IMU Based Context Detection of Changes in the Terrain Topography

Taylor Knuth
Northrop Grumman Corporation and
University College London
London, United Kingdom
ucestkn@ucl.ac.uk

Paul Groves
Department of CEGE
Univeristy College London
London, United Kingdom
p.groves@ucl.ac.uk

***Abstract*—This paper introduces an IMU based context machine learning algorithm for terrain topography classification. Four different terrains are considered: concrete, pebble, sand, and grass. The grass terrain is further split into two separate classes based off moisture content of the grass, wet and dry. Separate terrain topography datasets are created by walking on different terrains and logging the data. The subject has been equipped with an IMU attached on the surface of the shoe above the toes. Data is collected and stored via a Bluetooth smartphone controller over multiple recording sessions. Acceleration, angular rate, and magnetic field were recorded. The recorded data is extracted in two second sliding window intervals, whereupon the magnitude of the sensor outputs, in three dimensions, is calculated. A low-pass band filter is also applied to the magnitude for the acceleration, angular rate, and magnetic field data. The magnitude output is processed in the time domain to calculate variance, energy, kurtosis, range, skewness, and the zero-crossing rate. The magnitude data is converted into the frequency domain and the peak magnitude and its corresponding frequency in the sliding window are determined. A set of 44 features is extracted from each window and then tested and trained to classify terrain topography using five different machine learning methods: Artificial Neural Network, Decision Tree, k-Nearest Neighbor, Naïve-Bayes, and Support Vector Machine. The 44-feature set is optimized using a wrapper selection algorithm for the Decision Tree and k-Nearest Neighbor algorithms. The results show that by utilizing sensor data from an IMU in combination with machine learning methods a terrain topography classification algorithm can accurately predict various terrains over which the user traverses.**

Keywords—context detection, terrain topography, machine learning, feature optimization, terrain classification

I. INTRODUCTION

Context detection using microelectromechanical systems (MEMS) inertial measurement units (IMUs) can be used to determine whether a person is walking, standing, running, or in a vehicle [1]. Data collected from MEMS IMUs has also been used in the identification of daily activities, such as lying down, standing up [2], brushing teeth, vacuuming, and daily exercise routines [3]. Beyond identifying daily routines of individuals there are various uses in sports analytics to track player movement and health [4].

The many uses of context detection across a variety of fields are well known, but reliable and accurate context detection is necessary for correct navigation and positioning algorithms. Context detection is a major component of Pedestrian Dead Reckoning (PDR) [5] and Zero Velocity Update (ZVU) algorithms [6]. PDR and ZVU rely on context detection to determine when a step is made, when the stance phase occurs, or the period where zero velocity is experienced. The research in this paper relies on context detection to classify differences in terrain topography.

Terrain topography comes from the study of topography, which is the study of forms and features of land surfaces, including the analysis of all the features of the Earth's surface, including human-made features [7]. This includes the analysis of the Earth's surface as it refers to the differences in terrains experienced by individuals as they interact with natural and human-made features [8]. An investigation of the interaction between these features in terrain topography with context detection is presented as a method to detect changes in various classes of terrain.

Terrain topography identification can improve ZVU and PDR algorithms by understanding how the pedestrian gait cycle varies on different terrains. As the gait varies on different terrains, the length of the stance phase varies and the detection of a step for a zero velocity interval is affected [9]. Context detection of terrain topography can also be used in map-matching position techniques. With a known initial position and heading, a real time terrain classification algorithm could identify transitions in terrain and match them to known terrain changes from a terrain database to calibrate the position of the user. There are existing efforts to identify various terrain topography to improve navigation techniques relating to autonomous robots. Using force and torque sensors with visual imagery different terrains are classified then compared against a global map of terrain irregularities to determine position [10]. Identifying changes in terrain topography or different types of terrains with foot-mounted inertial sensors can improve navigation and positioning solutions by matching the terrain information to a topographic map of where current navigation solutions are unreliable due to degraded global navigation satellite system (GNSS) signal or improve PDR and ZVU algorithms using context detection to determine the influence of terrains on the algorithms.

Current methods of terrain topography classification rely on visual and acoustic analysis. Aerial image data for terrain classification is a traditional camera-based visual approach. The aerial image is analyzed to distinguish different terrain types through methods such as automatic texture measure. Aerial

images are required because a large area was required to group areas into terrain types [11]. The aerial based approach presents restrictions because residential areas are simply classified as urban or suburban without any distinction of the actual terrain. Another method of terrain classification requires soil classification using geographic object-based image analysis (GEOBIA). GEOBIA partitions remote sensing imagery or digital elevation models into homogeneous image objects based on image segmentation [12]. This study was limited to soil types and not inclusive of various types of terrain topography with only 58% accuracy in correctly classifying the soil type. Another visual based classification method fused color information method from 3D LiDAR scans to scan the area in front of the robot at a high frame rate to extract features for terrain analysis [13]. Acoustical analysis for terrain classification is used by analyzing vibrations generated by autonomous ground vehicles [14]. Unmanned vehicles are modeled as a vibrating system [15] and using irregular surfaces of the terrain and time-varying vertical displacements of the chassis, a vibration of the frequency response for each terrain serves as a signature for classification.

Many of these approaches are effective in terrain classification, but there are constraints when applying this to the navigation field. There are limitations in terms of accuracy, cost, and size of the systems. The visual based systems are affected by lighting of the terrain and oversimplification of terrain type while the acoustic analysis requires a vibration analysis from the chassis and frequency response. A shift to more cost-effective methods for terrain classification using the analysis of human gait as it changes from effects of the various terrains, has taken place over the last few years. Human gait analysis has relied on inertial data to improve context detection, but the field of terrain classification using inertial data is new.

There are limited approaches in using inertial data to capture and classify terrain topography. In the study by Hashmi, Riaz, Hussain, and Shahzad they identified whether a subject was indoors or outdoors, and on a hard or soft surface [16]. The sensors used for the study were two six degree of freedom (DOF) IMUs. The IMUs were two Android smart phones with a triaxial accelerometer and triaxial gyroscope. One phone was attached to the chest and the other sensor was attached to the lower back. The placement of the sensors is a limitation because previous studies have shown more accurate results in IMU detection for gait analysis when the IMU is attached to the foot [17]. Another limitation of the study is the reliance on 194 features in the Support Vector Machine (SVM) algorithm, the 194 features lead to overfitting of the model. When classifying the terrain into two classes, the accuracy varied from 77.5 to 97.5%, but when identifying more than two classes the results varied from 69.7 to 89.1% [16].

