the above-mentioned datasets are prepared using manual features and the best feature selection methods. The data from detection and tracking

TABLE 6. Detection results of manual features.

Feature size	MLP				LSTM				GRU			
	Accuracy	F-score	Precision	Recall	Accuracy	F-score	Precision	Recall	Accuracy	F-score	Precision	Recall
70	**0.65**	0.55	0.56	0.55	**0.58**	0.38	0.36	0.44	0.62	0.51	0.56	0.53
34	0.62	0.56	0.56	0.57	**0.58**	0.42	0.37	0.49	**0.65**	0.48	0.43	0.55
36	0.62	0.51	0.57	0.51	0.54	**0.53**	**0.62**	**0.58**	0.58	0.49	0.54	0.5
58	0.62	0.51	0.56	0.51	**0.58**	**0.53**	0.53	0.53	0.58	0.47	0.52	0.48
62_{rsa}	0.58	0.56	0.56	0.57	**0.58**	0.43	0.38	0.52	0.54	**0.54**	**0.67**	**0.63**
62_{rsd}	**0.65**	**0.58**	**0.58**	**0.58**	0.54	0.41	0.37	0.47	0.42	0.35	0.34	0.39
62_{rba}	0.62	0.54	0.56	0.54	0.54	0.46	0.46	0.48	0.46	0.42	0.48	0.4

TABLE 7. Comparison of confusion matrices.

Method	MLP (50 feat.)				LSTM (20 feat.)				GRU (30 feat.)			
		1	**2**	**3**		**1**	**2**	**3**		**1**	**2**	**3**
ANOVA	**1**	11	1	1	**1**	10	2	1	**1**	10	2	1
	2	3	1	2	**2**	3	2	1	**2**	1	5	0
	3	0	3	4	**3**	2	1	4	**3**	0	0	7
		1	**2**	**3**		**1**	**2**	**3**		**1**	**2**	**3**
Chi2	**1**	10	2	1	**1**	10	1	2	**1**	11	1	1
	2	2	2	2	**2**	0	3	3	**2**	4	0	2
	3	1	3	3	**3**	1	3	3	**3**	3	1	3
		1	**2**	**3**		**1**	**2**	**3**		**1**	**2**	**3**
Kruskal	**1**	8	1	4	**1**	13	0	0	**1**	13	0	0
	2	1	4	1	**2**	6	0	0	**2**	4	0	2
	3	1	0	6	**3**	7	0	0	**3**	5	0	2
		1	**2**	**3**		**1**	**2**	**3**		**1**	**2**	**3**
CMIM	**1**	12	1	0	**1**	6	3	4	**1**	9	3	1
	2	3	1	2	**2**	0	2	4	**2**	1	1	4
	3	1	2	4	**3**	1	1	5	**3**	3	1	3
		1	**2**	**3**		**1**	**2**	**3**		**1**	**2**	**3**
DISR	**1**	11	1	1	**1**	11	1	1	**1**	11	1	1
	2	2	1	3	**2**	3	1	2	**2**	3	1	2
	3	0	3	4	**3**	4	0	3	**3**	1	1	5
		1	**2**	**3**		**1**	**2**	**3**		**1**	**2**	**3**
MIFS	**1**	10	2	1	**1**	5	3	5	**1**	8	3	2
	2	2	1	3	**2**	0	4	2	**2**	1	2	3
	3	0	3	4	**3**	1	1	5	**3**	0	1	6
		1	**2**	**3**		**1**	**2**	**3**		**1**	**2**	**3**
ReliefF	**1**	11	1	1	**1**	7	4	2	**1**	8	2	3
	2	2	1	3	**2**	2	2	2	**2**	4	2	0
	3	5	0	2	**3**	4	0	3	**3**	5	0	2
		1	**2**	**3**		**1**	**2**	**3**		**1**	**2**	**3**
SURF	**1**	12	1	0	**1**	12	1	0	**1**	8	2	3
	2	4	1	1	**2**	5	1	0	**2**	5	1	0
	3	1	2	4	**3**	5	2	0	**3**	2	1	4
		1	**2**	**3**		**1**	**2**	**3**		**1**	**2**	**3**
SURF*	**1**	11	1	1	**1**	13	0	0	**1**	10	2	1
	2	4	1	1	**2**	6	0	0	**2**	4	1	1
	3	3	1	3	**3**	7	0	0	**3**	2	1	4
		1	**2**	**3**		**1**	**2**	**3**		**1**	**2**	**3**
MultiSURF	**1**	9	1	3	**1**	10	3	0	**1**	8	1	4
	2	2	1	3	**2**	2	4	0	**2**	1	1	4
	3	0	4	3	**3**	2	5	0	**3**	0	0	7

