

UNIVERSITY OF SZEGED
FACULTY OF SCIENCE AND INFORMATICS
DEPARTMENT OF TECHNICAL INFORMATICS

Doctoral School of Computer Science

**New methods in the application of inertial and
magnetic sensors in online pattern recognition
problems**

Ph.D. Thesis

Péter Sarcevic

Supervisor
Dr. Szilveszter Pletl

Szeged
2018

Contents

1	Introduction	3
1.1	Thesis outline and contributions	4
2	Used methods	5
2.1	Linear Discriminant Analysis-based dimension reduction	5
2.2	Classification methods	6
2.2.1	Nearest Centroid Classifier	6
2.2.2	Multi-Layer Perceptron	6
2.2.3	Naïve Bayes classifier	7
2.2.4	Support Vector Machine	7
2.2.5	k-Nearest Neighbor	8
2.2.6	Classification Tree	8
2.3	Genetic algorithm-based optimization	8
3	Movement recognition using wearable inertial and magnetic sensors	11
3.1	Introduction	11
3.2	Related work and motivation	12
3.2.1	Related work	12
3.2.2	Motivation	16
3.3	Experimental setup	18
3.4	Data acquisition for movement classification	18
3.5	Online movement classification algorithm	19
3.5.1	Error compensation	20
3.5.2	Windowing	24
3.5.3	Feature extraction	25
3.5.4	Dimension reduction	27
3.5.5	Classification	27
3.5.6	Performance evaluation	27
3.5.7	Comparison of selected TDFs with FDFs and TDFs with high memory requirements	36
3.6	Hierarchical-distributed movement classification algorithm	38

3.6.1	Applied movement hierarchies and classification	39
3.6.2	Performance evaluation	41
3.7	Comparison of time- and frequency-domain analysis	45
3.7.1	Applied time-domain features	45
3.7.2	Applied frequency-domain features	46
3.7.3	Performance evaluation	46
3.8	Conclusion	48
4	Real-time vehicle classification using a single magnetometer	52
4.1	Introduction.....	52
4.2	Related work and motivation	53
4.2.1	Vehicle detection using magnetic sensors	53
4.2.2	Vehicle classification using magnetic sensors.....	54
4.2.3	Motivation	56
4.3	Measurement system and data acquisition.....	57
4.3.1	Measurement unit	57
4.3.2	Data acquisition	58
4.4	Sensor calibration and vehicle detection algorithm	59
4.4.1	Algorithm description.....	59
4.4.2	Experimental results	60
4.5	Vehicle classification algorithm	61
4.5.1	Feature extraction	62
4.5.2	Classification	64
4.5.3	Experimental results	64
4.6	Rule-based false detection filtering algorithm.....	70
4.6.1	Used data in rule construction	70
4.6.2	Optimization	71
4.6.3	Performance evaluation	71
4.7	Conclusion	74
	Publications of the author	77
	Bibliography.....	79
	Acknowledgments.....	88
	Summary	89
	Összefoglaló.....	94

Chapter 1

Introduction

Accelerometers, gyroscopes and magnetic sensors are used in a large variety of applications. Earlier, inertial sensors were mainly used only in aerospace and military applications because of their high-cost. With the advances in Micro-ElectroMechanical Systems (MEMS), the availability of small, light, and low-cost sensors with low power consumption have opened new possibilities for their use [1]. Although MEMS magnetic sensors exist, magnetometers based on Anisotropic MagnetoResistive (AMR) technology are more often used.

Accelerometers are motion sensors, which measure linear acceleration in one or more axes. The velocity of an object can be calculated by integrating the object's acceleration over time. By integrating again, the position can be determined. These sensors can be also applied to estimate orientation using the fact that in stationary position the magnitude of the acceleration vector should be 1g ($g \approx 9.81\text{m/s}^2$) due to Earth's gravity, which points to the center of the Earth. This can be used to switch screen orientation in mobile phones or tablets, or for flight stabilization in the case of drones [2]. Another very important application of accelerometers is shock and vibration analysis [3-6], where the acceleration signals are utilized.

Gyroscopes are rotation sensors used to measure angular velocity and are sensitive around a single axis or multiple axes (usually two or three). Orientation can be calculated by integrating once the measured angular velocity. Angular rate sensors are mostly used together with accelerometers, and together they compose Inertial Measurement Units (IMU).

Magnetometers are devices that measure magnetic fields. Scalar magnetometers measure only the magnitude of the vector passing through the sensor regardless of the direction, while vector magnetometers measure the flux density value in a specific direction in three-dimensional space [7]. These sensors are usually used as compasses; thus, they are capable of estimating heading direction based on the Earth's magnetic field [8]. The changes in the magnetic field can be utilized to detect metallic objects.

All three sensor types can be found in consumer electronics such as smartphones, tablets, or smartwatches, and are also used in other commercial applications like video game controllers.

The most common applications of these sensors are Inertial Navigation Systems (INS), where they are used to continuously calculate the position, the orientation, and the speed of a moving object [9-12].

Accelerometers, gyroscopes, and magnetometers are also widely used in pattern recognition applications, such as human motion recognition, gesture recognition [13-14], fall detection and classification [15-16], vibration analysis, etc. Many of these applications use these sensors in the form of Wireless Sensor Networks (WSN). The necessary computation for these applications can be done offline, where an external unit is used for computation, or online, where the algorithms are implemented on the applied embedded systems. Efficient implementation of pattern recognition applications on embedded systems requires appropriate usage of energy, memory, and processing. These systems must meet computational and storage requirements. This is a challenging task, because these applications usually require high sampling rates of the sensors, real-time data processing, and high transmission capabilities.

1.1 Thesis outline and contributions

In this thesis, research results achieved in two pattern recognition applications of inertial and magnetic sensors are presented. The developed algorithms for both applications are online, thus, they are easily implementable on the used microcontroller-based embedded systems.

The used dimension reduction, classification, and optimization methods are described in Chapter 2.

In Chapter 3, the first application is presented, which applies accelerometers, gyroscopes, and magnetometers to classify different arm and body movements. Two sensor boards are used, which are attached to the wrists of the subjects. The proposed algorithms are based on features extracted from the signals generated by the changes in the position and the orientation of the sensors.

The second application is described in Chapter 4, which utilizes a single magnetometer-based system to classify vehicles into multiple classes. The sensor is in stationary position, and the distortions in the measured magnetic field caused by passing vehicles are used during the extraction of features.

The two thesis groups and the corresponding publications can be seen in Table 1.1.

Table 1.1: Thesis groups and corresponding publications.

Thesis group	Publication
Movement recognition using wearable inertial and magnetic sensors	[S3], [S4], [S5], [S6], [S7], [S8]
Real-time vehicle classification using a single magnetometer	[S1], [S2], [S9], [S10]

Chapter 2

Used methods

2.1 Linear Discriminant Analysis-based dimension reduction

The Linear Discriminant Analysis (LDA) method is a widely-used subspace learning method in statistics, pattern recognition and machine learning. This method aims to seek a set of optimal vectors, denoted by $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m] \in \mathfrak{R}^{d \times m}$, projecting the d -dimensional input data into an m -dimensional subspace, such that the Fisher criterion is maximized [17-18]. The Fisher criterion, given in (2.1), aims at finding a feature representation, by which the within-class distance is minimized, and the between-class distance is maximized.

$$\arg \max_{\mathbf{W}} \text{tr}((\mathbf{W}^T \mathbf{S}_w \mathbf{W})^{-1} (\mathbf{W}^T \mathbf{S}_b \mathbf{W})) \quad (2.1)$$

where \mathbf{S}_b and \mathbf{S}_w are the between-class scatter matrix and the within-class scatter matrix, respectively, and are defined as

$$\mathbf{S}_w = \sum_{j=1}^c \sum_{i=1}^{N_j} (\mathbf{x}_i^j - \boldsymbol{\mu}_j)(\mathbf{x}_i^j - \boldsymbol{\mu}_j)^T, \quad (2.2)$$

$$\mathbf{S}_b = \sum_{j=1}^c (\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T, \quad (2.3)$$

where \mathbf{x}_i^j represents the i -th sample of class j , $\boldsymbol{\mu}_j$ is the mean vector of class j , c is the number of classes, N_j is the number of samples in class j , and $\boldsymbol{\mu}$ is the overall mean vector of all classes. The mean vector of a class and the overall mean vector can be calculated as follows,

$$\boldsymbol{\mu}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \mathbf{x}_i^j, \quad (2.4)$$

$$\boldsymbol{\mu} = \frac{1}{c} \sum_{j=1}^c \sum_{i=1}^{N_j} \mathbf{x}_i^j. \quad (2.5)$$

The solution to the problem of maximizing the Fisher criterion is obtained by an eigenvalue decomposition of $\mathbf{S}_w^{-1} \mathbf{S}_b$, and taking the eigenvectors corresponding to the highest eigenvalues. There are $c-1$ generalized eigenvectors. If the number of features is less than $c-1$, then the number of eigenvectors will be equal to the number of features.

2.2 Classification methods

2.2.1 Nearest Centroid Classifier

The Nearest Centroid Classifier (NCC) is used in various areas of pattern recognition because it is simple and fast. The method determines the Euclidean distance from an unknown object to the centroid of each class and assigns the object to the class with the shortest distance. The Euclidean distance between the $x_i \in \mathfrak{R}^n$ feature vector and the n -dimensional m_j vector of mean values for class j can be calculated as

$$\text{dist}(x, m_j) = \sqrt{\sum_{i=1}^n (x_i - m_{ji})^2}. \quad (2.6)$$

2.2.2 Multi-Layer Perceptron

Artificial Neural Networks (ANNs) are inspired by biological neural systems and are used to approximate target functions [19]. The Multi-Layer Perceptron (MLP) is a feedforward ANN, where neurons are organized into three or more layers (an input and an output layer with one or more hidden layers), with each layer fully connected to the next one using weighted connections. An illustration of a three-layer MLP neural network can be seen in Figure 2.1.

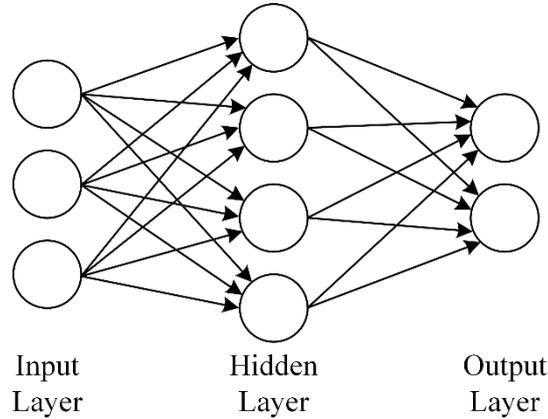


Figure 2.1: An example architecture of a three-layer MLP network.

A neuron has an activation function that maps the sum of its weighted inputs to the output. The o_j output of one node can be defined as

$$o_j = f(\mathbf{w}_j \cdot \mathbf{x} + b), \quad (2.7)$$

where \mathbf{x} is the input vector, \mathbf{w}_j is the vector containing the weights, b is the bias value, and f is the applied activation function. The activation or transfer functions are usually sigmoid or linear functions, while the bias allows a shift in the transfer function.

Most commonly MLP networks are trained using the backpropagation algorithm, which is a supervised learning method. The algorithm employs gradient descent to attempt to

minimize the squared error between target values and the network output values. This is achieved by updating the weights and the biases in the direction of the negative gradient of the performance function.

A faster convergence is produced with the conjugate gradient algorithm, which performs a search along conjugate directions. The Scaled Conjugate Gradient (SCG) backpropagation algorithm, developed by Moller [20], was designed to avoid the line-search of the conjugate gradient algorithms. The SCG algorithm requires more iterations to converge, but the number of computations in each iteration is reduced because no line search is used.

The Levenberg-Marquardt (LM) backpropagation algorithm is also widely used for training, since it is one of the fastest methods. It updates weight and bias values according to LM optimization [21].

MLPs with different initial random weights provide different results, because the corresponding performance functions are not the same. Thus, multiple trainings are required for one configuration to achieve better results.

In the case of classification applications, the vector of extracted feature values forms the input vector, while in the output layer a neuron is assigned to each class.

2.2.3 Naïve Bayes classifier

The Naïve Bayes Classifier (NBC) is a highly practical Bayesian learning method. It is based on the simplifying assumption that, given the target value of the instance, the attribute values are conditionally independent, and the probability of observing the conjunction for attributes is just the product of the probabilities for the individual attributes [19]. The approach used by the NBC is presented in (2.8).

$$v_{NB} = \arg \max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j) \quad (2.8)$$

where v_j denotes the target value output of the classifier, V is the finite set of target values, a_i are the attribute values, and $P(v_j)$ are the probabilities of v_j target values.

The NBC includes first a learning step in which the various $P(v_j)$ and $P(a_i|b_j)$ terms are estimated. New instances are classified by applying the rule in (2.8).