Another study that used an IMU to classify terrain focused on the level of the ground surface, whether it was flat or uneven. When considering whether the surface was even or uneven the predictive algorithm was 95.1 and 97.6% correct, respectively [18]. The detection of the surface conditions was limited to laboratory conditions with a safety harness attached to the participant to eliminate any fall risk while walking on uneven surfaces. The 9 DOF IMU sensor was attached to the hip. Performing the experiments under laboratory conditions with safety harnesses and an IMU attached to the posterior surface of the torso impacts the natural gait of the participant.

The ability to identify whether someone is indoors or outdoors, on a hard or soft surface, or on level ground is not limited to an IMU. Humans are aware of their surroundings while walking and performing daily tasks and can classify the terrain through reliance on sensory awareness and vibrotactile messaging [19]. This ability to use sensory awareness, to differentiate natural or human-made terrains is the basis of the assumption that with the benefit of data from an IMU an algorithm can be created using context detection in association with machine learning to discern terrain topographies.

II. Proposed Method and Dataset

To utilize context detection for terrain topography classification, datasets for different terrain styles or classes of terrains need to be collected. Datasets of separate terrains were created by collecting data via a MEMs IMU attached to a pedestrian user. IMUs are used for context detection in a variety of uses, and due to their small size, they can be attached to a user's foot to collect data across various terrains. For terrain topography classification the IMU is attached to the foot of the user. This is because the gait of a pedestrian changes as they transition to differing terrains [20] and attaching the IMU to the foot allows for greater sensitivity when recording the inertial data. To reduce impact on pedestrian natural gait an IMU was placed on top of the right shoe, as shown in Fig. 1. Attaching the IMU to the surface of the shoe, above the toes, provides the most consistent step and zero-velocity interval detection [21] [22] when compared to placement on the ankle or heel. Attaching the IMU to the ankle requires a support structure that limits the motion of the foot and when the IMU is attached to the heel, the straps used to secure the IMU typically wrap under the arch of the foot and affect the natural gait.

Fig. 1. IMU placement on right foot

To collect the data for terrain profiles, a 9 DOF IMU from Inertial Elements is used [23]. The IMU is low-cost, compact and allows the user to collect data via Bluetooth to a mobile phone using the mobile app, OsmiumScope. To further reduce the impact of an IMU's placement on the natural gait of the user, the IMU is attached using Velcro®. Fig. 2 is an image of the IMU attached above the toe of the shoe with Velcro®. By attaching the IMU with Velcro®, there are no straps under the arch of the foot to alter each footstep or any casing at the heel that changes the gait by strapping the IMU to the ankle. The Velcro® strips are adhesive and cover the entire base of the IMU housing, thus a larger contact area from the IMU reduces the likelihood of the IMU shifting during data collection and

affecting the inertial measurements. The 9 DOF IMU includes a triaxial accelerometer, triaxial gyroscope, and triaxial magnetometer. The collected data on the mobile phone controller was exported to MS Excel and Matlab R2018b for analysis.

Fig. 2. Inertial Elements IMU attached to the right foot with Velcro®

Initial testing began with three terrains: concrete, sand, and pebble. These terrains were chosen due to ease of accessibility and variation of terrain style. Data was collected using the IMU attached to a NIKE® athletic trainer. A single user and pair of athletic trainers were used to collect data, thus eliminating variation in pedestrian gait and focusing on variations in terrain topography. Data was collected at the Royal Air Force Mildenhall (RAFM) base in England (Fig. 3) on a concrete running track (Fig. 4) to create the concrete profile. For the pebble terrain, the southern end of Chesil Beach in Portland, England, was used to create the pebble profile. And data for the sand terrain beach profile was collected at Weymouth Beach in Weymouth, England. Fig. 5 shows the location of the two beaches used for data collection of the pebble and sand terrains.

Fig. 3. RAF Mildenhall location in Suffolk, England

Fig. 4. Satellite imager of concrete test track at RAF Mildenhall

Fig. 5. Locations of the pebble beach (Chesil Beach) and sandy beach (Weymouth Beach) within England

The pebble and sand beaches are defined using the modified Wentworth Scale [24][25], based on the grain size of the two beaches. Chesil Beach is classified as a pebble beach according to the modified Wentworth Scale as the diameter of the aggregate ranges from 30 to 200 mm. Weymouth beach is classified as very fine to fine sand by the Wentworth Scale because the grain size varies from 0.0625 to 0.25 mm in diameter. Each beach and their respective grain sizes are shown in Fig. 6 and Fig. 7. Data was collected on each terrain in two sessions, one session for training and one session for testing.

Fig. 6. (a) Chesil Beach (b) pebble grains at Chesil Beach

Fig. 7. (a) Weymouth Beach (b) fine sand grains at Weymouth Beach

III. Results

A. Feature Extraction

Prior to feature extraction the raw sensor samples are divided into datasets with a predetermined length of time, or a window length, to generate features for analysis. The window length includes all the raw sensor data over a certain amount of time. A two second sliding window with 50% overlap is used for training and testing. An example of consecutive sliding windows over a 10 second period with the accelerometer output data in three dimensions is shown in Fig. 8. Using 50% overlap allows for each sample to be processed in two windows without imbalance, ensuring that the data is sampled twice with no missing data.

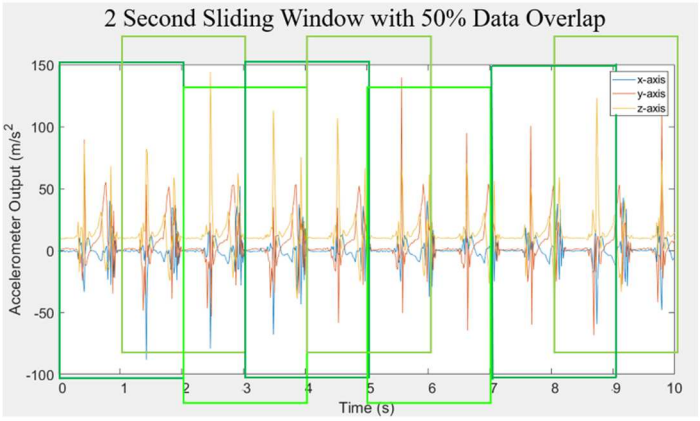

Fig. 8. Example of sliding window overlap with a two second sliding widow, allowing for a 50% data overlap

To reduce the effects of orientation changes in the model, the magnitude of each sensor type is calculated using the three dimensions of x, y, and z and performed in MS Excel and Matlab R2018b (1). Additionally, the mean of the magnitude is removed from each window to eliminate non-zero mean prior to converting the time domain to the frequency domain. A low-pass band filter is applied to the magnitude of each of the three collection sensor types: accelerometers, gyroscopes, and magnetometers. The low-pass band filter (LPF) of the sensors' magnitude data is an additional extracted feature for input to help identify patterns in the time domain by cutting off frequencies higher than the sampling rate of the IMU of 62.5 Hz.

$$magnitude = \sqrt{{f_x}^2 + {f_y}^2 + {f_z}^2} \quad (1)$$

Where f_x, f_y, and f_z are the sensor outputs.