results are collected in separate folders. For best selection algorithms, the scikit-learn and scikit- rebate libraries are used. For deep network algorithms, we have used the torch library, which contains the LSTM and RNN algorithms, and we only extended them to include the proposed algorithms described in section D. We have used the Pytorch-lightening framework for training steps where the parameters and hyperparameters are set in the configuration file. For training, deep neural network algorithms run sequentially for each dataset. The learning rate was initially selected as $2e-2$, but it was reduced in the training steps after every 20 epochs.

To the best of our knowledge, this study is the first study that creates 114 different datasets from 17 skeletal keypoints by detecting and tracking samples. In addition, this study is the first to apply different feature selection algorithms to gait classification. Moreover, by providing different datasets and applying three different deep neural networks, we reached a classification F-score of 85%, which is promising for the proposed tiny dataset.

V. CONCLUSION

In this study, we have used a dataset collected at UCT Prague Hospital that includes a total of 84 measurements made during three exercises designed by the hospital's physicians to classify three classes of gait disorders. The presented method proceeded to perform feature extraction using the 10 best feature selection algorithms, as well as manual feature extraction, on observations of patients' gaits during these three exercises. The base features were extracted from the RCNN keypoint detector from the Torch library. Then, we applied feature extraction to add more meaningful features to the extracted base dataset. Furthermore, the base keypoint features of the tracking algorithm were also collected to investigate the impact of both types of extracted data. We have trained three different deep learning models to determine the best and most effective models. According to the results of this study, when the 30 features selected by the ANOVA algorithm were used to train the GRU network, the best results were achieved, with both the F-score and the classification accuracy reaching 85%. The results also show that the data prepared with the detector approach were more effective than the tracking data due to the nature of the designed clinical exercises. Furthermore, the best feature selection algorithms showed significant improvements in their F-scores compared with manual feature extraction.

A. FUTURE WORK

In future work, we intend to collaborate with hospitals and rehabilitation centres to collect more datasets, which will be used to evaluate a GRU trained with 30 features, as it was the best classifier in this study. Furthermore, to help physicians to make clinical decisions, the proposed best classification method will be used in smart home care systems. Further studies may focus on in-home data collection, which may improve the classification accuracy. Moreover, due to the temporal nature of the data captured, some methods, such as dynamic time wrapping, may be explored in order to further improve the classification accuracy. Finally, the investigation of the possibility of the proposed method including other movement disorders is another possible future study.

DECLARATIONS

This research was approved by the Ethics Committee of the University Hospital Královské Vinohrady Prague (EK-VP/4310120), where the measurements were made. Each patient signed an informed consent form that described the research conditions.

REFERENCES

[1] G. V. Kale and V. H. Patil, "A study of vision based human motion recognition and analysis," *Int. J. Ambient Comput. Intell.*, vol. 7, no. 2, pp. 75–92, Jul. 2016.

[2] M. R. Keyvanpour, S. Vahidian, and M. Ramezani, "HMR-vid: A comparative analytical survey on human motion recognition in video data," *Multimedia Tools Appl.*, vol. 79, nos. 43–44, pp. 31819–31863, Nov. 2020.

[3] L. Chen, H. Wei, and J. Ferryman, "A survey of human motion analysis using depth imagery," *Pattern Recognit. Lett.*, vol. 34, no. 15, pp. 1995–2006, Nov. 2013.

[4] S. Maudsley-Barton, J. McPhee, A. Bukowski, D. Leightley, and M. H. Yap, "A comparative study of the clinical use of motion analysis from Kinect skeleton data," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 2808–2813.

[5] M. Bennasar, Y. Hicks, and R. Setchi, "Feature selection using joint mutual information maximisation," *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8520–8532, Dec. 2015.

[6] G. Manikandan and S. Abirami, "An efficient feature selection framework based on information theory for high dimensional data," *Appl. Soft Comput.*, vol. 111, Nov. 2021, Art. no. 107729.