2.2.4 Support Vector Machine

In Support Vector Machines (SVM), a data point is viewed as a p -dimensional vector, and the goal is to separate such points with $(p-1)$ -dimensional hyperplanes [22]. The hyperplane can be defined as

$$F(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b, \quad (2.9)$$

where \mathbf{x} is the vector to be recognized, \mathbf{w} is the normal vector to the hyperplane, and b determines the offset from the origin along the normal vector. (2.10) defines the normal vector, and is the subject to the condition expressed in (2.11).

$$\mathbf{w}^* = \sum_{i=1}^l \alpha^{i*} \cdot y^i \mathbf{x}^i, \quad (2.10)$$

where α^i is the i -th Lagrange multiplier, $y^i \in \{-1, 1\}$, and l is the number of support vectors.

$$\alpha^{i*} [y^i (\mathbf{w}^{*T} \mathbf{x}^i + b^*) - 1] = 0, \forall \alpha^i \neq 0 \quad (2.11)$$

The function of the hyperplane is not suitable for solving linearly non-separable problems or dealing with more than two classes. To classify data into multiple classes, two common methods can be used: “one-versus-one” (OvO) and “one-versus-all” (OvA).

2.2.5 k-Nearest Neighbor

The k-Nearest Neighbor (k-NN) algorithm classifies the objects based on the closest training examples in the feature space [23-24]. To classify a new observation, the method finds the k nearest samples in the training data and assigns the new sample to the class which provides the most neighbors. The Euclidean distance is used to measure the distance between two instances [19], which can be given as

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{r=1}^n (x_{ir} - x_{jr})^2}, \quad (2.12)$$

where \mathbf{x}_i and \mathbf{x}_j are the two instances, and x_{ir} and x_{jr} denote the r th values in the feature vectors.

The selection of a proper k parameter is very important, since it can affect the decision of the k-NN classifier. A predefined rule does not exist for the selection of the k value.

2.2.6 Classification Tree

The Classification Tree (CT) is a rule-based algorithm, which uses a tree-like set of nodes for classifying inputs [23, 25]. The tree has predefined conditions at each node of the tree, and makes binary decisions based on these rules. The condition of the following node is checked until a leaf is found that contains the classification result.

The conditions can be given as “is $x_i \leq \tau_i$?”, where x_i is the feature value and τ_i is the threshold value for a given i th feature.

Learning of classification trees is usually done using algorithms that employ top-down, greedy search through the space of possible trees [19]. The most widely known algorithms are the ID3 and its successor C4.5.

2.3 Genetic algorithm-based optimization

Evolutionary algorithms are optimization techniques inspired by biological evolution. Their basic principle is a search on a population of solutions, which is controlled by the laws of biology. They are an effective tool for solving nonlinear, multicriteria optimization tasks.

Genetic algorithms (GA) belong to evolutionary algorithms and are based on Darwin’s theory of evolution [26]. According to this theory, populations of living beings are constantly

evolving, they become more and more perfect, and better individuals develop. The development process is controlled by natural selection, according to which better individuals survive, while weaker ones perish.

GAs use a population of artificial chromosomes. A chromosome represents a solution to a problem and can be given most commonly as a binary string or a vector of real values, which are also called as allele values or simply alleles. Each solution has a fitness value, which is a real number that measures how good a solution is to the given problem. The fitness value is determined using the fitness function.

By analogy with biology, the chromosome is referred to as the genotype, whereas the solution it represents is known as the phenotype.

The algorithm starts with a randomly generated initial population. The evolution in the algorithm is an iterative process, with the population in each iteration called a generation. A fitness-based selection and recombination is made to produce a successor population, the next generation. During recombination, parents are selected, and their genetic material is recombined to produce child chromosomes. This process is iterated until some stopping criterion is reached. The algorithm this way finds a good solution, since during the iterations, a sequence of successive generations evolves, and the average fitness tends to improve.

The generation of successors in a GA is determined by a set of genetic operators [19, 27]:

- Selection – during the selection process individuals are selected from the population, which will be involved in the creation of successors. Selection is based on fitness values. Individuals with higher fitness should have a greater chance of selection than those with lower fitness. This creates a selective pressure towards more highly fit solutions.

Many selection methods exist. The most-known is the roulette wheel method. In this method, the population is depicted as a roulette wheel, where the individuals have a size proportional to their fitness value. The circumference of the circle is equal to the sum of the chromosomes' fitness values. The selection is done using generated random numbers with uniform distribution, which show which sector of the circle (individual) should be selected. Thus, chromosomes with higher fitness value have a higher probability to be selected.

Other popular methods include: stochastic universal sampling, tournament selection, truncation selection, etc.

- Crossover – the selected individuals are sorted into pairs. The crossover operator achieves the mixing of the genetic material from two selected parent chromosomes to produce one or two child chromosomes.

A commonly used crossover operator is the one-point crossover. This method uses a generated random number with uniform distribution, which is the crossover point

that splits the chromosomes of the parents into two parts. One part of the parents' chromosomes is exchanged, which leads to two new individuals.

Two- and multi-point crossovers also exist. The uniform crossover method is also widely used, which constructs a child by selecting uniformly between parent allele values at each position.

- Mutation – the mutation operator can change the allele values of an individual. A random number with uniform distribution is generated for each position of the chromosome, which is compared to a defined mutation rate. If the number is larger than the mutation rate, no change will occur at the given position. If the value is smaller than or equal to the mutation rate, then the value is flipped from 0 to 1 or vice versa in the case of binary strings, or is replaced with a random real number in the case of vectors of real values.

Chapter 3

Movement recognition using wearable inertial and magnetic sensors

3.1 Introduction

The analysis and real-time monitoring of human body motion is a widely-studied field of industrial, entertainment, health, and medical applications [28]. Such systems can be used for robot control, human-computer interaction, assisted living, gaming, fall detection, epileptic seizure detection, telerehabilitation, analysis of daily activities, emergency detection, health monitoring, or even human worker activity recognition in industrial environments.

Human motion can be split into two basic categories, activities and movements. Movements typically last for several milliseconds or seconds, while an activity comprises of different movements, and can last for even minutes or hours [22]. For example, a “walking” activity contains several short physical leg movements. But more complex activities can also be defined, such as “cooking”, which is composed of multiple shorter activities in a specific sequence, like “walking”, “arm raising”, “standing”, etc.

Sensor-based motion recognition integrates the emerging area of sensor networks with machine learning techniques. Inertial and magnetic sensors are widely used in wearable devices for motion recognition, due to their small size, low cost, and small energy consumption. These wearable devices applied to human bodies form Wireless Body Sensor Networks (WBSN) [29]. Another option for human motion monitoring can be the use of Personal Area Networks (PAN), which are composed of environmental sensors, like Radio-Frequency Identification (RFID) readers, video cameras, or sound, pressure, temperature, luminosity, and humidity sensors. The vision-based activity recognition systems are the most popular types of PANs. One of the main advantages of body sensor networks to systems using cameras with fix places is that they support persistent monitoring of a subject during daily activities both in indoor and outdoor environments. The vision-based systems are also influenced by environmental factors, such as lighting conditions, and they incur a significant amount of computational cost.

Due to the difficult implementation of pattern recognition algorithms on resource constrained wireless nodes, the design of WBSN-based applications is a very complex task [30-31]. They should also be wearable, which affects the possible usable battery size and therefore its duration.

3.2 Related work and motivation

3.2.1 Related work

In the research of using inertial and magnetic sensors in human movement recognition systems, various types and positions of the sensors, and methods for recognition were tested for different applications [32]. Classification is typically done in a two-stage process. First, features are derived from windows of sensor data. A classifier is then used to identify the motion corresponding to each separate window of data.

Table 3.1 summarizes the applied activity classes, sensor types and their placements, feature extraction modes, processing window widths, sampling frequencies, classification methods, and achieved accuracies in relevant works. The used abbreviations are described in the following subchapters.

Table 3.1: Summary of relevant works.

Related work	Activity classes	Sensors and placement	Feature extraction	Processing window width / Sampling frequency	Classifiers	Accuracy
[33]	8 classes (moving, complex)	ACC: waist, thigh, ankle	TDFs FDFs Wavelets	2s / 64Hz	k-NN	95%
[23]	21 classes (stationary, moving, complex)	ACC, GYR, MAG: knees, wrists, chest	TDFs FDFs	5s / 25Hz	PCA+BDM	99.1%
					PCA+SVM	98.6%
					PCA+DTW	98.5%
					PCA+k-NN	98.2%
					PCA+ LSM	89.4%
					PCA+MLP	86.9%
[34]	6 classes (stationary, moving, complex)	ACC: chest	TDFs FDFs	10s / 20Hz	LDA+MLP	94.43%
[35]	8 classes (stationary, transitional, moving)	ACC: right thigh	TDFs	1s / 20Hz	MLP+HMM	80.88%
[36]	4 classes	ACC: chest	TDFs	1s / 100Hz	SVM	94.73%

	(stationary, transitional)					
[22]	6 classes (stationary, moving, complex)	ACC, GYR: right wrist, right foot	TDFs	1.6s / 20Hz	SVM	97.2%
[25]	6 classes (stationary, moving, complex)	ACC, GYR, MAG: multiple places	TDFs FDFs	3s / 6.25Hz (ACC), 100Hz (GYR), 7.69Hz (MAG)	CT	97%
					Decision table	88%
					NBC	78%
[37]	13 classes (moving, complex)	ACC, GYR: dominant wrist	TDFs FDFs	3.88s / 33Hz	SVM	97.2%
					MLP	96.73%
					RBF	95.67%
[24]	6 classes (stationary, moving, complex)	ACC: waist	TDFs	1s / 10Hz	MLP	98.3%
					k-NN	94.1%
[38]	12 classes (stationary, transitional, moving)	ACC: chest, right thigh, left ankle	TDFs FDFs Wavelets	1s / 25Hz	k-NN	99.25%
					Random forest	98.95%
					SVM	95.55%
					SLGMM	85.05%
					HMM	83.89%
					GMM	75.60%
[39]	6 classes (stationary, moving)	ACC, GYR: waist	TDFs	0.64s, 1.28s, 2.56s / 50Hz	Lazy learner	99%
					CT	97%
					Rule-based classifier	97%
					NBC	84%
[40]	5 classes (moving, complex)	ACC: right wrist	TDFs FDFs	1s / 100Hz	CT	97.8%

Activity classes

In the related work, many activity classification approaches were used. The most widely used activities were standing and walking, which can be found in almost all works. Besides standing, other stationary activities can also be found in the literature, such as lying [34], sitting [25, 41-42], or both [23, 35, 38-39]. Different transitional movements were also parts of the activity classes in some works, e.g. sit-to-stand and stand-to-sit [35, 38, 41], lie-to-sit and sit-to-lie [35], lie-to-stand and stand-to-lie [41], or stopping after walking [36]. Regarding the classification of longer motional activities, various speeds and types of forward movements were also tested, such as slow, normal and rush walking [25], jogging

[22, 33, 42-43], and running [23-25, 33, 40]. Some works tried to differentiate different directions of an activity type, like level walking, walking downstairs and upstairs [23, 33-34, 38-39, 42] or walking backwards [43]. Yang et al. [42] recorded even continuous rotational movements, such as walking left-circle or right-circle, and turning left or right. Special complex activities were also parts of the constructed databases, e.g. falling [24, 41], jumping [23, 33, 42], writing [22], brushing teeth [40, 44], eating and drinking [44], sweeping the floor, lifting a box onto a table, bouncing a ball [43], driving [34], cycling [23, 44], etc.

Sensors and placement

The accelerometer (ACC) is the most popular sensor for monitoring the motion of the human body. This sensor measures acceleration in one or more axes. As seen in Table 3.1, many researchers used only a single unit to achieve activity recognition, but they differed in the placement of the sensor. Others applied multiple sensors fixed to different parts of the body. Beside the works listed in Table 3.1, a single accelerometer fixed to the chest was applied in [41], the authors of [45] utilized two accelerometer-based data loggers, which were mounted on each wrist, while Bao and Intille [44] applied five biaxial sensors placed on each subject's right hip, dominant wrist, non-dominant upper arm, dominant ankle, and non-dominant thigh.

Gyroscopes (GYR), which measure angular velocity around one or more axes, are less popular in movement recognition applications, and are mostly used together with accelerometers. None of the related researches used only gyroscopes. Tri-axial accelerometers and gyroscopes used together provide six degrees of freedom (6DoF) sensor units. Yang et al. [42] used measurement units containing a triaxial accelerometer and a biaxial gyroscope, and placed them to eight places on the body: the wrists, the ankles, the knees, the hip, and the left elbow.

The fusion of inertial sensors and magnetometers (MAG) was also reported in the literature. The magnetic sensors measure the Earth's magnetic field, and thus, they are able to detect rotational movements compared to the magnetic north. Magnetic sensors are usually used together with the inertial sensors, which provides 9DoF measurement systems, but [46] utilized only magnetometers for activity classification. The authors used sensor gloves with 9 magnetic sensors on both hands, and tried to classify simple (walking, running) and complex (shave, brush teeth, use electric toothbrush, etc.) activities. The sensors of mobile phones for activity recognition were utilized in [47], while Field et al. [43] utilized an inertial motion caption system, comprised of 17 inertial sensors attached to different parts of the body. The 9DoF sensors were combined to get a global orientation through a Kalman Filter.

Feature extraction

As activity and movement recognition is a typical pattern recognition problem, feature extraction plays a crucial role during the recognition process. Sensor-based features can be classified into three categories: time-domain features (TDFs), frequency-domain features (FDFs), and features computed using time-frequency analysis.

Most of the related researches used TDFs and/or FDFs. The type of features and their frequency of usage in references are shown in Table 3.2. It can be concluded that the most used TDFs are the mean and the standard deviation, and the most frequent FDFs are the spectral energy and the frequency-domain entropy.

Table 3.2: Used feature types in related works.