The features from each window are extracted for training and classification. Feature measurements provide descriptions of patterns from which differences between the terrains can be discerned. Features were extracted in both time and frequency domains. Features in the time domain are based off variations in motion during each two second sliding window, whereas features in the frequency domain are based off the periodic changes in motion during each sliding window.

The time domain features used for training and classification are variance (2), energy (3), kurtosis (4), range (5), skewness (6), and the zero-crossing rate (ZCR) (7). Each of these equations is calculated for each two second overlapping sliding window using the signal inputs from the magnitude of each sensor type and its corresponding low-pass filter. This creates a set of six features for every equation to be extracted from each sliding window. However, the ZCR only uses the input signal from the accelerometer magnitude and LPF accelerometer, creating two additional features. These six features and the number of signal inputs were chosen for analysis in the time domain based off the effectiveness of behavior classification in previous studies [26][27][28].

$$\sigma = \sqrt{E\{(x-\mu)^2\}} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})^2} \quad (2)$$

$$energy = \sum_{n=1}^{N} \|x_n\|^2 \quad (3)$$

$$kurtosis = \frac{E\{(x-\mu)^4\}}{\sigma^4} = \frac{1}{N\sigma^4}\sum_{n=1}^{N}(x_n - \bar{x})^4 \quad (4)$$

$$range = max\{x\} - min\{x\} \quad (5)$$

$$skewness = \frac{E\{(x-\mu)^3\}}{\sigma^3} = \frac{1}{N\sigma^3}\sum_{n=1}^{N}(x_n - \bar{x})^3 \quad (6)$$

$$ZCR = \frac{1}{N-1}\sum_{n=1}^{N-1} \text{II}\{(x_n - \bar{x})(x_{n+1} - \bar{x}) < 0\} \quad (7)$$

Where σ is the variance, N is the number of samples in the sliding window, μ is the mean, x_n is the n-th epoch of data for the window, the indicator function II is 1 if the argument is true elsewise it is 0.

The frequency domain features used for training and classification are the peak magnitude of the signal input (8) for each sensor in the sliding window and its corresponding frequency. A total of 12 features are extracted in the frequency domain from the three sensor magnitude inputs and their LPF inputs with six features being extracted from the peak magnitude and six features from the frequency for each sliding window. The frequency domain features are obtained by converting the time domain into the frequency domain using a Fast Fourier Transform (FFT) calculated with Matlab R2018b . The output of the FFT is a sequence of coefficients representing amplitudes of frequency components of the input signal and the distribution of the signal energy The frequency domain features were chosen based off the effectiveness of previous studies [5][29][30].

$$Peak_{Magnitude} = max\|x_n\| \quad (8)$$

Where x_n is the n-th epoch of data for the window.

In total, 44 features are extracted for every sliding window using the magnitude of the three sensor types and the magnitude

of the three sensor types after being post-processed through an LPF. Table 1 provides a description of each feature number, type, and in which domain the feature is found.

TABLE I. TERRAIN EXTRACTED FEATURE SET FOR A TWO SECOND SLIDING WINDOW

Number	Feature	Domain
F1	Variance – Accelerometer	Time
F2	Variance – LPF Accelerometer	Time
F3	Variance – Gyroscope	Time
F4	Variance – LPF Gyroscope	Time
F5	Variance – Magnetometer	Time
F6	Variance – LPF Magnetometer	Time
F7	Energy – Accelerometer	Time
F8	Energy – LPF Accelerometer	Time
F9	Energy – Gyroscope	Time
F10	Energy – LPF Gyroscope	Time
F11	Energy – Magnetometer	Time
F12	Energy – LPF Magnetometer	Time
F13	Kurtosis – Accelerometer	Time
F14	Kurtosis – LPF Accelerometer	Time
F15	Kurtosis – Gyroscope	Time
F16	Kurtosis – LPF Gyroscope	Time
F17	Kurtosis – Magnetometer	Time
F18	Kurtosis – LPF Magnetometer	Time
F19	Range - Accelerometer	Time
F20	Range – LPF Accelertometer	Time
F21	Range – Gyroscope	Time
F22	Range – LPF Gyroscope	Time
F23	Range – Magnetometer	Time
F24	Range – LPF Magnetometer	Time
F25	Skewness – Accelerometer	Time
F26	Skewness – LPF Accelerometer	Time
F27	Skewness – Gyroscope	Time
F28	Skewness – LPF Gyroscope	Time
F29	Skewness – Magnetometer	Time
F30	Skewness – LPF Magnetometer	Time
F31	ZCR – Accelerometer	Time
F32	ZCR – LPF Accelerometer	Time
F33	Peak Magnitude – Accelerometer	Frequency
F34	Peak Magnitude – LPF Accelerometer	Frequency
F35	Peak Magnitude – Gyroscope	Frequency
F36	Peak Magnitude – LPF Gyroscope	Frequency
F37	Peak Magnitude – Magnetometer	Frequency
F38	Peak Magnitude – LPF Magnetometer	Frequency
F39	Max Frequency – Accelerometer	Frequency
F40	Max Frequency – LPF Accelerometer	Frequency
F41	Max Frequency – Gyroscope	Frequency
F42	Max Frequency – LPF Gyroscope	Frequency
F43	Max Frequency – Magnetometer	Frequency
F44	Max Frequency – LPF Magnetometer	Frequency

An initial test of the training data was used to analyze if different machine learning algorithms could accurately detect changes in the terrain topography. Five algorithms: Artificial Neural Network (ANN); Decision Tree (DT); k-Nearest Neighbor (kNN); Naïve-Bayes (NB); and Support Vector Machine (SVM) were chosen to use the training data to create an initial classification model. These five machine learning algorithms were chosen because of their capability for context and behavioral detection [30][31].

ANN machine learning uses neurons, or simple processing elements, connected to form a network that mimics a biological neural network [32]. The ANN repeatedly applies the training data and automatically adjusts the neuron parameters. A single neuron is not sufficient to solve complex classification because it does not contain enough adaptive parameters for nonlinearities. To account for the nonlinearities, multiple layers are linked into a network. Training data forms the input layer, and the result is the output layer. Between the input and output layers is the hidden layer. The hidden layer identifies patterns and transforms the input from one or more input layers before assigning a weight to the inputs and processing an output. A two-layer example of a neural network is shown in Fig. 9.