[7] U. Moorthy and U. D. Gandhi, "A novel optimal feature selection technique for medical data classification using ANOVA based whale optimization," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 3, pp. 3527–3538, Mar. 2021.

[8] H. Nasiri and S. A. Alavi, "A novel framework based on deep learning and ANOVA feature selection method for diagnosis of COVID-19 cases from chest X-ray images," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–11, Jan. 2022.

[9] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, "Feature selection using an improved chi-square for Arabic text classification," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 32, no. 2, pp. 225–231, Feb. 2020.

[10] M. Hussein and F. Özyurt, "A new technique for sentiment analysis system based on deep learning using chi-square feature selection methods," *Balkan J. Electr. Comput. Eng.*, vol. 9, no. 4, pp. 320–326, Oct. 2021.

[11] I. S. Thaseen, C. A. Kumar, and A. Ahmad, "Integrated intrusion detection model using chi-square feature selection and ensemble of classifiers," *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 3357–3368, Apr. 2019.

[12] F. Fleuret, "Fast binary feature selection with conditional mutual information," *J. Mach. Learn. Res.*, vol. 5, no. 9, pp. 1531–1555, 2004.

[13] P. A. Estevez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 189–201, Feb. 2009.

[14] C. Li, X. Luo, Y. Qi, Z. Gao, and X. Lin, "A new feature selection algorithm based on relevance, redundancy and complementarity," *Comput. Biol. Med.*, vol. 119, Apr. 2020, Art. no. 103667.

[15] P. E. Meyer and G. Bontempi, "On the use of variable complementarity for feature selection in cancer classification," in *Proc. Appl. Evol. Comput., EvoWorkshops, EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, EvoSTOC*, Apr. 2006, pp. 91–102.

[16] S. Ali Khan, A. Hussain, A. Basit, and S. Akram, "Kruskal–Wallis-based computationally efficient feature selection for face recognition," *Sci. World J.*, vol. 2014, pp. 1–6, Jun. 2014.

[17] R.-J. Palma-Mendoza, D. Rodriguez, and L. de-Marcos, "Distributed ReliefF-based feature selection in spark," *Knowl. Inf. Syst.*, vol. 57, no. 1, pp. 1–20, Oct. 2018.

[18] B. Zhang, Y. Li, and Z. Chai, "A novel random multi-subspace based ReliefF for feature selection," *Knowl.-Based Syst.*, vol. 252, Sep. 2022, Art. no. 109400.

[19] D. M. D. Raj and R. Mohanasundaram, "An efficient filter-based feature selection model to identify significant features from high-dimensional microarray data," *Arabian J. Sci. Eng.*, vol. 45, no. 4, pp. 2619–2630, Apr. 2020.

[20] R. J. Urbanowicz, R. S. Olson, P. Schmitt, M. Meeker, and J. H. Moore, "Benchmarking relief-based feature selection methods for bioinformatics data mining," *J. Biomed. Informat.*, vol. 85, pp. 168–188, Sep. 2018.

[21] L. di Biase, L. Raiano, M. L. Caminiti, P. M. Pecoraro, and V. Di Lazzaro, "Parkinson's disease wearable gait analysis: Kinematic and dynamic markers for diagnosis," *Sensors*, vol. 22, no. 22, p. 8773, Nov. 2022.

[22] M. K. Graham, J. P. Staab, C. M. Lohse, and D. L. McCaslin, "A comparison of dizziness handicap inventory scores by categories of vestibular diagnoses," *Otol. Neurotol.*, vol. 42, no. 1, pp. 129–136, 2021.

[23] K. Jun, D.-W. Lee, K. Lee, S. Lee, and M. S. Kim, "Feature extraction using an RNN autoencoder for skeleton-based abnormal gait recognition," *IEEE Access*, vol. 8, pp. 19196–19207, 2020.

[24] M. Yang, H. Zheng, H. Wang, and S. McClean, "Feature selection and construction for the discrimination of neurodegenerative diseases based on gait analysis," in *Proc. 3rd Int. Conf. Pervasive Comput. Technol. Healthcare*, Apr. 2009, pp. 1–7.

[25] R. Altilio, M. Paoloni, and M. Panella, "Selection of clinical features for pattern recognition applied to gait analysis," *Med. Biol. Eng. Comput.*, vol. 55, no. 4, pp. 685–695, Apr. 2017.