Feature type	References
Time-domain features	
standard deviation or variance	[22-25, 33-40, 48-49]
mean	[22-25, 33-35, 37-40, 44, 49]
root mean square	[22, 24, 37-38, 40]
correlation	[22, 24-25, 34, 44]
number of zero crossings	[25, 38, 40, 48]
kurtosis	[23, 37-38, 49]
range	[22, 24, 36, 38]
skewness	[23, 37-38, 49]
maximum	[22, 37, 39]
minimum	[37, 39]
number of rapid changes	[48]
magnitude of the first peak of the autocorrelation	[48]
Frequency-domain features	
frequency-domain entropy	[25, 33-34, 37-38, 44]
spectral energy	[24-25, 33, 38, 40, 46]
magnitude of the defined first few highest peaks	[33, 37, 46, 49]
frequency of the defined first few peaks with highest amplitude	[23, 46, 49]
correlation between axes	[33, 37]
median frequency	[25, 48]
DC component	[38]
median power	[48]
principal frequency	[33]

Using wavelet analysis, the signal is decomposed into a series of coefficients, which carry both spectral and temporal information about the original signal. Two works, [38] and [33], tested this feature extraction method for the classification of activities. The authors of [33] utilized the next features: the sum of the squared detail coefficients at different levels, the sum of the squares of the detail and wavelet packet approximation coefficients across

different levels, the standard deviations and root mean square (RMS) values of detail and wavelet packet approximation coefficients at a few different levels, and the sum of the absolute values of coefficients at different levels. Attal et al. [38] applied the following features: the sum of detail coefficients of wavelets, the sum of squared detail coefficients of wavelets, the energy of detail wavelets coefficients, and the energy of approximation wavelets coefficients.

Processing window width and sampling frequency

Windowing plays also a very important role during the extraction of features. Usually features are computed in fixed-size windows, which are shifted also with a fixed time. In the related work, the width of the applied processing windows is between 1s and 10s, and the smallest size, 0.64s, was tested by [39]. The sampling frequency is also a very important factor in the processing phase. In relevant works, the applied frequencies were between 10Hz and 100Hz.

Classifiers

The classification of the defined activities using the computed feature vectors can be done using different classification methods. As shown in Table 3.1, the most popular classifiers in relevant works are: SVMs, the k-NN method, decision trees or CTs, NBCs, and MLP neural networks. Some other methods were also tested, as Radial Basis Function (RBF) neural networks, the Least-Squares Method (LSM), Bayesian Decision Making (BDM), Dynamic Time Warping (DTW), decision table, rule-based classifier, Gaussian Mixture Modeling (GMM), Supervised Learning GMMs (SLGMM), the k-means method, random forest, lazy learner, and Hidden Markov Models (HMMs). In some researches, the classifiers were used together with some dimension reduction methods. The most common methods are the Principal Component Analysis (PCA) and the LDA. The authors of [49] applied the Generalized Discriminant Analysis (GDA) method with the multiclass relevance vector machine classifier. Some researchers even tested two different classification methods together: neural networks and HMMs [35], hierarchical temporal memories and SVMs [41], CTs and HMMs [46], k-means clustering and HMMs [47].

3.2.2 Motivation

The goal of this research was to develop a wearable wireless system which does not disturb the user in free movement, and which can efficiently recognize basic body and arm movements using an online classification algorithm. It was also important to explore different setups to minimize the cost, the energy consumption, and the memory requirements, besides maximizing the classification efficiency.

In the proposed prototype system, 9DoF sensor boards mounted on WSN motes were used, which were attached to the wrists of the subjects. The developed system was used to

record measurements for multiple activities. The proposed system does not require any additional server for the processing of the data, and it is also suitable for the logging of the activities.

Related works mainly do not deal with the implementability of the algorithms on the used hardware or use a centralized server to do the necessary computation. The use of processing servers can cause several disadvantages. First, the communication in the network is very costly due to the high sampling frequencies of the sensors, and secondly, since the subjects are moving, they can get out of the range of the server if its place is fixed. Some works implement their algorithm on a smartphone, but the performance of these systems can be affected by the varying placement of the units, or their use during the operation of the algorithm. Based on the above considerations, it was reasonable to develop an online method, and to examine the hardware implementability of different classification algorithms. Since related studies mainly consider complex activities or use more than 1-2 seconds of data for the classification of motions, it was necessary to investigate the barriers in the performance when decreasing the processing window width. Related works which utilize multiple sensor types also do not consider the effect of different sensor types on classification efficiency.

In this work, two online movement classification approaches are proposed.

In the first approach, the classification is done on one of the units, which uses features extracted on both units. This algorithm requires the transmission of the computed feature vector from one of the motes to the mote computing the movement class. To find the optimal setup, multiple classification methods were investigated for various datasets, which were generated based on different sampling frequencies, processing window widths, feature extraction modes, and used sensor types. LDA-based dimension reduction was also tested to investigate its effect on the tested classification methods in the meaning of recognition efficiency, memory consumption, and training time. The extraction and reduction of feature vectors were also tested in multiple ways. The features were computed utilizing the sensor axes separately and using the magnitude. To reduce the required computation, only time-domain analysis was performed during feature extraction. An aggregation-based feature reduction method is also proposed, which can help the system to be less sensitive to differences in orientations of the sensors on the arms.

The second approach is a hierarchical-distributed algorithm, where both units compute their own movement class, and one of the units then combines the two classes to determine the class of the entire body. Two different movement hierarchies were examined. This algorithm was also tested for datasets generated using different processing window widths, feature extraction modes, and used sensor types.

Since the proposed algorithms apply only TDFs with low memory needs, a comparison of datasets based on only time- and only frequency-domain analysis is also presented to examine their performance on the collected measurement data.

3.3 Experimental setup

The sensor devices used in body sensor networks must be designed with the aim of providing the highest degree of mobility for the patients. They must be small, lightweight, and wireless wearable units.

The used prototype system, which can be seen in Figure 3.1 consists of an IRIS WSN mote, and a 9DoF digital sensor board connected to it. The IRIS mote is equipped with an Atmel ATmega 1281L 8-bit microcontroller, and an RF231 IEEE 802.15.4 compatible radio transceiver. The current draw of the microcontroller is 8mA in active mode, and 8 μ A in sleep mode, while the radio transceiver consumes 17mA during transmission, and 16mA during reception. The maximal data throughput of the radio transceiver is 250kbps, and its outdoor range is over 300m. The connected 9DoF sensor board is made up of an ADXL345 tri-axial MEMS accelerometer, an ITG3200 tri-axial MEMS gyroscope, and an HMC5883L tri-axial magnetoresistive technology-based magnetometer. The ADXL345 is a low power accelerometer (the current draw is 40 μ A in measurement mode, and 0.1 μ A in sleep mode), which can measure up to $\pm 16g$ in 13-bit resolution with the highest sampling rate of 3.2kHz. The gyroscope features a 16-bit analog-to-digital converter, and it can measure angular rate in a range of $\pm 2000\text{deg/s}$ with 8kHz frequency. The normal operating current of the gyroscope is 6.5mA, while the sleep mode current is 5 μ A. The measurement range of the magnetic sensor is $\pm 810\mu\text{T}$ in 12-bit resolution with 160Hz maximal sampling rate, and it consumes 2 μ A current draw in idle mode, while 100 μ A in measurement mode.

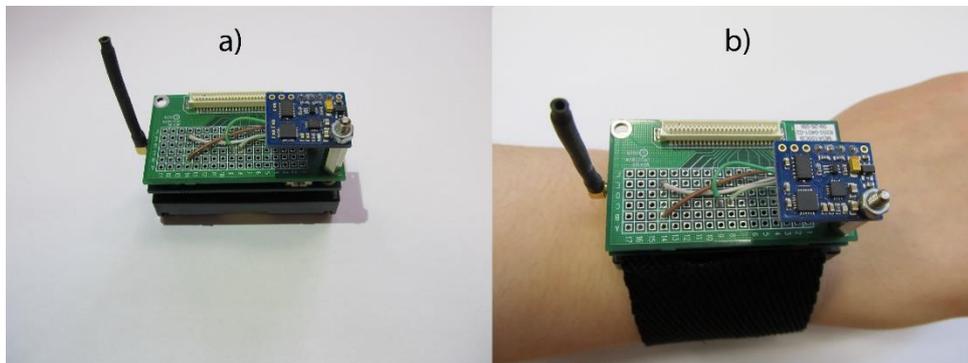


Figure 3.1: a) The prototype measurement system, b) The unit attached to the wrist.

A TinyOS-based driver was developed and implemented to configure the sensors and cyclically read the measurement data. The data are read from the sensors via the I2C interface, and sent via wireless communication to a BaseStation mote, which uses serial communication to forward the data to a PC.

3.4 Data acquisition for movement classification

Eleven activities were defined in order to recognize specific arm movements in stationary positions and also during the movement of the body. The used activities are the following:

1. “standing without movement of the arms”,
2. “sitting with the arms resting on a table”,
3. “walking”,
4. “turning around in one place”,
5. “jogging”,
6. “raising and lowering the left arm during standing”,
7. “raising and lowering the right arm during standing”,
8. “raising and lowering both arms during standing”,
9. “raising and lowering the left arm during walking”,
10. “raising and lowering the right arm during walking”,
11. “raising and lowering both arms during walking”.

Data were collected with the help of nine male subjects (ages between 20 and 50, and height between 165cm and 190cm) for all activities. The IRIS motes with the attached 9DoF sensor motes were mounted on each wrist of the subjects. The data were recorded in fixed-length sessions of 20s for all activities using 125Hz sampling frequency, which means 2500 measurements per sensor. The measurements were performed in a laboratory environment.

3.5 Online movement classification algorithm

In the proposed algorithm, the classification is performed in four main stages. The software architecture with the four stages can be seen in Figure 3.2. In the first step, the measurement data are preprocessed (Stage I.). In the second stage (Stage II.), features are extracted from the signals on each unit. Possible aggregation of the extracted features is also done in this stage. The proposed algorithm assumes the transmission of the vector of the extracted features from one mote to the other, and the rest of the algorithm should be implemented in the microcontroller of the receiving device. Dimension reduction is done in the third stage (Stage III.), while classification is performed in the fourth stage (Stage IV.). Two different algorithms were applied and tested. In the first type, the third stage is not performed, and the classifiers receive the feature vectors directly, while in the second case the data sets are dimensionally reduced, so the classifiers have less input parameters. The advantage of the dimension reduction method is that it removes the redundant information.

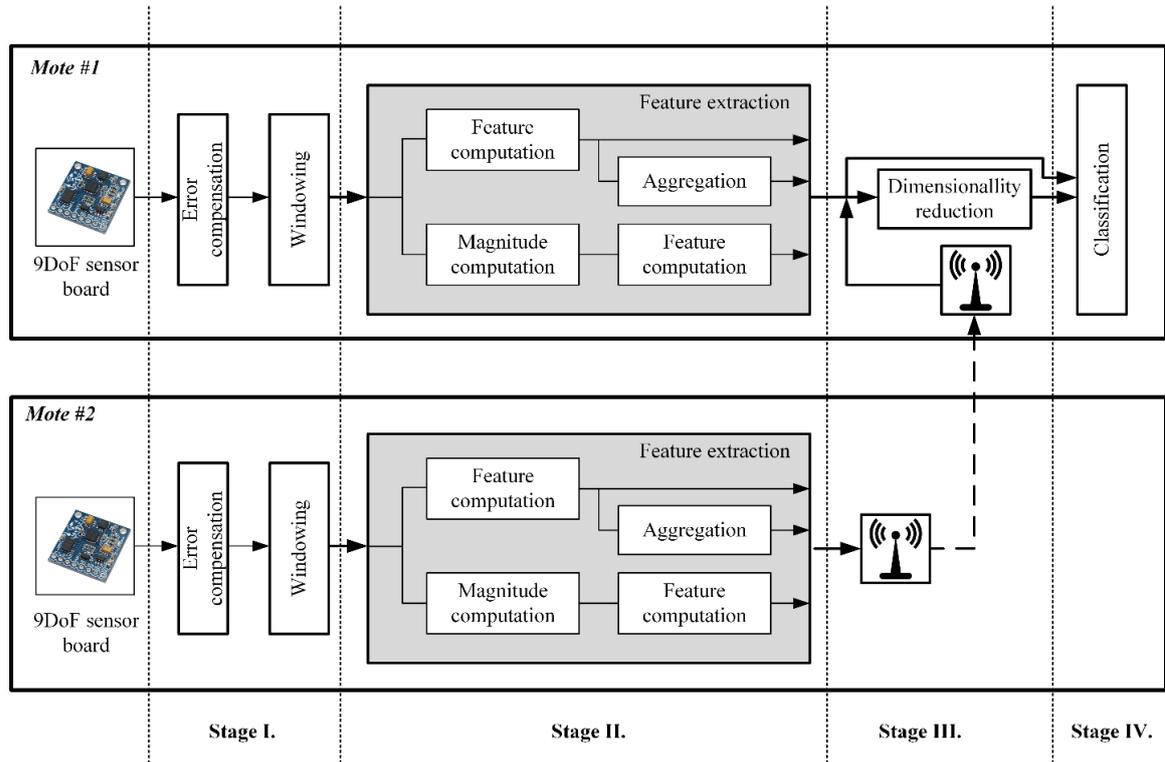


Figure 3.2: Software architecture.