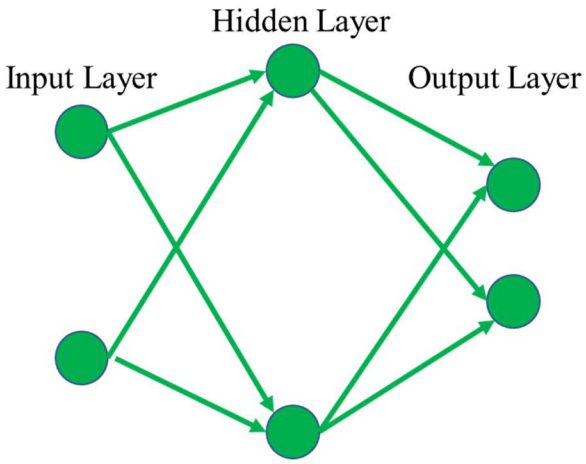

Fig. 9. Example structure of a two-layer artificial neural network

The DT machine learning method recursively partitions the input to reach a decision [33]. With the training data features as the input, it is then split into branch segments. The branches are

split at nodes based off similarities in the data. The root is the starting point of the tree and nodes without any further descendants are terminals. The data is classified navigating from the root to the terminals. Along the path, the internal nodes split the data into two or more segments according to the decision criteria until all samples at a node belong to the same class. A simple decision tree with two features is shown in Fig. 10.

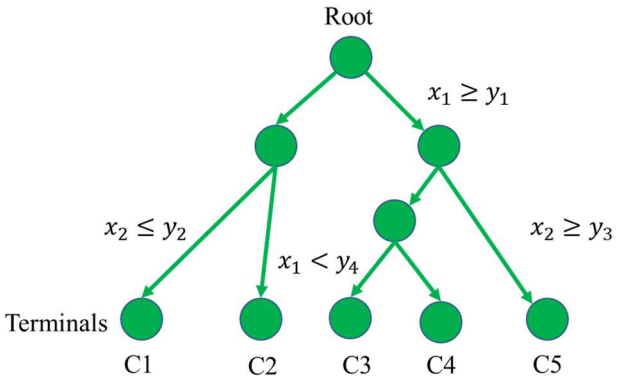

Fig. 10. Example of a two feature decision tree where the results are classified into five classes ranging from C1 to C5

The kNN classification method provides output as a class membership. Each object is classified by plurality based off its distance to the nearest set of neighbors [34]. The distance is approximated between the k-nearest neighbors to save on computational load until the final distance is calculated at the end with the objects on the edge of each classification group. kNN relies on a physical distance for classification from each object and the object is assigned to the class with the most common neighbors. If there is only one nearest neighbor, k=1, then the object is assigned to the class of its single nearest neighbor. Weights can be assigned to the nearest neighbor classes to contribute more to the average rather than relying on distant outliers [35]. A common method is giving each neighbor a weight of *1/d*, where *d* is the distance to the neighbor. A visual example of kNN classification for two classes can be found in Fig. 11.

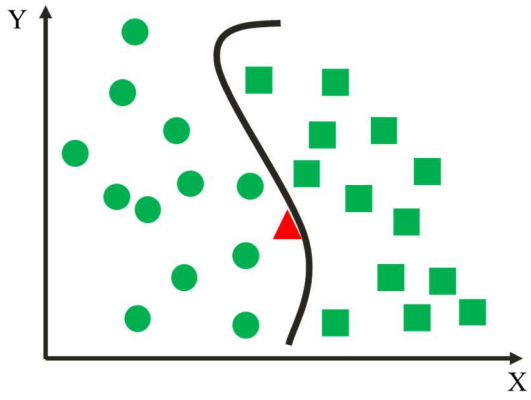

Fig. 11. Example of kNN classification of a red triangle based off distance to the two seperate classes – square and circle.

NB constructs classifiers by assigning class labels to problem instances. The problem instances are represented as vectors of feature values, where the class labels are drawn from finite data sets. The NB classifier assumes that any value of a particular feature is independent from any other feature when given the class variable. Parameter estimation for NB uses the method of maximum likelihood. NB methods require small amounts of training data to estimate the classification parameters [36]. The limitation of NB is that because it assumes the variables to be independent it does not match reality. Many variables are not independent of each other. NB also requires a large processing time because each attribute is given the same priority.

The final classification method, SVM, does not depend on prior probabilities and can be trained with smaller sample sets. SVM is best suited for binary classification. SVM relies on statistical learning theory and kernel-based methods for classification [37]. The SVM learning classifier is constructed to find the optimum hyperplane in the feature space so the margin between the two classes is maximized and the error is minimized (Fig. 12). The training data with distinct labels are separated on either side of the hyperplane and the distances of the hyperplane to the nearest training point are maximized. This distance is the optimal margin and the samples on the margin are support vectors. As SVM is a primary binary classifier, increasing the number of classes decreases the accuracy of the algorithm.

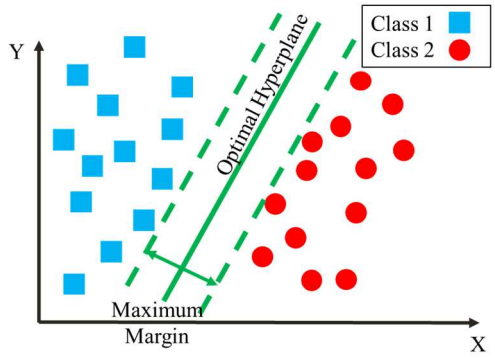

Fig. 12. SVM classifier example of two classes with optimal hyperplane

Using the 44 features as input, each of the five machine learning methods was trained and tested using 10-fold cross validation to develop a model that can determine the accuracy in predicting the three terrain classes of concrete, pebble, and sand. 10-fold cross validation divides the dataset randomly into 10 equally sized data subsets or folds. During the machine learning process, 9 randomly selected folds are used as training sets and the final tenth fold is used as the test set. This procedure is repeated 10 times to guarantee that all samples are used equally in testing and testing to maintain independence of the training and testing data for model learning.

After training and testing the model with the 44 extracted features, the model is analyzed statistically. The accuracy of the five models is compared by calculating the percentage of correctly identified instances divided by the total number of instances (9).

$$A = \frac{T_C}{T_I} \tag{9}$$

Where A is the model performance accuracy, T_C is the number of correctly identified instances, and T_I is the total number of instances.