[26] K. Jun, Y. Lee, S. Lee, D.-W. Lee, and M. S. Kim, "Pathological gait classification using Kinect v2 and gated recurrent neural networks," *IEEE Access*, vol. 8, pp. 139881–139891, 2020.

[27] M. Eltoukhy, J. Oh, C. Kuenze, and J. Signorile, "Improved Kinect-based spatiotemporal and kinematic treadmill gait assessment," *Gait Posture*, vol. 51, pp. 77–83, Jan. 2017.

[28] M. N. I. Shuzan, M. E. H. Chowdhury, M. B. I. Reaz, A. Khandakar, F. F. Abir, M. A. A. Faisal, S. H. M. Ali, A. A. A. Bakar, M. H. Chowdhury, Z. B. Mahbub, M. M. Uddin, and M. Alhatou, "Machine learning-based classification of healthy and impaired gaits using 3D-GRF signals," *Biomed. Signal Process. Control*, vol. 81, Mar. 2023, Art. no. 104448.

[29] J. Kohout, J. Crha, K. Trnkova, K. Sticha, J. Mares, and M. Chovanec, "Robot-based image analysis for evaluating rehabilitation after brain surgery," *Mendel*, vol. 24, no. 1, pp. 159–164, Jun. 2018.

[30] M. Shayestegan, J. Kohout, K. Trnková, M. Chovanec, and J. Mareš, "Motion tracking in diagnosis: Gait disorders classification with a dual-head attentional transformer-LSTM," *Int. J. Comput. Intell. Syst.*, vol. 16, no. 1, p. 98, Jun. 2023.

[31] M. Shayestegan, J. Kohout, K. Štícha, and J. Mareš, "Advanced analysis of 3D Kinect data: Supervised classification of facial nerve function via parallel convolutional neural networks," *Appl. Sci.*, vol. 12, no. 12, p. 5902, Jun. 2022.

[32] J. W. Johnson, "Adapting mask-RCNN for automatic nucleus segmentation," 2018, *arXiv:1805.00500*.

[33] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[35] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7291–7299.

[36] Q. Li, R. Li, K. Ji, and W. Dai, "Kalman filter and its application," in *Proc. 8th Int. Conf. Intell. Netw. Intell. Syst. (ICINIS)*, Nov. 2015, pp. 74–77.

[37] M. Dalaison and R. Jolivet, "A Kalman filter time series analysis method for InSAR," *J. Geophys. Res., Solid Earth*, vol. 125, no. 7, pp. 1–21, Jul. 2020.

[38] V. Dentamaro, D. Impedovo, and G. Pirlo, "Gait analysis for early neurodegenerative diseases classification through the kinematic theory of rapid human movements," *IEEE Access*, vol. 8, pp. 193966–193980, 2020.

[39] G. V. Veres, L. Gordon, J. N. Carter, and M. S. Nixon, "What image information is important in silhouette-based gait recognition?" in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2004, p. II.

[40] I. Birch, M. Birch, and N. Asgeirsdottir, "The identification of individuals by observational gait analysis using closed circuit television footage: Comparing the ability and confidence of experienced and non-experienced analysts," *Sci. Justice*, vol. 60, no. 1, pp. 79–85, Jan. 2020.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[42] M. Bejani, D. Gharavian, and N. M. Charkari, "Audiovisual emotion recognition using ANOVA feature selection method and multi-classifier neural networks," *Neural Comput. Appl.*, vol. 24, no. 2, pp. 399–412, Feb. 2014.

[43] H. A. Mengash, L. Hussain, H. Mahgoub, A. Al-Qarafi, M. K. Nour, R. Marzouk, S. A. Qureshi, and A. M. Hilal, "Smart cities-based improving atmospheric particulate matters prediction using chi-square feature selection methods by employing machine learning techniques," *Appl. Artif. Intell.*, vol. 36, no. 1, Dec. 2022, Art. no. 2067647.

[44] H. N. Alshaer, M. A. Otair, L. Abualigah, M. Alshinwan, and A. M. Khasawneh, "Feature selection method using improved CHI square on Arabic text classifiers: Analysis and application," *Multimedia Tools Appl.*, vol. 80, no. 7, pp. 10373–10390, Mar. 2021.

[45] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462–472, Oct. 2017.