3.5.1 Error compensation

The applied sensor types suffer from various types of errors that can be classified as deterministic or systematic errors and random or stochastic errors. Deterministic errors are caused by manufacturing defects and can be compensated by sensor calibration. The calibration is a process of comparing the sensor's output with the known reference information to estimate these error coefficients, which form an exact relationship between observed readings and expected outputs.

These errors have especially high effect in applications which utilize accelerometers and gyroscopes to obtain position and orientation, since the measurements need to be integrated. Because of the integration process, even very small errors at the output accumulate very rapidly and the position error becomes considerably large [50]. Although the measurements in the proposed system are not integrated during feature extraction, it is very important to calibrate the sensors before use, since the errors can affect also the movement classification process.

Error models

Systematic errors include bias, scale factors and nonorthogonality errors [51].

The bias error is the deviation of the output from the zero level when the input to the sensor is zero. The bias should be zero for all axes in an ideal sensor.

$$\mathbf{b} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}, \quad (3.1)$$

where b_x , b_y , and b_z in the \mathbf{b} bias vector are the bias values on each axis.

Scale factors determine the sensor's sensitivity at each axis. In an ideal situation, the sensors which make up the triad should be identical, and for the same input they should give the same output. The \mathbf{S} matrix consisting the scale factors can be represented as:

$$\mathbf{S} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix}, \quad (3.2)$$

where S_x , S_y , and S_z are the scale factors for each axis.

Nonorthogonality between axes is the inaccuracy resulting from the imperfection in sensor mounting during its manufacturing. Misalignment errors are introduced due to nonalignment of sensor's sensitive axis and the mounting platform. Ideally, the sensor axes should be perfectly perpendicular.

$$\mathbf{M} = \begin{bmatrix} 1 & m_{xy} & m_{xz} \\ m_{yx} & 1 & m_{yz} \\ m_{zx} & m_{zy} & 1 \end{bmatrix}, \quad (3.3)$$

where m_{xy} , m_{xz} , m_{yx} , m_{yz} , m_{zx} , and m_{zy} in the \mathbf{M} matrix are the misalignment error coefficients.

The relation between the measured and the real values can be given as:

$$\begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix} = \begin{bmatrix} 1 & m_{xy} & m_{xz} \\ m_{yx} & 1 & m_{yz} \\ m_{zx} & m_{zy} & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} + \begin{bmatrix} \eta_x \\ \eta_y \\ \eta_z \end{bmatrix}, \quad (3.4)$$

where o_x , o_y , and o_z are the measured sensor outputs, while r_x , r_y , and r_z are the real measurement values on each axis. The terms η_x , η_y , and η_z assign the noise values that are generally assumed to be white Gaussian.

Combining the \mathbf{M} and \mathbf{S} matrices into one matrix, a calibration model utilizing 12 parameters can be given as:

$$\begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix} = \begin{bmatrix} m_{xx} & m_{xy} & m_{xz} \\ m_{yx} & m_{yy} & m_{yz} \\ m_{zx} & m_{zy} & m_{zz} \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \quad (3.5)$$

A simplified relation, which applies 9 parameters, can be given using the model seen in Figure 3.3 [52-53]. The misalignment error matrix in this case can be given as (3.6).

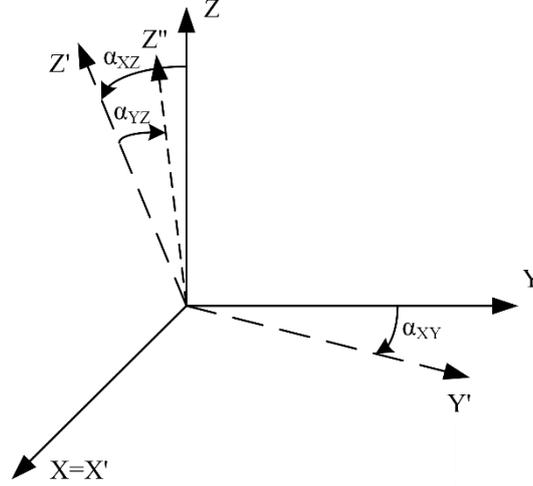


Figure 3.3: Used model for the misalignment angles.

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ \sin(\alpha_{xy}) & \cos(\alpha_{xy}) & 0 \\ \sin(\alpha_{xz})\cos(\alpha_{yz}) & \sin(\alpha_{yz}) & \cos(\alpha_{xz})\cos(\alpha_{yz}) \end{bmatrix}, \quad (3.6)$$

where α_{zy} , α_{xz} , and α_{yz} are the misalignment angles.

The real measurement values using the measured outputs and the calibration parameters can be given as:

$$\mathbf{r} = (\mathbf{MS})^{-1}(\mathbf{o} - \mathbf{b}), \quad (3.7)$$

where $\mathbf{o} = [o_x \ o_y \ o_z]^T$ and $\mathbf{r} = [r_x \ r_y \ r_z]^T$.

In the case of magnetometers, beside the systematic errors, the measurements are also affected by external magnetic influences, which can be classified as hard and soft iron effects [53-54]. Hard iron errors are time-invariant, undesired magnetic fields generated by ferromagnetic materials with permanent magnetism, which are additive to the Earth's magnetic field. Soft-iron distortion is the result of a material that influences, or distorts, a magnetic field, but does not necessarily generate a magnetic field itself, and is therefore not additive.

The hard iron effect can be given as:

$$\mathbf{b}_{HI} = \begin{bmatrix} b_{HIx} \\ b_{HIy} \\ b_{SIz} \end{bmatrix}, \quad (3.8)$$

where b_{HIx} , b_{HIy} , and b_{HIz} are the bias values for of each sensor axis in the \mathbf{b}_{HI} bias vector.

The soft iron effect can be represented as:

$$\mathbf{A}_{SI} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad (3.9)$$

where a_{ij} are the coefficients of the A_{SI} matrix.

The relation between the measured and real values in the case of magnetometers can be given as:

$$\mathbf{o} = \mathbf{MS}(A_{SI}\mathbf{r} + \mathbf{b}_{HI}) + \mathbf{b} + \boldsymbol{\eta}, \quad (3.10)$$

where $\boldsymbol{\eta} = [\eta_x \quad \eta_y \quad \eta_z]^T$.

Calibration principles

Calibration can be done with and without additional equipment. High-precision equipment can be used in laboratory environment to generate known references which can be compared with the sensor's outputs. Due to the growing usage of the utilized miniature sensors, and unavailability and high cost of calibration equipment, in recent years, intensive research was done to develop calibration methods which do not require additional equipment.

Calibration without additional equipment can be also separated into two basic approaches [51]. The first one uses measurements acquired in specific stationary positions or during specified movements, which are utilized to compute calibration parameters based on different basic principles. The second is the Kalman filter-based approach [50, 55-57], which is a widely known state estimation technique which tries to obtain the unknown parameter based on the system model and observations accrued over a period of time. These techniques aim to estimate navigation states along with calibration parameter together in one framework.

The computation of the calibration parameters can be done online or offline. In the case of online calibration, the parameters are computed in real-time on the measurement device, while in offline algorithms previously collected measurements are utilized.

In traditional accelerometer calibration methods, the sensor is positioned and held stationary at various known reference orientations throughout the test [52]. This is known as the multi-position or the 1g test, since later the algorithms use the fact that the magnitude of the acceleration vector should be 1g when the sensor is not in motion due to the Earth's gravity. The known algorithms mainly apply a six-position test, where during the measurements the three axes of the sensors are aligned to be nearly +1g and -1g. The reference orientations can be set using precise machines [58], which provide information about the exact orientations, or by hand, where the exact orientations are not known [59]. The calibration of accelerometers using the multi-position approach can be reduced to 3D-ellipsoid fitting problems [60].

The calibration of gyroscopes depends on the quality grade of the sensor, since high-precision gyroscopes are capable of measuring the Earth's angular velocity, which enables the use of multi-position tests [53]. Lower-grade MEMS gyroscopes are not capable of sensing the Earth's angular velocity, thus, they need to be exposed to different reference angular rates. This can be achieved by mounting the sensor on a precise rotation turntable

(rate table) [58]. If such reference angular rates are not available, calibration is commonly performed by simply rotating the gyroscopes manually [57, 61]. Calibrated accelerometers and/or magnetometers embedded in the sensor unit together with the gyroscope can also be used to provide reference information. Both sensors can provide information about orientation changes. In the case of accelerometers, the Earth's gravity vector can be used in stationary positions [62], while the Earth's magnetic field vector is utilized in the case of magnetometers [63-64].

Magnetometers are also calibrated based on data acquired during multi-position tests. The basic concept of calibration is similar as in the case of accelerometers, but hard iron and soft iron effects must be also taken into consideration [65]. The reference input is the Earth's local magnetic field vector, which is constant in one point, but can largely differ in different places on the Earth. The local Earth's magnetic field vector can be extracted from geomagnetic models [54]. The calibration parameters of the magnetometers should be estimated for the platform on which they will be eventually used, since the material of the platform can affect the magnetic field. If there is no relative motion between the platform and the sensor, it can be modeled as a constant time-invariant distortion [53]. External magnetic sources may also affect the magnetometer measurements as additional time-varying distortion components if the sensor and the platform are not isolated from the environment [54].

Used parameters

Due to high error rates caused by structural errors of the sensors, the raw measurements are compensated in the preprocessing phase. The calibration parameters were obtained using an offline, evolutionary algorithm-based method, which applies multi-position measurements during the computation [S11].

3.5.2 Windowing

The extraction of feature values is performed in fixed-size segments, which are shifted with constant sizes. To generate a high number of input vectors, small window shifts were used. For hardware implementation, the size of the shifts depends on the available resources and the required response time, since the algorithm updates the movement classes after each window shift, and the reduction of the size of the shifts increases the necessary computation performance.

Both the CPU computation performance and the power resources are limited in IRIS WSN motes, so it is important to minimize the usage of these resources while maximizing the recognition efficiency. The required computation performance and the current draw of the sensors can be reduced if the sampling frequency is decreased.

3.5.3 Feature extraction

Feature types

The used features were chosen by their memory usage, required computation, and possible quantity of information. Due to easy implementation and low memory usage, only time-domain analysis was performed on the signals. Many of the chosen features were previously used for electromyography (EMG) pattern recognition [66], and most of them were not applied previously for movement classification. The used TDFs require only one or two previous measurements, so there is no need to store all the measurement data in the window as it is required for frequency domain analysis. But even standard deviation, which is one of the most frequently used features, requires the storage of the measurement vector in the window, since first the average needs to be calculated. The following TDFs were chosen for this research:

- *Mean Absolute Value (MAV)*: The calculation of the MAV feature can be expressed as follows,

$$\text{MAV} = \frac{1}{N} \sum_{i=1}^N |x_i|, \quad (3.11)$$

where N is the number samples in the window, and x_i are the signal amplitudes at the given index.

- *Willison Amplitude (WAMP)*: The number of amplitude changes of incoming signals within a window, which are higher than a given threshold level. The computation of the WAMP can be expressed as

$$\text{WAMP} = \sum_{i=1}^{N-1} [f(x_i - x_{i+1})], f(x) = \begin{cases} 1, & \text{if } (x \geq th) \\ 0, & \text{otherwise} \end{cases} \quad (3.12)$$

where th is the threshold, which is the peak-to-peak noise level.

- *Number of Zero Crossings (NZC)*: The number of times when the amplitude values cross the zero-amplitude level, and the difference between the values with opposite signs is larger than a defined threshold. The computation of the NZC feature can be represented as given in (3.13).

$$\text{NZC} = \sum_{i=1}^{N-1} [sgn(x_i \cdot x_{i+1}) \cap |x_i - x_{i+1}| \geq th], sgn(x) = \begin{cases} 1, & \text{if } (x \geq 0) \\ 0, & \text{otherwise} \end{cases} \quad (3.13)$$

- *Number of Slope Sign Changes (NSSC)*: The number of direction changes, where among the three consecutive values the first or the last changes are larger than the predefined limit. The computation of this feature can be represented as follows,

$$\text{NSSC} = \sum_{i=2}^{N-1} [f[(x_i - x_{i-1}) \cdot (x_i - x_{i+1})]], f(x) = \begin{cases} 1, & \text{if } (x \geq th) \\ 0, & \text{otherwise} \end{cases} \quad (3.14)$$

- *Maximal (MAX) and Minimal (MIN) value*: The highest and lowest measured value in the processing window.

- *RMS*: The calculation of the RMS in a processing segment can be done as

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}. \quad (3.15)$$

- *Waveform Length (WL)*: The cumulative length of the waveform over the time segment, which is calculated by the sum of absolute changes between two measurements:

$$\text{WL} = \sum_{i=1}^{N-1} |x_{i+1} - x_i|. \quad (3.16)$$

Extraction modes

The used input vectors were generated and tested with the use of two TDF calculation modes:

- *Separately used axes (SEP)*: The features are extracted separately for the X, Y, and Z axes of the sensors.
- *Vector magnitude-based (VL)*: The changes in the vector length are used for the computation of the TDFs. The advantages of this feature extraction mode are that three times less features are generated than with the SEP mode, and that it should be less sensitive to slight differences between movements of different subjects, or small displacements of the sensor nodes on the wrists. However, it should not be able to recognize different poses in stationary positions. The magnitude-based feature extraction cannot provide valuable information in the case of the magnetometer measurements, because the magnitude of the magnetic field is constant in ideal situations, thus, any measured distortions are caused by the changes in the indoor environment. Using the other two sensor types, the accelerometer and the gyroscope, this feature extraction mode can provide important information for the classification process. Except the NZC feature, which cannot give helpful information, since the magnitude cannot be negative, all other of the previously described TDF types can be effective.