The second evaluation metric of the five different machine learning methods is accomplished by calculating the F_1 score. The F_1 evaluation metric measures a model's accuracy by combining the precision and recall scores of the model. The F_1 score computes how many times a model predicted the correct instance, across the entire dataset [38]. Precision, P, is the number of results correctly attributed to a class divided by the total number attributed to said class (10). Recall, R, is the number of results accurately attributed to the class divided by the number that truly belong in the class (11). The accuracy of the classification is calculated with the F_1 score (12), which is the harmonic mean of P and R.

$$P = \frac{T_p}{T_p + F_p} \tag{10}$$

$$R = \frac{T_P}{T_p + F_n} \tag{11}$$

$$\mathrm{F}_1 = 2\left(\frac{PR}{P + R}\right) \tag{12}$$

Where T_p is the number of true positives, F_n is the number of false negatives, and F_p is the number of false positives.

Comparing the initial results from the five machine learning methods indicate that using an ANN based classification algorithm yielded the most accurate prediction and highest F_1 score. Table 2 shows the model accuracy, precision, recall, and F_1 scores for all five machine learning methods.

TABLE II. MACHINE LEARNING METHOD COMPARISONS OF MODEL ACCURACY FOR A THREE-TERRAIN TOPOGRAPHY CLASSIFICATION USING 44 INPUT FEATURES

Algorithm	Model Accuracy (%)	Precision (%)	Recall (%)	F_1 Score (%)
Artificial Neural Network (ANN)	99.35	99.53	99.82	99.67
Decision Tree (DT)	99.31	99.66	99.66	99.66
k-Nearest Neighbor (kNN)	98.79	99.49	99.29	99.39
Naïve-Bayes (NB)	96.98	97.77	99.17	98.47
Support Vector Machine (SVM)	63.93	63.93	100	78.00

B. Additional Terrain Classification

A terrain topography classification tool was created for three terrain styles of concrete, pebble, and sand. The tool has a model accuracy with the top three performing methods varying from 98.79 to 99.35% in distinguishing between the three terrain types. The terrain topography classification tool is only as powerful as the number of terrain classes that are identifiable using inertial data inputs and the classification algorithm. As the inertial data is collected via the IMU attached to the user's foot, additional terrain styles need to be recorded and data profiles created.

When considering additional terrain styles, pedestrian usage was considered. As urban centers are reclaiming green space around the world, many pedestrians walk on grass terrain daily. The city of London boasts 3,000 parks of varying sizes that are designated as public open spaces. These green spaces cover almost 18% of London, more than the area of the city covered by roads and railways combined [39]. A robust terrain topography classification algorithm needs to classify grassy terrains.

Data for the grass terrain was collected in the same manner as the concrete, pebble, and sand terrains. The IMU was attached to the right foot of the user and controlled via Bluetooth. The recording session was conducted at RAFM on a grass field. Data was initially collected in two separate sessions, one for training data and the second session for testing data.

After data collection, the sensor output was post-processed for feature extraction. The same 44 features, as previously mentioned in Table 1, were extracted for each two second sliding window. By using the same features, the initial classification algorithm is the foundation and can be added to. The extracted features from the four terrains were used as input to train and test the machine learning methods. The same five algorithms were used as the foundation and trained and tested using 10-fold cross validation. The comparative results of the four-terrain topography classification are shown in Table 3.

TABLE III. MACHINE LEARNING METHOD COMPARISONS OF MODEL ACCURACY FOR A FOUR-TERRAIN TOPOGRAPHY CLASSIFICATION USING 44 INPUT FEATURES

Algorithm	Model Accuracy (%)	Precision (%)	Recall (%)	F_1 Score (%)
Artificial Neural Network (ANN)	99.40	99.83	99.63	99.73
Decision Tree (DT)	99.30	99.63	99.67	99.65
k-Nearest Neighbor (kNN)	98.65	99.54	99.09	99.32
Naïve-Bayes (NB)	97.57	98.55	98.99	98.77
Support Vector Machine (SVM)	43.14	63.93	57.02	60.28

As before when classifying three terrains the ANN, DT, and kNN algorithms were the most accurate methods. Introducing a fourth terrain class reduced the accuracy in the SVM algorithm, but overall, there are no major differences in the model performance when adding a fourth class to the classifier.

When classifying the fourth terrain of grass, a fifth terrain was considered by splitting the grass terrain into two, dependent on the moisture content of the grass. Previous studies and results have shown that the strength and stiffness of soils vary based on the water and moisture content in the soil [40][41]. This variation in soil strength is perceivable with sensory awareness and was considered during data collection. Testing of this sensory awareness using the IMU was completed by completing

two separate recording sessions on days with varying levels of moisture content on the grass field at RAFM. The first collection session was taken during the European drought of 2022, amid the driest summer on record in Suffolk [42], where RAF Mildenhall is located. The second collection session was taken a couple of months later after precipitation had returned to normal levels, and the session began immediately following a rainstorm. The dataset from the first session is labelled as dry grass and the dataset from the second session is labelled as wet grass. A visual comparison of the same field is shown on the two test dates in Fig. 13. The two labelled datasets split the grass terrain into two classes.

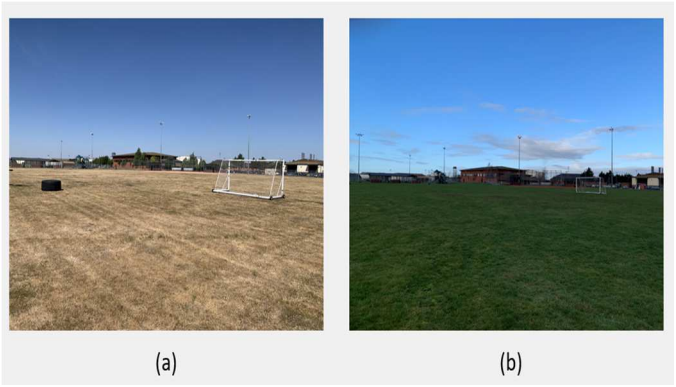

Fig. 13. Test field at RAF Mildenhall with varying moisture content. (a) Dry grass (b) Wet grass

Using the same feature extraction and testing and training methods for the three and four terrain classification a new algorithm is compared with the five machine learning methods. The comparison of model accuracy, precision, recall and F_1 scores is found in Table 4, found below.