[46] T. Sueyoshi and S. Aoki, "A use of a nonparametric statistic for DEA frontier shift: The Kruskal and wallis rank test," *Omega*, vol. 29, no. 1, pp. 1–18, Feb. 2001.

[47] S. Iniyan and R. Jebakumar, "Mutual information feature selection (MIFS) based crop yield prediction on corn and soybean crops using multilayer stacked ensemble regression (MSER)," *Wireless Pers. Commun.*, vol. 126, no. 3, pp. 1935–1964, Oct. 2022.

[48] L. T. Vinh, S. Lee, Y.-T. Park, and B. J. d'Auriol, "A novel feature selection method based on normalized mutual information," *Appl. Intell.*, vol. 37, no. 1, pp. 100–120, Jul. 2012.

[49] N. Spolaôr, E. A. Cherman, M. C. Monard, and H. D. Lee, "Relieff for multi-label feature selection," in *Proc. Brazilian Conf. Intell. Syst.*, 2013, pp. 6–11.

[50] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, "Relief-based feature selection: Introduction and review," *J. Biomed. Informat.*, vol. 85, pp. 189–203, Sep. 2018.

[51] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Mach. Learn.*, vol. 53, nos. 1–2, pp. 23–69, 2003.

[52] T. T. Le, R. J. Urbanowicz, J. H. Moore, and B. A. McKinney, "Statistical inference relief (STIR) feature selection," *Bioinformatics*, vol. 35, no. 8, pp. 1358–1365, Apr. 2019.

[53] X. Cui, Y. Li, J. Fan, and T. Wang, "A novel filter feature selection algorithm based on relief," *Int. J. Speech Technol.*, vol. 52, no. 5, pp. 5063–5081, Mar. 2022.

[54] F. A. del Campo, M. C. G. Neri, O. O. V. Villegas, V. G. C. Sánchez, H. D. J. O. Domínguez, and V. G. Jiménez, "Auto-adaptive multilayer perceptron for univariate time series classification," *Expert Syst. Appl.*, vol. 181, Nov. 2021, Art. no. 115147.

[55] S. Gao, Y. Huang, S. Zhang, J. Han, G. Wang, M. Zhang, and Q. Lin, "Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation," *J. Hydrol.*, vol. 589, Oct. 2020, Art. no. 125188.

[56] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[57] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.

MOHSEN SHAYESTEGAN received the M.S. degree in control and automation engineering from University Pura Malaysia (UPM), in 2013, and the Ph.D. degree in electrical, electronic, and systems engineering from National University Malaysia (UKM), in 2018. He is currently a full-time Senior Researcher with the University of Pardubice, Czech Republic. His current research interests include artificial intelligence (AI), deep learning (DL), machine learning, robotics, image processing, and computer vision (CV).

JAN KOHOUT received the M.Sc. degree from the Faculty of Electrical Engineering, Czech Technical University, and the Ph.D. degree in technical cybernetics from the University of Chemistry and Technology Prague, Prague, Czech Republic. He is currently an Assistant Professor in signal and image processing with the Department of Computing and Control Engineering, University of Chemistry and Technology Prague. His research interest includes biomedical data pre-processing.

LUDMILA VEREŠPEJOVÁ is currently pursuing the Ph.D. degree with the Department of Otorhinolaryngology, Third Faculty of Medicine, Charles University and University Hospital Kralovske Vinohrady. She is a postgraduate student of biomedicine in the field of experimental surgery. She is focused on the development of new diagnostic methods that enable clinicians to objectively evaluate facial nerve mimetic function.

MARTIN CHOVANEC received the M.D. degree from Charles University, Prague, in 2002, the Ph.D. degree, in 2006, and the Habilitation degree in the field of otorhinolaryngology and head and neck surgery, in 2016. Since 2015, he has been the Head of the Department of Otorhinolaryngology, Third Faculty of Medicine, Charles University and University Hospital Kralovske Vinohrady. His expertise is in skull base surgery, otological and neurotological surgery, and head and neck oncology.

JAN MAREŠ received the M.Sc. and Ph.D. degrees in technical cybernetics from the University of Pardubice, Czech Republic. He is currently an Associate Professor in signal and image processing with the Department of Computing and Control Engineering, University of Chemistry and Technology Prague, Prague. His research interest includes biomedical data processing for early diagnosis.