Feature aggregation

The usage of the separately extracted features for the three sensor axes can result in a very high number of features, which can increase the complexity of the classification algorithm. Also, it can have a negative effect on the recognition efficiency if the subjects do not fix the units correctly to their wrists. A possible solution to both previous problems can be the aggregation (AGG) of the separately computed features. As expressed in (3.17), this can be done by calculating a linear combination of the feature values computed for each axis for a specific feature type.

$$\text{feat}_{AGG} = w_X \cdot \text{feat}_X + w_Y \cdot \text{feat}_Y + w_Z \cdot \text{feat}_Z, \quad (3.17)$$

where feat_{AGG} is the aggregated feature value, feat_X , feat_Y , and feat_Z are the extracted features

for each axis, and w_X , w_Y , and w_Z are the corresponding weights.

3.5.4 Dimension reduction

The LDA method was used to perform dimensionality reduction on the datasets. The result of this method is a matrix of parameters, which must be multiplied with the feature vector to get the inputs of the classifier. Thus, it requires only multiplication and summation for its implementation.

3.5.5 Classification

In this research seven possibly applicable classification methods were chosen and tested:

- NCC,
- MLP,
- NBC,
- OvO SVM,
- OvA SVM,
- k-NN,
- CT.

3.5.6 Performance evaluation

Altogether 340 datasets were constructed using different combinations of used sensor types, TDF calculation modes, processing window sizes, and sampling frequencies.

The cost of the system can be decreased by decreasing the number of used sensor types, but in recognition efficiency their fusion can result in a drastic improvement. In order to explore the effect of the used sensor types in the application, seven sensor combinations were defined, since the three sensor types can be used alone, in pairs, and together. The SEP and AGG feature extraction modes were tested for all seven sensor combinations, while the VL mode was used only for the accelerometer and the gyroscope alone, and their data used together, since, as described in Subchapter 3.5.3, the magnetometer data cannot provide valuable information using this feature extraction mode. Thus, 17 combinations were constructed using the applied sensor types and feature extraction modes.

The use of large processing windows can increase the required computation, and it can make harder the detection of transitions between activities. Since one of the goals of this research is to explore the recognition efficiency using processing windows in millisecond range, the following window width and shift pairs were tested: 80ms width and 40ms shift; 200ms width and 40ms shift; 400ms width and 80ms shift; 800ms width and 80ms shift.

The necessary computation can be lowered by decreasing the sampling frequency, but it can have a negative effect if any important spectral components disappear. The spectral analysis of the obtained measurements shows, that in the case of the accelerometer and the gyroscope, the highest frequencies of the dominant spectral components are below 15Hz,

while in the case of the magnetometer data, no higher components can be noticed above 5Hz. To find the optimal setup, where the chosen TDFs can be still effective, datasets were generated using five sampling frequencies: 25Hz, 50Hz, 75Hz, 100Hz, and 125Hz. The data for the four lower frequencies were obtained by downsampling the measurement data collected with 125Hz sampling frequency.

Data from five of the nine subjects were used for the training of the classifiers, while the data from the remaining four subjects were tested as unknown inputs for the validation of the trained classifiers. All six classification techniques were tested for all datasets with and without dimension reduction. No results could be achieved using the NBC without LDA, since some classes have features with zero variance.

In this study, both the OvA and the OvO methods were tested and used for comparison in the case of the SVM classifier.

The k-NN classification algorithm was tested with 1 to 10 neighbors. Analyzing the efficiencies on validation data, without dimension reduction a convergence (97%) can be noticed at 1-2 neighbors in almost 55% of the setups, while other setups mostly converge at 3-4 neighbors. With LDA 1-4 neighbors are needed to achieve convergence as well, but in most cases 4 neighbors are necessary.

The training of the MLP was tested using 1 to 15 neurons in the hidden layer. The 70% of the training data were used as training inputs, and 30% as validation inputs for the training method. The validation datasets were used as unknown inputs for testing the efficiency of the classifier. Hyperbolic tangent sigmoid transfer function was used in the hidden layer, while the neurons in the output layer were created using the linear transfer function. Due to its lower memory requirements and higher training speed, the scaled conjugate gradient method was used for training. The results show that in both cases (with and without using LDA), at least 9 hidden layer neurons are needed to achieve convergence (97%), and 9-12 neurons were required in more than 70% of the setups. It can be also noticed, that without dimension reduction the distribution of the converge points is equal, while with LDA more setups converge at 9-10 neurons.

In the further comparison, the setup with the highest recognition rate on unknown samples for both the k-NN and the MLP algorithms was used.

Efficiency comparison of the classification methods

Table 3.3 summarizes the average rankings of the thirteen classification methods on training and validation data, and on weighted overall efficiencies. Since it is important to classify both the known and the unknown data correctly, the weighted efficiency was calculated using the sum of the achieved recognition rates on known and unknown data, but the efficiency on validation data was used with a double weight. The average ranking was computed using the ranking order of the methods for each of the 340 setups. The average efficiencies are also presented in Table 3.3. Comparing the rankings on validation data, it

can be stated, that the MLP and the LDA-MLP methods are the most powerful classifiers. The MLP was the best in almost 48% of the datasets, and its average ranking is 2.85, while the average ranking of the LDA-MLP is 3.13. The NCC method achieved the worst results with an average of 10.67, but the LDA-CT, CT and OvO SVM methods had also poor results with a ranking above 9. The results obtained only on training data show, that the CT and the LDA-CT provide the highest results, with an average recognition rate of 96.58% and 94.46%, respectively. They are followed by the LDA-k-NN (89.46%) and the k-NN (86.09%) algorithms. These classification techniques are designed to best fit on training data, but are not too efficient on unknown data. The MLP, which proved to be the best method in the case of validation data, provided 82.00% efficiency on known datasets, and was fifth in the rankings. Analyzing the overall recognition, it can be seen, that the LDA-k-NN is the best classifier with an average ranking of 2.89. This method is followed by the MLP (3.62) and the LDA-MLP (4.36).

Table 3.3: Average ranking and efficiency of different classification methods on different data types.

Method	Average ranking on training data	Average ranking on validation data	Average ranking on weighted overall efficiencies	Average efficiency and standard deviation on training data	Average efficiency on and standard deviation on validation data
CT	1.03	9.01	5.84	96.58% ±3.09%	58.91% ±8.53%
OvA SVM	11.22	9.86	10.61	44.97% ±29.77%	38.65% ±24.17%
OvO SVM	8.79	8.37	8.45	70.36% ±18.75%	58.41% ±14.19%
NCC	11.76	10.67	11.49	63.90% ±13.71%	56.46% ±13.42%
k-NN	4.69	6.96	6.28	86.09% ±7.81%	63.76% ±9.96%
MLP	5.63	2.85	3.62	82.00% ±10.58%	70.45% ±8.82%
LDA-CT	2.09	9.39	6.16	94.46% ±3.72%	60.33% ±10.62%
LDA-OvA SVM	10.28	7.95	9.11	72.30% ±15.42%	62.47% ±13.80%
LDA-OvO SVM	7.44	4.24	5.73	78.13% ±14.26%	67.05% ±12.49%
LDA-NCC	10.15	6.87	8.37	72.15% ±15.87%	62.83% ±13.93%
LDA-k-NN	3.32	4.46	2.89	89.46% ±8.08%	67.03% ±11.42%
LDA-MLP	6.49	3.13	4.36	80.84% ±10.74%	69.33% ±9.53%
LDA-NBC	8.10	7.21	8.08	77.16% ±13.66%	64.03% ±11.64%

Rates for the 340 datasets when the tested classification techniques performed better without LDA, and the average rate of differences are tabulated in Table 3.4. It can be observed, that the LDA-based dimension reduction has in overall a slight negative effect on the efficiency of the MLP. It decreases the efficiency in around 70% of the datasets, but the differences are not significant. Also, very small differences can be noticed for the CT, but the dimension reduction decreases the ability to recognize known data for almost all setups, while in around half of the datasets it increases the overall efficiency and the recognition rate

on validation data. The LDA method has a very positive effect on the other classification techniques. The most significant improvement was achieved with the NCC method, for which the application of dimension reduction increased the recognition rates in average by 10%. The obtained efficiencies were also higher in around 87% of the setups for all three compared result types. For the other three algorithms, higher classification rates were achieved in around 60-70% of the datasets both on training and validation data. The highest effect can be noticed on the OvA SVM, since without dimension reduction almost 37% lower efficiencies were obtained for both known and unknown data.

Table 3.4: Effect of LDA-based dimension reduction on the tested classification techniques.

Method	Higher results on training data	Higher efficiency on validation data	Higher weighted overall efficiency	Average rate and standard deviation on training data	Average rate and standard deviation on validation data
CT	99.41%	50.88%	57.94%	2.29% ±1.52%	-1.31% ±10.54%
OvA SVM	40.00%	38.53%	40.59%	-36.85% ±42.1%	-36.45% ±40.76%
OvO SVM	41.18%	25.29%	33.82%	-8.81% ±23.3%	-11.83% ±18.98%
NCC	13.24%	13.82%	12.35%	-10.44% ±12.93%	-9.44% ±12.18%
k-NN	27.65%	30.29%	22.65%	-3.59% ±5.51%	-4.29% ±9.13%
MLP	75.88%	64.71%	67.35%	1.54% ±3.67%	1.86% ±4.61%

Efficiency comparison of the tested sampling frequencies and processing window sizes

The further comparison of the results, achieved with different sampling frequencies and window sizes, was done using the best achieved overall weighted efficiencies.

The results show, that using the five tested sampling frequencies, the average difference between the highest and lowest efficiencies is 6.74% ±8.47% for training data, and 6.83% ±6.45% for validation data. The impact of increasing the sampling frequency is almost the same for the four different processing window sizes, but it has different effect on the 17 combinations of extraction modes and used sensors. Analyzing results on validation data, larger differences can be noticed when the magnetometer is used alone. In the case of the SEP mode, the difference between the largest and smallest efficiency is 3-7%, and the recognition rate is decreasing with the increasing of the sampling frequency. The other setups provide almost constant efficiency or a rising tendency by increasing the sampling frequency. The AGG setup provides differences between 2.5% and 4.5% using only the magnetometer data, and around 3% for the data of the angular velocity sensor. Higher differences can be also observed when the SEP feature extraction is performed on the fused data of the magnetic sensor and the gyroscope (3-5.7%), when the AGG features are applied on the data of the magnetometer and the accelerometer together (2.5-8%), and when the data of the accelerometer and gyroscope are used together and VL-based feature extraction is

done (3.2-6%). The other setups provided below 2% differences.

The size of the processing window width has a more significant effect on recognition rates, since the larger windows always result in higher efficiency. In overall, the highest classification efficiencies are higher than the lowest rates for $13.24\% \pm 6.34\%$ on training data, and $28.1\% \pm 14.48\%$ on validation data. The improvements do not differ greatly for different sampling frequencies, but they are more significant in the case of the 17 different combinations of sensors and feature extraction modes. Especially high differences on validation data can be noticed for the three setups when the VL-based feature computation was used: gyroscope – 27.6-29.2%, accelerometer – 18.7-27%, and the gyroscope and the accelerometer together – 26-28.6%. The lowest improvements can be observed in the case of the two setups when the three sensors were used together: SEP – 9.7-11.1%, AGG – 7.1-9.5%. The increasing of the window size also has lower effect in the case of the gyroscope when the features are computed using the SEP and AGG methods, 11.5-12.5% and 6.4-12.5% respectively, and when the SEP technique is used on the fused data of the gyroscope and the accelerometer, where the differences are between 10.4% and 12.8%.

Efficiency comparison of the tested feature extraction modes and sensor combinations

The best results for the 17 different combinations in the four different processing window widths can be seen in Figure 3.4. It can be observed, that using only the magnetic sensor with the AGG feature extraction can provide the lowest recognition rates, since with the smallest window size only 39.95% can be achieved, while with even the largest processing window the efficiency increases only to 60.05%. Using the SEP mode, the recognition rates are much higher, 57.03% with the 80ms window and 67.18% with the 800ms window size.

Using only the angular rate sensor provides the highest results with the SEP method: 66.6-80.7%. The VL mode provides smaller classification rates, but the difference decreases by increasing the size of the processing window, since with the smallest window size the difference is 20%, while with the largest window a recognition rate of 78.33% can be achieved, which is only 2.37% lower than with the SEP mode. The number of features was 48 for the SEP mode and 14 for the VL mode, which is a significant difference. Using the AGG extraction mode, for which the size of the feature vector was 16, significant difference to the VL mode can be noticed for the smaller window sizes. The recognition efficiency was 8,5% higher for the 80ms window, and 10,18% for the 200ms window, but for the two larger sizes the VL achieved better results, 2.55% and 9.57% respectively.

Using only the accelerometer, similar results can be achieved as with the gyroscope. For the two smaller windows with the SEP and AGG modes the accelerometer performed lower results, while with increasing the window size, the accelerometer provides higher efficiencies than the gyroscope. For the SEP mode, the differences were 1-3%, but for the two smaller windows with the AGG mode the recognition rates are lower for 3-5%, and higher for the two larger windows for 7%. With the VL-based feature vectors the

accelerometer provides better results. Using the 80ms processing window size, the difference was around 10%, but the difference decreases, and was only 1% for the 800ms window.