TABLE IV. MACHINE LEARNING METHOD COMPARISONS OF MODEL ACCURACY FOR A FIVE-TERRAIN TOPOGRAPHY CLASSIFICATION USING 44 INPUT FEATURES

Algorithm	Model Accuracy (%)	Precision (%)	Recall (%)	F_1 Score (%)
Artificial Neural Network (ANN)	91.30	95.36	95.42	95.45
Decision Tree (DT)	85.77	92.35	92.33	92.34
k-Nearest Neighbor (kNN)	83.52	90.06	93.18	91.59
Naïve-Bayes (NB)	76.89	84.60	89.41	86.94
Support Vector Machine (SVM)	43.29	64.16	57.09	60.42

Further analysis is only performed on the ANN and DT algorithms, because the remaining three algorithms accuracy is less than 85% accurate. Although the ANN algorithm is 91.3% accurate with an F_1 score of 95.5% and the DT is 85.8% accurate with and F_1 of 92.3%, when analyzing the confusion matrix, the inaccuracies in the classification algorithm are attributed to misclassifications between the dry and wet grass. The confusion matrices for the ANN classification (Fig. 14) and the DT classification (Fig. 15) highlight how the separated dry and wet grass terrain is misidentified at a much higher percentage than the overall model accuracy. The model accuracy for the ANN algorithm for wet and dry grass terrains is 85.6%. The model accuracy for the DT algorithm for wet and dry grass is 79.2%. The decrease in model accuracy of the split grass terrain indicates that the algorithm cannot repeatedly distinguish the differences between wet and dry grass terrain. Whereas the rest of the terrains can be identified.

Actual Terrain	Predicted Terrain				
	Sand	**Pebble**	**Concrete**	**Wet**	**Dry**
Sand	**1037**	30	5	29	50
Pebble	39	**731**	6	49	18
Concrete	2	0	**3534**	0	0
Wet	24	23	0	**1179**	177
Dry	49	17	4	191	**1000**

Fig. 14. Confusion matrix of a five terrain classification problem using the ANN machine learning method

Actual Terrain	Predicted Terrain				
	Sand	**Pebble**	**Concrete**	**Wet**	**Dry**
Sand	**897**	71	0	87	96
Pebble	76	**695**	0	42	30
Concrete	0	0	**3534**	0	0
Wet	101	42	0	**1004**	256
Dry	94	28	0	243	**896**

Fig. 15. Confusion matrix of a five terrain classification problem using the DT machine learning method

The results from the five-class terrain classification algorithm indicate that the classification algorithm is terrain dependent and not reliable for moisture content detection. Thus, adding the fourth and fifth moisture content classes of the grass terrain is not viable for classification. In future work of matching terrains to a map database, only the terrain class will matter because the wet or dry grass is still a type of grass.

C. Feature Optimization

In machine learning, relevant features can optimize the results and reduce computational load of the algorithm. The extracted features can be classified by their relevancy with three qualifiers: irrelevant, weakly relevant, and strongly relevant [43]. The initial results from the three and four terrain classifiers indicate that the three best performing algorithms are ANN, DT, and kNN. To find the features that are strongly relevant a feature selection algorithm is used to compare features and eliminate redundant or irrelevant data from the feature list.

By utilizing a wrapper selection algorithm, the extracted features can be optimized to improve the model accuracy and reduce computation time [44]. The training for a wrapper selection algorithm requires significant computation time but is used to detect interactions between variables [45]. The wrapper

algorithm used for this research utilizes the best first attribute selection search function in a forward direction and terminates the search five attributes beyond the local maximum.

The wrapper selection algorithm takes the set of 44 features and then wraps the selected machine learning method through feature subsets. The first feature is calculated by running a machine learning algorithm to determine the single feature with the highest correlation to the dependent terrain variable. The first feature is then placed into a feature subset. The second feature is then calculated by using the selected machine learning algorithm with the first selected feature to determine the next feature with the second highest correlation to the dependent terrain variable. The second feature is then placed into the feature subset and the process is repeated until the output performance no longer improves for five consecutive iterations beyond the local maximum. Including the termination criterion of five, forces the wrapper selection algorithm to select the best subset of features by checking algorithm performance beyond the local maximum. The optimized feature subset consists of all the features that improve the selected learning algorithm performance. Fig. 16 is a visual representation of the wrapper selection algorithm.

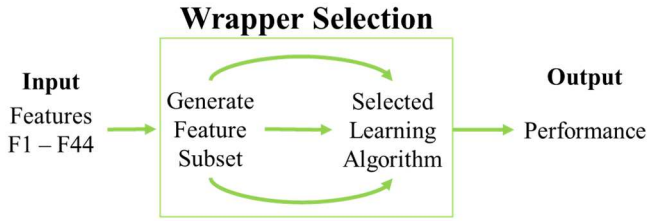

Fig. 16. Wrapper method for feature selection. The algorithm performance either improves each iteration or if it decreases for five iterations then the wrapper generates a feature subset using only features that improve performance

Due to the large computational time required by the wrapper method using an ANN algorithm wrapper selection and the small difference in model accuracy when compared to the DT algorithm, the wrapper selection algorithm was only applied to the DT and kNN algorithms. The wrapper selection method was applied to the three-terrain class problem as well as the four-terrain class problem to optimize the input features from the 44 extracted features for model performance and computation time.

A comparison of the three-terrain optimized feature subset selection between the DT and kNN methods for the selected learning algorithm in the wrapper selection algorithm is presented in Table 5.

TABLE V. THREE-TERRAIN OPTIMIZED FEATURE SUBSET MODEL ACCURACY COMPARISON USING WRAPPER SELECTION

Algorithm	Model Accuracy (%)	Precision (%)	Recall (%)	F_1 Score (%)
Decision Tree (DT)	99.31	99.67	99.64	99.66
k-Nearest Neighbor (kNN)	99.26	99.64	99.62	99.63

When comparing the results from the subset feature optimization using a wrapper selection algorithm the DT model has a higher model accuracy and F_1 score then the kNN model. Additionally, the DT model requires less computational time to build and run. The DT optimized feature subset includes five features, whereas the kNN feature subset includes nine features. A comprehensive list of the five features subset is found in Table 6, and it's corresponding confusion matrix is shown in Fig. 17. When identifying between the tree terrains of concrete, pebble, and sand a DT algorithm with an optimized feature subset of five features can accurately identify the terrain with a model performance accuracy of 99.34%.

TABLE VI. FIVE FEATURE OPTIMIZED SUPSET FOR THREE-TERRAIN CLASSIFICATION

Feature	Instrument
F5	Variance – Magnetometer
F23	Range – Magnetometer
F25	Skewness – Accelerometer
F30	Skewness – LPF Magnetometer
F35	Peak Magnitude – Gyroscope

Actual Terrain	Predicted Terrain		
	Sand	**Pebble**	**Concrete**
Sand	**1139**	12	0
Pebble	16	**827**	0
Concrete	0	0	**3534**

Fig. 17. Confusion matrix of a three-terrain classification using the DT machine learning method with an optimized feature subset of five

A second comparison of the same two machine learning methods for the four-terrain optimized feature subset selection is presented in Table 7.