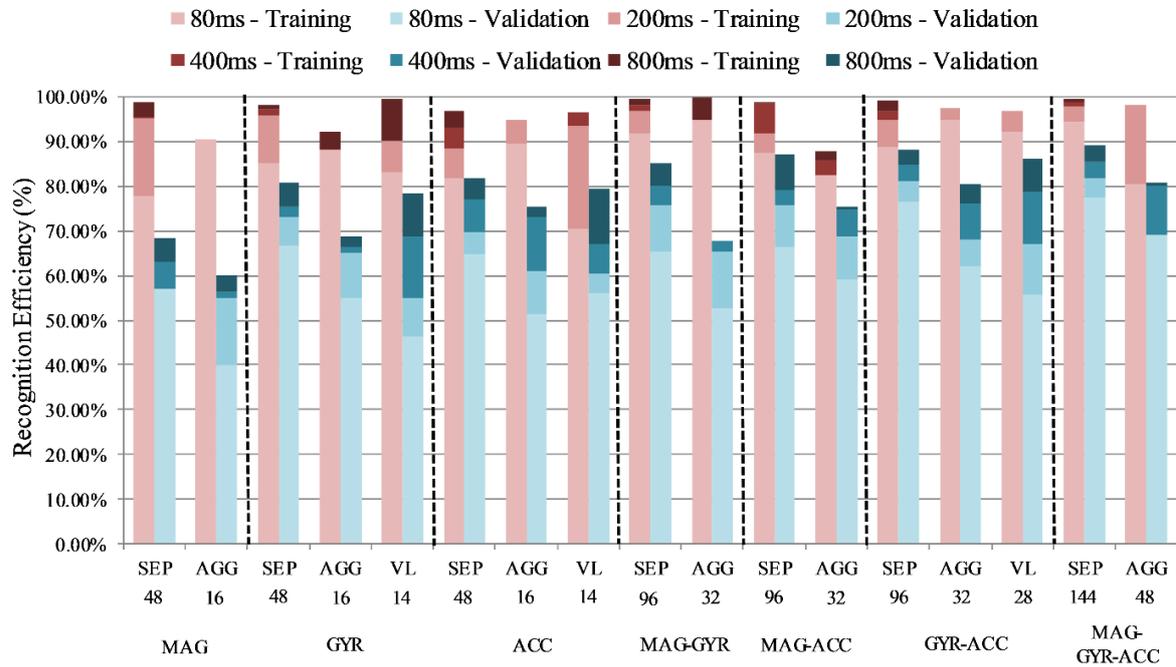


Figure 3.4: Achieved classification efficiencies on training and validation data using different processing window sizes. The horizontal axes show the feature extraction mode in the first row, the required feature numbers in the second row, and the used sensor types in the third row.

The usage of the magnetometer itself cannot provide usable results, but it can improve the performance of the inertial sensors, since the largest classification rates are 85.03% and 87.2% respectively. In the case of the gyroscope, in average, the results were improved for $3.26\% \pm 3.59\%$ for the SEP mode, while with the accelerometer it provides an improvement of $5.11\% \pm 3.19\%$ for the SEP, and $7.45\% \pm 7.44\%$ for the AGG mode. For the setup where the data from the magnetometer and the gyroscope were fused, and the AGG feature extraction mode was performed, in average the results were even slightly lower than when the data from the gyroscope was used alone.

The highest recognition rate on validation data, 89.14% (99.48% on training data), was reached using all three sensor types with the SEP feature extraction in the largest processing window. This setup requires the usage of 144 features. With the same extraction mode, but without using the magnetic sensor, 87.96% classification efficiency can be achieved on validation data, and 98.91% on training data, with a required feature number of 96. By decreasing the size of the processing window, the classification rate significantly decreases, but even with the smallest window, an efficiency of 77.32% can be achieved with, and 76.5% without the magnetometer. The difference between these two setups is a little above 1% in

efficiency, but the number of features, the energy consumption, and the cost are all increased if the magnetic sensor is added to the system. Similar differences can be noticed with the AGG extraction mode also.

The setup where the features were computed using the VL-based extraction, and the data from the angular velocity sensor and the accelerometer were used together, also proved to be very useful. The feature vectors consisted of 28 different features, and with the largest processing window the recognition rate was 86.17%. This extraction mode fails when the processing windows are small, since the efficiency with the 80ms size was only 55.8%, and the AGG-based features provide higher efficiencies in these cases.

Training time comparison of the classification methods

Training time is not a crucial factor for the implementation of a classifier, but it can prove to be very important, especially when different combinations of features should be tested. To generate comparable data, all trainings were done on the same PC with the next characteristics: Intel core i7 3.5GHz processor, 16GB RAM, GeForce GTX 770 video card.

The computation of the LDA matrices proves to be very fast, and even for the largest setup, which contains 144 inputs, less than 1.8s is required.

The k-NN method does not require any training, since it uses the entire dataset for the classification. The shortest, longest, and mean training times for the other classification methods are summarized in Table 3.5. It can be stated, that the most time consuming from the tested classification methods is the OvA SVM algorithm, since the training of the larger setups can last for more than 2 hours, but even the shortest time was almost 1 minute. The OvO SVM method proves to be much faster, but the longest time is still above 1 hour, while the shortest is 17s. The dimension reduction has a significant impact on the SVM-based methods, since it decreases the training time by $93.51\% \pm 11.4\%$ for the OvA, and by $92.45\% \pm 10.79\%$ for the OvO method. Beside the high reduction in training time, caused by the LDA, the longest required intervals are still too high for both methods. The training of the CT method requires between 0.37-15s, and the LDA method does not reduce the training time for all setups, but the longest training was three times shorter than without the dimension reduction. The computation of the parameters for the NCC classifiers is very low for low dimension setups, but for the largest setups it can last for even 25s. The effect of the LDA can be noticed only at the larger setups, and it reduces the maximal time to 2s. The training of the LDA-NBC classification method, similarly to the LDA-NCC, lasts between a few hundredths and 2s. The training of the MLP classifiers is also very time-consuming. The longest interval using 10 hidden layer neurons was 1331.6s. Besides, that even the length of only one training is long, to find the optimal setup, multiple trainings are required with different neuron numbers in the hidden layer. This significantly increases the required training time. The LDA-based dimension reduction has a significant effect on this classification method, since it reduces the longest training time to 97.83s, and in average it

reduces the training time by 48.43% \pm 34.92%.

Table 3.5: Smallest, highest, and mean required training times of the tested classification methods.

Method	Shortest training time	Longest training time	Mean training time
CT	0.37s	15.01s	3.41s
OvA SVM	57.64s	7415.20s	3352.70s
OvO SVM	16.99s	3791.40s	1414.70s
NCC	~0s	2.12s	0.24s
MLP	10.16s	1331.60s	84.81s
LDA-CT	0.59s	5.30s	1.86s
LDA-OvA SVM	8.28s	2576.20s	194.2s
LDA-OvO SVM	2.43s	860.70s	52.38s
LDA-NCC	0.03s	24.90s	1.64s
LDA-MLP	6.36s	97.83s	26.71s
LDA-NBC	0.06s	2.18s	0.34s

Memory requirement comparison of the classification methods

The required space for the implementation of a classifier is a very important factor, since microcontroller-based systems have limited amounts of memory.

The required number of parameters for the implementation of the NBC, the NCC, and the MLP classifiers can be calculated using the number of features and classes. The number of hidden layer neurons is also needed in the case of the MLP-based methods. In the case of the k-NN, the number of samples in the classes is required, since the algorithm uses the entire feature set to determine the class. The required memory for the SVMs and the CTs cannot be calculated as a function of the number of features and classes, because the number of necessary support vectors in the case of SVMs and necessary nodes in the case of CTs differs. For comparison, the required memory spaces were calculated in bytes (1 floating-point number is equal with 4 bytes).

The LDA projection matrices have 10 rows, because 11 classes are used, and the number of columns is equal to the number of features. If the number of features is less than 10, the number of rows will be equal to the number of features.

The training of the NCC was performed by calculating the mean values of different features for each class, and the highest and smallest feature values were also needed for normalization when the dimension reduction was not used.

For the implementation of MLPs, input ranges, weights and biases are needed. The input ranges consist of the highest and lowest values for all inputs, and are used for normalization. Two weight matrices are needed to connect the input layer with the hidden layer, and the hidden layer with the output layer. The first consists of $num_{HiddenLayerNeurons} \cdot num_{InputLayerNeurons}$, while the second of $num_{OutputLayerNeurons} \cdot num_{InputLayerNeurons}$ weights. Bias values are used in all

neurons of the hidden and the output layer. For comparison, based on the convergence in efficiency, 10 hidden layer neurons were used for the computation of the required memory.

The training of the NBC results in a $num_{Classes} \cdot num_{Features}$ sized array of parameter pairs, where the first parameter is the mean deviation, and the second is the standard deviation.

The memory requirements of the five determinable methods can be seen in Figure 3.5. It can be observed that they do not differ significantly. Considerable differences can be noticed only with a small number of features, e.g. with using 80 features, all methods require around 4kbytes of parameters, but with only 10 features the LDA-MLP needs around 1.5kbytes, while the NCC only 0.5kbytes, which is three times lower. Generally, the LDA-NCC needs the least memory space, only the NCC needs less when the number of features is smaller than 40.

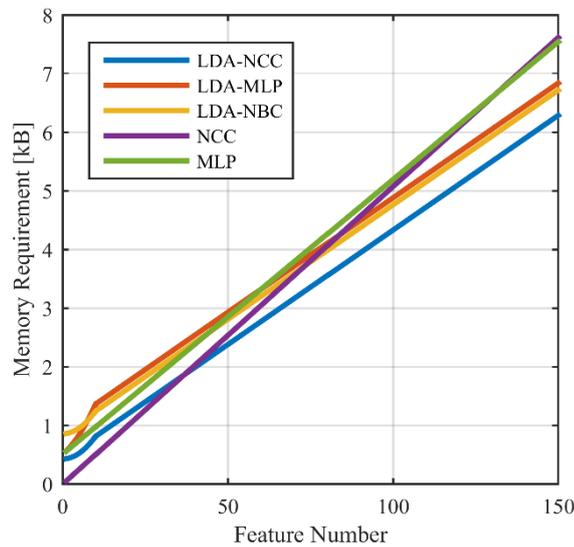


Figure 3.5: Memory requirement of the classification methods with determinable memory consumption.

The k-NN method is a very memory demanding method, since the entire database of features is needed for its implementation. In this research, more than 13000 feature vectors were used even in the smallest setups, which would result in more 760kB memory space for a feature number of 15.

The highest and lowest required memories for the CT and SVM-based methods can be seen in Table 3.6.

The implementation of the CTs requires the number of nodes (16-bit integer), parents (one 16-bit integer per node), children (two 16-bit integers per node), cut points (one floating-point number per node), cut types (one Boolean value per node), and cut predictors (one 8-bit number per node). Analyzing the results, it can be stated, that the required number of nodes and the classification efficiency are inversely proportional. As shown in Table 3.6, the achieved smallest needed memory space is 2.03kB, but high deviations can be noticed,

and for the setup with most required nodes more than 83kB of storage is needed. The LDA has a negative effect on the CT for all setups, and even the lowest required memory is 6.67kB. In average the LDA increases the required memory space by $60.44\% \pm 42.64\%$.

Table 3.6: Highest and lowest memories required for the implementation for the CT, OvA SVM, and the OvO SVM, with and without LDA-based dimension reduction.

Method	Lowest required memory	Highest required memory
CT	2.03kB	83.45kB
LDA-CT	6.67kB	84.86kB
OvA SVM	652.36kB	27458.01kB
LDA-OvA SVM	213.85kB	2358.40kB
OvO SVM	369.30kB	17913.09kB
LDA-OvO SVM	106.16kB	1176.95kB

In the case of the SVM-based methods, due to the used 11 classes, the OvA method needs 11 support vector sets, while $\text{numClasses} \cdot (\text{numClasses} - 1) / 2$ sets are needed for the OvO, what means 55 sets for the used 11 classes. The support vector sets are made up of different numbers of support vectors and a bias value. The dimension of each support vector is equal to the number of features, and they also include an alpha value. The obtained results show that both the OvA and OvO methods require a very high number of parameters for implementation, and thus, are not suitable for application in the developed system. The required memory space is less for the setups with higher efficiency rates, and it decreases by increasing the size of the processing window, since the classification rates increase. The lowest memory requirement, as shown in Table 3.6, was 652.36kB for the OvA mode, and 369kB for the OvO mode. In some setups, it can be even above 20MB using the OvA mode. The LDA has a very positive effect on the SVM classification algorithm, since it greatly decreases the required number of support vectors. The tested dimension reduction method decreases the number parameters for all setups in the case of the OvA SVM method, with an average of $55.91\% \pm 27.03\%$, while for the OvO SVM it reduced the memory consumption for 65.29% of the setups.

3.5.7 Comparison of selected TDFs with FDFs and TDFs with high memory requirements

To explore the capabilities of the applied TDFs, it was reasonable to compare the achieved results with recognition rates obtained with FDFs used in the literature. The following FDFs were utilized in the feature sets: spectral entropy, spectral energy, magnitude of largest peak, frequency of largest peak, median frequency, DC component, median power, and principal frequency. Two TDFs, which require the storage of the measurement vectors for their computation, were also added to the datasets: standard deviation and correlation between

axes. Feature extraction was performed on the sensor axes separately and on the magnitude, and the aggregation-based feature reduction was also applied. Classification was done using the MLP classifier, which earlier proved to be the most powerful method.