TABLE VII. FOUR-TERRAIN OPTIMIZED FEATURE SUBSET MODEL ACCURACY COMPARISON USING WRAPPER SELECTION

Algorithm	Model Accuracy (%)	Precision (%)	Recall (%)	F_1 Score (%)
Decision Tree (DT)	99.34	99.73	99.61	99.67
k-Nearest Neighbor (kNN)	99.24	99.62	99.62	99.62

The results from the subset feature optimization using a wrapper selection algorithm indicate that the DT model has a higher model accuracy and F_1 score then the kNN model. One difference between the three and four-class terrain topography classification algorithm is the kNN optimized feature subset for the four-class algorithm consists of seven features wheras the DT optimized feature subset consits of 14 features. The kNN algorithm requires less computational time for four classes than the DT algorithm. The difference in model accuracy performance and F_1 score do not constitute a necessity to use the

DT algorithm. The entire optimized feature subset for the kNN algorithm is presented below in Table 8, and it's corresponding confusion matrix is shown in Fig. 18. The four-terrain topography algorithm can identify between the four terrains of concrete, grass, pebble and sand with a model performance accuracy of 99.24%.

TABLE VIII. SEVEN FEATURE OPTIMIZED SUBSET FOR FOUR-TERRAIN CLASSIFICATION

Feature	Instrument
F7	Energy –Accelerometer
F11	Energy – Magnetometer
F16	Kurtosis – Gyroscope
F23	Range – Magnetometer
F30	Skewness – Magnetometer
F41	Max Frequency – Gyroscope
F43	Max Frequency – Magnetometer

Actual Terrain	Predicted Terrain			
	Sand	**Pebble**	**Concrete**	**Grass**
Sand	**1127**	16	0	8
Pebble	12	**824**	3	4
Concrete	0	2	**3532**	0
Grass	11	4	2	**2647**

Fig. 18. Confusion matrix of a four-terrain classification using the knn machine learning method with an optimized feature subset of seven

IV. CONCLUSIONS AND FUTURE WORK

In this work, two terrain topography classification algorithms have been demonstrated. The terrain topography data was recorded in real world environments with an IMU attached to the foot above the toe to reduce the impact on the user's natural gait. The terrain classification algorithm is not limited to binary classification. A comparison study of multiple machine learning algorithms was presented and feature optimization was performed with the assistance of a wrapper selection to limit overfitting of the algorithms.

The DT machine learning method is the most effective manner to classify the initial three terrains of concrete, pebble, and sand. Using a wrapper selection algorithm, the 44 input features extracted from a two second sliding window were reduced to an optimized feature subset with five features. The DT classification algorithm was trained and tested with a model performance accuracy of 99.31% and an F_1 score of 99.66%

An additional grass terrain was added to the classification algorithm because grass terrain makes up roughly one-fifth of the walkable space in an urban environment, such as London [39]. Adding a fourth class of grass terrain increases the usability of the terrain topography classification algorithm. The fourth class was created by collecting data on a grass field and post-processed and analyzed using the same method as the three-terrain classification algorithm. Four of the five tested machine learning methods are more than 97.5% accurate in determining the correct class of terrain. The wrapper selection algorithm was performed on the DT and kNN learning methods for feature optimization. The DT optimized feature subset accurately identified and predicted the correct terrain topography with a model performance accuracy of 99.34% using 14 features and the kNN utilizes seven optimized features with a model accuracy of 99.24%.

An investigation on splitting the class of grass terrain to account for variation in moisture content was performed. Splitting the grass terrain into wet and dry grass reduced the overall accuracy of the classification algorithms. When only considering wet and dry grass classes, they were misidentified as each other between 14.4 and 21.8%. The terrain topography algorithm is unable to accurately distinguish between wet and dry grass.

A robust algorithm that can detect changes in terrain topography with more comprehensive terrain classifications may be used in comparison to a terrain and topography map or terrain database as an aid for position determination. With a given initial position and a classification algorithm of changes in terrain styles, the position can be compared to known changes from a terrain map or database to calibrate the present position. Additional investigation is needed to understand how walking on different terrains affect existing Zero-Velocity Update and Pedestrian Dead Reckoning navigation solutions, especially in the detection of the stance phase and zero velocity intervals during the gait cycle. Further investigation is also required into the sliding window length to determine how the model performance and accuracy are affected by altering the window length. If model performance is unaffected with a shorter window length, then terrain transitions may be easier to classify terrain topography classification in real time.

In conclusion the terrain topography DT classification algorithm has a model accuracy of 99.31% for the three-terrain classification problem and the four-terrain classification problem uses the kNN method with a model accuracy of 99.24%. The terrain topography classification algorithms are an example of how an IMU is used to detect terrains on which a pedestrian interacts in addition to detecting user behavior.

REFERENCES

[1] Gao, H., and Groves, P. D. (2020). Improving environment detection by behavior association for context-adaptive navigation. *NAVIGATION: Journal of the Institute of Navigation*, *67*(1), 43-60.

[2] Lockhart, T. E., Soangra, R., Zhang, J., and Wu, X. (2013). Wavelet based automated postural event detection and activity classification with single IMU (TEMPO). *Biomedical sciences instrumentation*, *49*, 224.

[3] Maguire, D., and Frisby, R. (2009, October). Comparison of feature classification algorithm for activity recognition based on accelerometer and heart rate data. In *9th. IT and T Conference* (p. 11).

[4] Vleugels, R., Van Herbruggen, B., Fontaine, J., and De Poorter, E. (2021). Ultra-Wideband Indoor Positioning and IMU-Based Activity Recognition for Ice Hockey Analytics. *Sensors*, *21*(14), 4650.

[5] Groves, P. D., Martin, H., Voutsis, K., Walter, D., and Wang, L. (2013, September). Context detection, categorization and connectivity for

advanced adaptive integrated navigation. In *Proceedings of the 26th International Technical Meeting of the Satellite Division of the institute of navigation (ION GNSS+ 2013)* (pp. 1039-1056).

[6] Wu, C., Mu, Q., Zhang, Z., Jin, Y., Wang, Z., and Shi, G. (2016, June). Indoor positioning system based on inertial MEMS sensors: Design and realization. In *2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)* (pp. 370-375). IEEE.

[7] Tarolli, P., Sofia, G., & Ellis, E. (2017). Mapping the topographic fingerprints of humanity across Earth, Eos, 98.

[8] Clarke, K. C., and Romero, B. E. (2017). On the topology of topography: a review. *Cartography and Geographic Information Science*, *44*(3), 271-282.

[9] Kowalsky, D. B., Rebula, J. R., Ojeda, L. V., Adamczyk, P. G., and Kuo, A. D. (2021). Human walking in the real world: Interactions between terrain type, gait parameters, and energy expenditure. PLoS one, 16(1), e0228682.

[10] Nampoothiri, M. H., Vinayakumar, B., Sunny, Y., and Antony, R. (2021). Recent developments in terrain identification, classification, parameter estimation for the navigation of autonomous robots. SN Applied Sciences, 3, 1-14.