The obtained results show, that the applied TDFs have better performance in around 60% of the datasets in the case of the training data, while the rates are nearly equal on validation data. The rates, when the TDFs perform better on training data, are nearly equal for both different sampling frequencies and different processing window widths. In the case of the validation data, the rates show a rising tendency when the sampling frequency or the size of the window is increased. With the smallest frequency or window size, TDFs give better results in around 40% of the datasets, while this rate is almost 60% with the largest frequencies or windows. Since the number of measurements in the processing window increases both with increasing the sampling frequency or the size of the processing window, this is a significant result, because the chosen TDFs do not require the storage of the measurement values in the window.

Table 3.7 summarizes the obtained results with MLPs using TDFs and FDFs when the highest sampling frequency, 125Hz, was applied. The used abbreviations are the next: TR – training data, VA – validation data, TD – time-domain, FD – frequency domain.

Table 3.7: Achieved classification efficiencies (%) applying extraction based on TDFs and FDFs.

Used sensors	Extraction mode	Feature type and number	Processing window width							
			Dataset							
			80ms		200ms		400ms		800ms	
			TR	VA	TR	VA	TR	VA	TR	VA
MAG	SEP	TD-48	74.54	52.43	85.76	59.95	91.27	63.23	96.36	69.97
		FD-60	80.86	44.76	84.89	50.45	88.35	57.41	95.98	58.31
	AGG	TD-16	67.38	48.50	73.14	55.40	74.15	55.25	77.37	58.50
		FD-20	61.65	46.45	68.20	53.88	71.50	56.93	76.04	59.62
GYR	SEP	TD-48	69.93	62.47	84.09	70.90	84.25	72.86	94.05	76.39
		FD-60	77.61	69.98	82.52	76.29	91.14	78.76	90.14	82.83
	AGG	TD-16	62.56	58.03	71.32	65.37	68.51	63.86	74.71	69.21
		FD-20	63.15	60.30	68.97	65.42	70.10	69.55	72.63	71.42
	VL	TD-14	56.82	52.60	62.19	58.89	74.81	68.35	84.61	79.39
		FD-18	56.54	52.34	65.60	61.65	73.94	68.56	80.82	74.77
ACC	SEP	TD-48	72.90	65.77	84.60	73.83	87.67	76.61	90.75	82.12
		FD-60	76.07	65.08	83.00	72.55	88.17	76.90	94.30	83.11
	AGG	TD-16	66.96	63.62	76.42	69.34	77.35	72.95	79.39	74.28
		FD-20	57.74	52.48	67.04	58.94	68.84	66.83	72.34	70.96
	VL	TD-14	53.47	53.51	64.08	62.78	77.60	73.50	82.46	79.47
		FD-18	54.22	52.89	62.74	60.82	71.80	72.23	79.12	80.31
MAG	SEP	TD-96	89.80	67.53	94.06	70.71	96.70	72.92	83.85	74.25
GYR		FD-120	92.08	64.55	93.92	70.78	95.98	74.81	99.57	79.06

	AGG	TD-32	79.68	61.44	83.05	64.51	84.76	67.53	85.34	70.99
		FD-40	80.01	68.63	83.79	72.38	84.97	74.95	84.94	77.95
MAG ACC	SEP	TD-96	84.35	68.32	86.41	74.34	96.20	77.25	98.43	83.98
		FD-120	88.11	58.10	92.41	64.63	95.67	69.38	98.87	73.36
	AGG	TD-32	75.77	67.86	83.17	71.45	85.05	73.78	86.28	74.79
		FD-40	72.68	56.81	77.01	66.84	80.64	66.48	83.88	70.73
GYR ACC	SEP	TD-96	88.13	76.60	88.80	80.57	95.50	83.25	98.33	84.69
		FD-120	86.13	77.94	89.27	81.64	91.01	84.63	98.63	87.38
	AGG	TD-32	74.77	68.64	80.80	74.99	81.24	76.29	81.67	78.00
		FD-40	69.72	66.61	77.59	74.20	80.38	75.79	80.76	76.94
	VL	TD-28	67.46	61.31	76.17	71.07	84.08	78.45	91.51	85.61
		FD-36	67.45	62.95	76.90	71.02	80.88	75.10	89.81	82.74
MAG GYR ACC	SEP	TD-144	93.98	74.23	93.98	79.11	97.16	81.91	98.70	84.22
		FD-180	93.34	68.63	95.73	74.75	98.30	80.71	98.98	82.51
	AGG	TD-48	80.94	69.59	87.01	76.04	88.14	75.95	90.33	78.53
		FD-60	82.32	70.27	84.63	74.83	86.63	75.85	87.42	77.65

The recognition rates achieved with FDFs, just like with TDFs, increase with the increasing of the sampling frequency or the processing window width. The highest classification efficiency on validation data, 87.38%, was achieved using the gyroscope and the accelerometer data together, and applying the SEP extraction mode. It should be noted, that, as it can be seen in Table 3.7, the number of applied features is considerably higher in the datasets based on FDFs. The average difference between efficiencies obtained on validation data utilizing TDFs and FDFs is around 3%, while the highest differences, around 11%, can be noticed when features are extracted from the measurements of the magnetometers.

Analyzing the classification efficiencies using different feature extraction modes, it can be concluded, that the aggregation-based feature reduction is also useful when FDFs are applied. The features computed from the accelerometer measurements provide better recognition rates using TDFs, but both the gyroscope and the magnetometer give even better results with FDFs in the case of the AGG-based extraction. The magnitude-based extraction results in similar recognition efficiencies using TDFs and FDFs.

3.6 Hierarchical-distributed movement classification algorithm

Since the proposed algorithm in Subchapter 3.5, uses together the features extracted on both sensor nodes in the classification process, its implementation requires high energy consuming radio communication for data transfer between the nodes. It is reasonable to split the classification algorithm into a hierarchical approach to get a distributed network, so the nodes can calculate their own movement classes.

The software architecture of the hierarchical-distributed algorithm can be seen in Figure 3.6. Difference compared to the algorithm presented in Subchapter 3.5 can be noticed only

after the preprocessing part, since error compensation, windowing, and feature extraction are done identically. Classification is performed on both motes based on the computed feature vectors. Using the proposed hierarchical-distributed technique, only the computed movement class is needed to be transferred periodically from one mote to the other based on the value of the window shift. The determined classes are combined to get the movement of the entire body and arms. Besides that, using the proposed algorithm less data transfer is required via wireless communication, the classifiers have less input features and output classes, thus, it is more energy-efficient and easier to implement the algorithm on the motes. Adding new devices to the system would be also easier with this approach, since only the class combination stage on the receiving unit needs to be modified, and the size of the feature vector could also become too large in the case of the non-distributed (ND) approach.

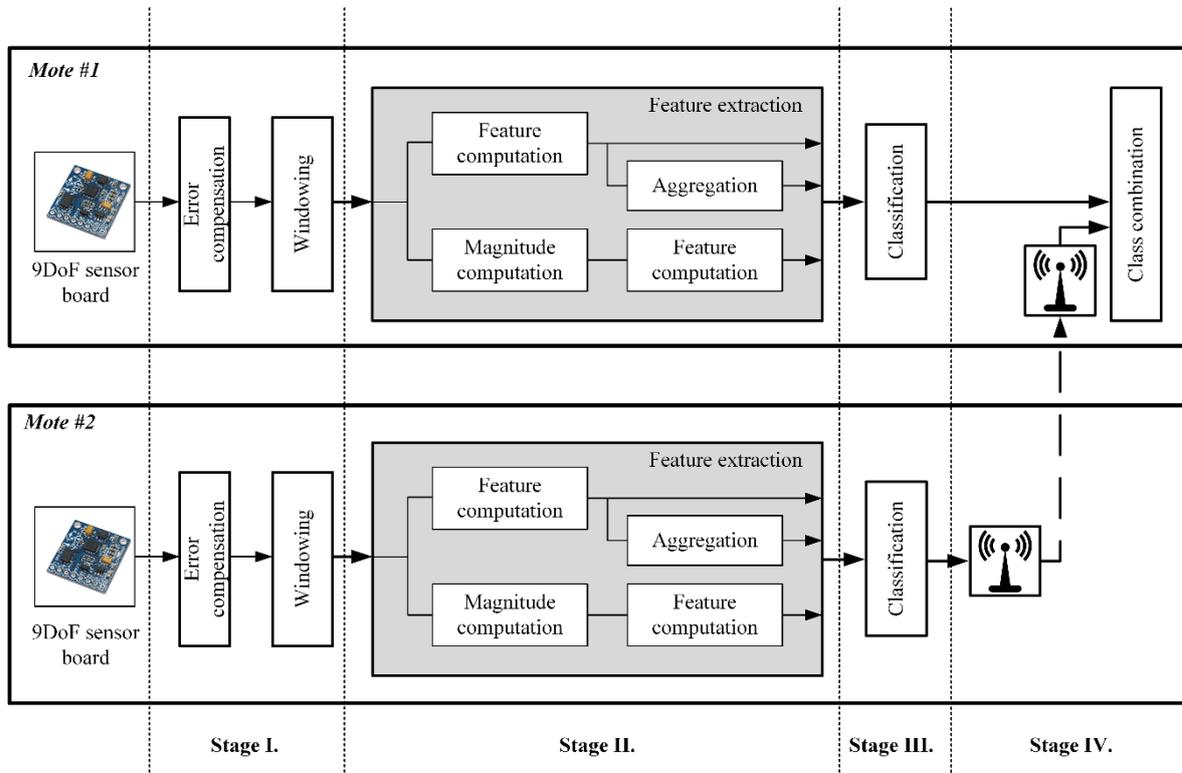


Figure 3.6: Software architecture of the hierarchical-distributed approach.

3.6.1 Applied movement hierarchies and classification

To develop a distributed algorithm in which the two motes can determine their own movement type, some classes can be merged by the role of the arm in the given movement. For example, classes 1 and 6 can be merged in the case of the right arm, because in both cases the right arm is not moving during standing. This way the reduction of the classes can be done in four cases, so the total number of classes can be reduced to seven for both arms. Merging for the left arm can be done for the classes: 1 and 7; 3 and 10; 6 and 8; 9 and 11. For the right arm these cases are: 1 and 6; 3 and 9; 7 and 8; 10 and 11.

Two different approaches were tested for the classification hierarchy. In the first approach (H1), the movements are equally distributed, all of them are on the same level. The first hierarchy can be seen in Figure 3.7.

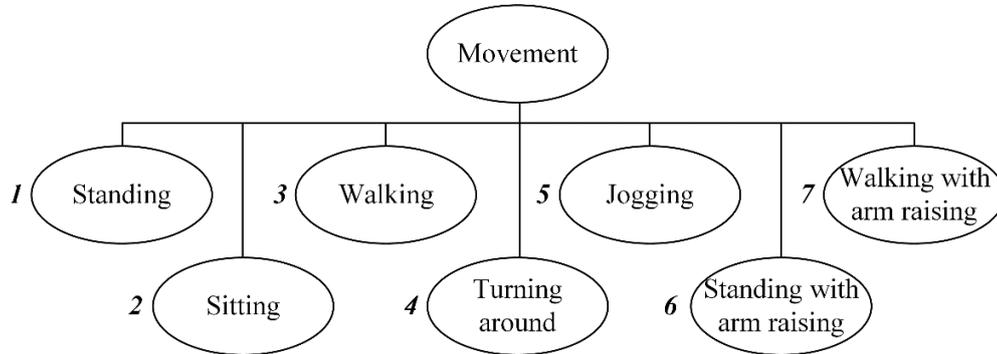


Figure 3.7: The first applied movement hierarchy.

In the second hierarchy (H2), the movements are split into three levels. The classification algorithm uses five distributions to decide which element of the hierarchy matches the actual movement. The second hierarchical approach can be seen in Figure 3.8, and the corresponding distributions (*D*) are:

- *D1*: *I* or *2*,
- *D2*: *a* or *b*,
- *D3*: *c* or *d* or *e*,
- *D4*: *I* or *II*,
- *D5*: *III* or *IV*.

For example, the first distribution *D1* decides that the actual movement is stationary or not. Through these distributions a hierarchical classification can be realized.

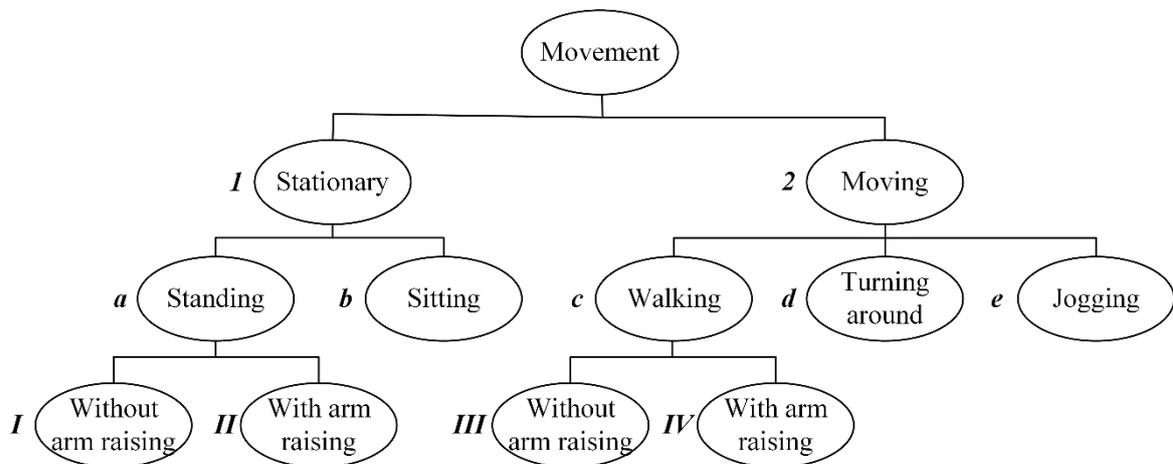


Figure 3.8: The second movement hierarchy approach.