[11] Weszka, J. S., Dyer, C. R., and Rosenfeld, A. (1976). A comparative study of texture measures for terrain classification. *IEEE transactions on Systems, Man, and Cybernetics*, (4), 269-285.

[12] Dornik, A., DRĂGUŢ, L., and Urdea, P. (2018). Classification of soil types using geographic object-based image analysis and random forests. *Pedosphere*, *28*(6), 913-925.

[13] Laible, S., Khan, Y. N., Bohlmann, K., and Zell, A. (2012). 3d lidar-and camera-based terrain classification under different lighting conditions. In *Autonomous Mobile Systems 2012* (pp. 21-29). Springer, Berlin, Heidelberg.

[14] Dupont, E. M., Moore, C. A., Collins, E. G., and Coyle, E. (2008). Frequency response method for terrain classification in autonomous ground vehicles. *Autonomous Robots*, *24*(4), 337-347.

[15] Vom Scheidt, J., Wunderlich, R., and Fellenberg, B. (1999). Random road surfaces and vehicle vibration. Progress in industrial mathematics at ECMI, 98, 352-359.

[16] Hashmi, M. Z. U. H., Riaz, Q., Hussain, M., and Shahzad, M. (2019). What lies beneath one's feet? terrain classification using inertial data of human walk. *Applied Sciences*, *9*(15), 3099.

[17] Strozzi, N., Parisi, F., and Ferrari, G. (2018). Impact of on-body IMU placement on inertial navigation. *IET Wireless Sensor Systems*, *8*(1), 3-9.

[18] Voloshina, A. S., Kuo, A. D., Daley, M. A., and Ferris, D. P. (2013). Biomechanics and energetics of walking on uneven terrain. Journal of Experimental Biology, 216(21), 3963-3970.

[19] Chapwouo T, L. D., and Menelas, B. A. J. (2018). Reaction time to vibrotactile messages on different types of soil. In Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (Vol. 3, pp. 155-161).

[20] Hu, B., Dixon, P. C., Jacobs, J. V., Dennerlein, J. T., and Schiffman, J. M. (2018). Machine learning algorithms based on signals from a single wearable inertial sensor can detect surface-and age-related differences in walking. Journal of biomechanics, 71, 37-42.

[21] Wahlström, J., & Skog, I. (2020). Fifteen years of progress at zero velocity: A review. *IEEE Sensors Journal*, *21*(2), 1139-1151.

[22] Strozzi, N., Parisi, F., & Ferrari, G. (2018). Impact of on-body IMU placement on inertial navigation. *IET Wireless Sensor Systems*, *8*(1), 3-9.

[23] Gupta, A. K., Skog, I., and Händel, P. (2015, December). Long-term performance evaluation of a foot-mounted pedestrian navigation device. In 2015 Annual IEEE India Conference (INDICON) (pp. 1-6). IEEE.

[24] Wentworth, C. K. (1922). A scale of grade and class terms for clastic sediments. *The journal of geology*, *30*(5), 377-392.

[25] Krumbein, W. C. (1934). Size frequency distributions of sediments. *Journal of sedimentary Research*, *4*(2), 65-77.

[26] He, W., Guo, Y., Gao, C., and Li, X. (2012). Recognition of human activities with wearable sensors. *EURASIP Journal on Advances in Signal Processing*, *2012*(1), 1-13.

[27] Saeedi, S., Moussa, A., and El-Sheimy, N. (2014). Context-aware personal navigation using embedded sensor fusion in smartphones. *Sensors*, *14*(4), 5742-5767.

[28] Shoaib, M., Bosch, S., Incel, O. D., Scholten, H., and Havinga, P. J. (2014). Fusion of smartphone motion sensors for physical activity recognition. *Sensors*, *14*(6), 10146-10176.

[29] Groves, P. D., Wang, L., Walter, D., Martin, H., Voutsis, K., and Jiang, Z. (2014, May). The four key challenges of advanced multisensor navigation and positioning. In *2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014* (pp. 773-792). IEEE.

[30] Gao, H. (2019). Investigation of Context Determination for Advanced Navigation using Smartphone Sensors (Doctoral dissertation, UCL (University College London)).

[31] Pei, L., Guinness, R., Chen, R., Liu, J., Kuusniemi, H., Chen, Y., Chen, L., and Kaistinen, J. (2013). Human behavior cognition using smartphone sensors. Sensors, 13(2), 1402-1424.

[32] Jain, A. K., Mao, J., and Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. Computer, 29(3), 31-44.

[33] Song, Y. Y., and Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry, 27(2), 130.

[34] Peterson, L. E. (2009). K-nearest neighbor. Scholarpedia, 4(2), 1883.

[35] Daberdaku, S., Tavazzi, E., and Di Camillo, B. (2020). A combined interpolation and weighted K-nearest neighbours approach for the imputation of longitudinal ICU laboratory data. Journal of Healthcare Informatics Research, 4, 174-188.

[36] Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).

[37] Noble, W. S. (2006). What is a support vector machine?. Nature biotechnology, 24(12), 1565-1567.

[38] Lipton, Z. C., Elkan, C., and Narayanaswamy, B. (2014). Thresholding classifiers to maximize F1 score. arXiv preprint arXiv:1402.1892.

[39] Hitchings, S., and Office of the Mayor of London. (2017). *Greener City Fund*. Retrieved from https://www.london.gov.uk/programmes-strategies/environment-and-climate-change/parks-green-spaces-and-biodiversity/green-space-funding/greener-city-fund

[40] Han, Z., and Vanapalli, S. K. (2016). Stiffness and shear strength of unsaturated soils in relation to soil-water characteristic curve. *Géotechnique*, *66*(8), 627-647.

[41] Kim, S. Y., Hong, W. T., and Lee, J. S. (2018). Silt fraction effects of frozen soils on frozen water content, strength, and stiffness. *Construction and Building Materials*, *183*, 565-577.

[42] Vospar, S,. and Press Office. (2022). *Met Office Report: Driest July in England since 1935.* Retrieved from https://www.metoffice.gov.uk/about-us/press-office/news/weather-and-climate/2022/driest-july-in-england-since-1935.

[43] Kumar, V., and Minz, S. (2014). Feature selection: a literature review. SmartCR, 4(3), 211-229.

[44] Karabulut, E. M., Özel, S. A., and Ibrikci, T. (2012). A comparative study on the effect of feature selection on classification accuracy. Procedia Technology, 1, 323-327.

[45] Phuong, T. M., Lin, Z., and Altman, R. B. (2005, August). Choosing SNPs using feature selection. In 2005 IEEE Computational Systems Bioinformatics Conference (CSB'05) (pp. 301-309). IEEE.