For classification, only MLP neural networks were applied, which showed to be the most powerful classifiers in Subchapter 3.5. In the case of the first hierarchy, one network is applied with seven outputs to determine the class of one arm. Five MLPs, one for each distribution, are required for the second hierarchical approach. The network for D3 has three outputs, while the rest have two. Depending on the outputs of the distributions, two or three networks have to be calculated to get the movement class. The entire feature vectors are used as inputs for all networks.

Class combination is done on the unit which receives the other`s movement class. The combination gives one of the eleven classes as the output, but it gives an unknown class if the classes are not a valid pair.

The effect of the LDA-based dimension reduction was not tested, since the dimension of the output vector in the case of this method is less than the number of classes, thus, using the second approach, most networks would have only one input.

3.6.2 Performance evaluation

To provide comparable results, all 17 combinations of used sensor types and feature extraction modes were tested. Altogether 68 datasets were constructed to examine the effect of the processing window size. The signals measured using the 125Hz sampling frequency were utilized during feature extraction.

Identically as in Subchapter 3.5, data from five of the nine subjects were used for the training of the classifiers, while the data from the rest of the subjects were utilized for validation. During the construction of the training and validation datasets, the data from the merged classes were used in equal quantities. The training and the validation of the classifiers were done using the datasets of the two arms together, thus, the same classifier is later used on both arms.

The efficiency of the MLPs was also tested with 1-15 neurons, and the same configurations were applied as during the testing of the non-distributed approach. Based on the obtained results, the required number of hidden layer neurons to achieve 97% convergence in efficiency on validation data can be seen in Table 3.8. The first hierarchy requires between 6 and 10 hidden layer neurons, while the distributions in H2 mostly need 1 to 4 neurons to achieve convergence.

Table 3.8: Required number of hidden layer neurons for convergence.

Network in the tested hierarchies	H1	D1	D2	D3	D4	D5
Required hidden layer neuron number	6-10	1-3	1-2	2-4	1	1-3

In the further comparison, the setup with the highest classification efficiency on validation data was used.

Efficiency comparison of the two movement hierarchies

The obtained results on training and validation datasets using the H1 and H2 approaches for the different combinations and processing window widths can be seen in Figure 3.9 and Figure 3.10, respectively.

Analyzing the obtained efficiencies, the two hierarchies provide similar results. The second hierarchy is better in nearly 57% of the datasets in the case of training data, while in around 67% of the datasets H1 gives higher rates for validation data. The average absolute difference is 2.14% for training and 1.76% for validation data.

Comparing the achieved classification efficiencies with results obtained with the non-distributed approach, it can be also observed that the ND outperforms both hierarchical approaches, since it provides better results for all datasets both in the case of training and validation data. The average difference is around 25% compared to both hierarchical approaches for both data types, with a standard deviation of $\pm 10\%$ for training and $\pm 7\%$ for validation data.

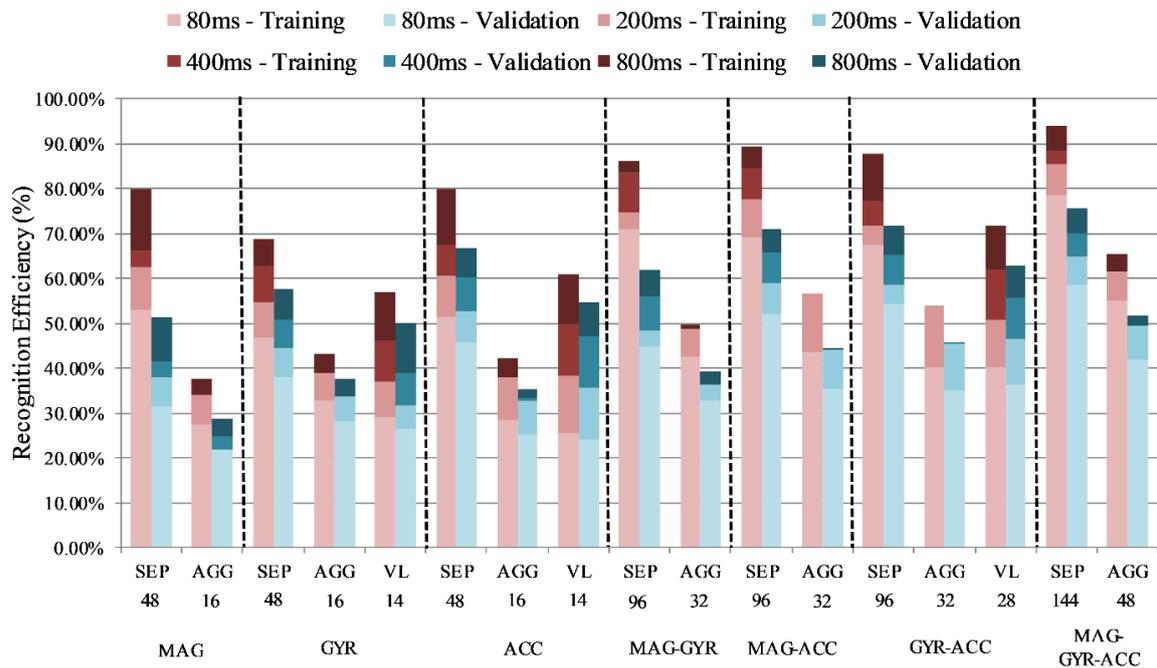


Figure 3.9: Achieved classification efficiencies on training and validation data using different processing window sizes with the H1 approach. The horizontal axes show the feature extraction mode in the first row, the required feature numbers in the second row, and the used sensor types in the third row.

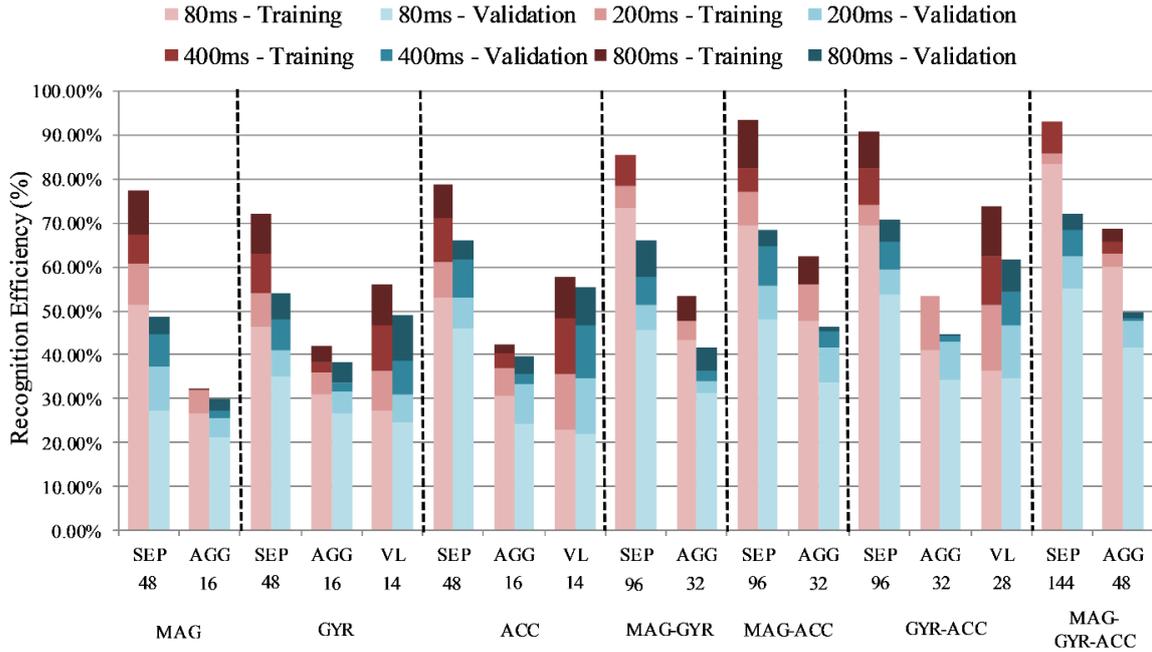


Figure 3.10: Achieved classification efficiencies on training and validation data using different processing window sizes with the H2 approach. The horizontal axes show the feature extraction mode in the first row, the required feature numbers in the second row, and the used sensor types in the third row.

Efficiency comparison of the tested processing window sizes

The increasing of the processing window width results in the increasing of the recognition rates. Similar improvements can be noticed in the case of the two tested hierarchies. The average differences between recognition rates using the largest and smallest window widths are the following: $18.49\% \pm 8.91\%$ (H1) and $18.54\% \pm 9.97\%$ (H2) for training data, $16.18\% \pm 7.06\%$ (H1) and $17.46\% \pm 6.86\%$ (H2) for validation data.

The most significant difference can be noticed for the configurations where the magnitude signals of the ACC are utilized with or without the GYR data. For these configurations higher than 25% increasing can be observed for both training and validation data and for both approaches. The smallest increase can be noticed in the configurations where the MAG or the GYR features are applied, and the aggregation-based extraction is used. For these setups the increase is mostly below 10% for both dataset types.

Efficiency comparison of the tested feature extraction modes and sensor combinations

Analyzing the results on validation data, the lowest efficiencies can be achieved using only the MAG sensor with the AGG extraction mode. The recognition rates are around 30% with the largest processing windows, and slightly above 20% with the smallest windows for both hierarchies. The inertial sensors alone provide the highest results with the SEP mode, which are around 55% using the GYR and around 66% using the ACC in the case of the largest

processing window. Applying the smallest window, only the ACC with the SEP extraction gives higher results than 40%, which is 46% for both approaches.

Adding the magnetic sensor to the inertial sensors increases the efficiencies. The highest improvement, around 10%, can be noticed for all window sizes in the case of both hierarchies when the ACC with AGG features are used, and utilizing the GYR with SEP features only in the case of H2. For the other setups between 2-5% improvement can be noticed. Using the two inertial sensors together provides almost identical results for the two hierarchical approaches. In the case of the smallest window size, the AGG and the VL modes give only around 35%, while with the SEP mode the recognition rates are slightly under 55%. Applying the largest tested processing windows, the highest results, slightly above 70%, can be achieved using the SEP mode, while the VL and the AGG modes provide around 62% and 45%, respectively.

The highest classification efficiencies were obtained with the fusion of the three sensor types. The SEP mode in the largest window provides the best results, 75.76% with H1 and 72.05% with H2. Using the smallest windows, the efficiencies are much lower, 58.43% with H1 and 55.11% with H2. The AGG extraction mode, which applies three times less features, provides only around 50% with the largest window size, and nearly 41% using the smallest window. The AGG-based results are around 5% higher than when the ACC and the GYR are used without the MAG.

Memory requirement comparison of the proposed hierarchies

The required memory for the implementation of the hierarchical-distributed and the ND approaches on one unit can be seen in Figure 3.11. The number of hidden layer neurons during the computation were the following: H1 – 8, D1 – 2, D2 – 2, D3 – 3, D4 – 1, D5 – 2, ND – 10. The results show that for the implementation of H1 less memory is needed than in the case of the ND approach, since the movements need to be classified into fewer classes, which reduces the size of the ANN. It can be also noticed that H2 is the most memory consuming of the three methods, which requires nearly the double size of memory needed for H1.

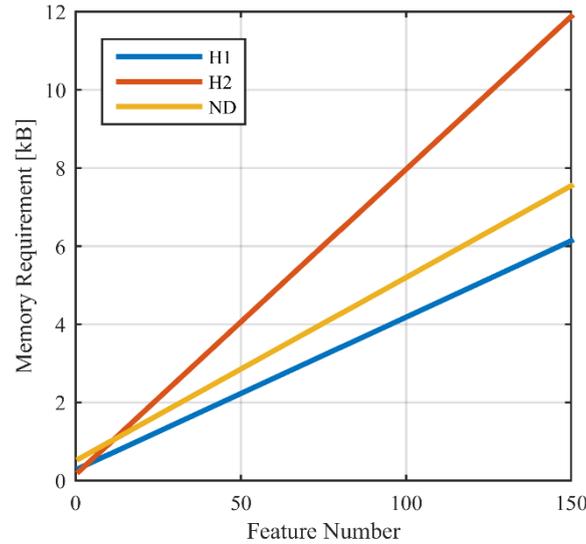


Figure 3.11: Memory requirements for the implementation of the classifiers.

3.7 Comparison of time- and frequency-domain analysis

Since in the proposed algorithms only TDFs with low memory and computation requirements were applied, it is important to compare the capabilities of the two extraction modes on the collected measurement data.

3.7.1 Applied time-domain features

TDFs are computed using the measurement signals from the sensors. Their main advantage is the easy implementation on embedded systems, since there is no need for any transformation. Many features also do not require the storage of the measurement values in the entire processing window. Their disadvantage is that they do not provide direct information about frequency components in the signals. The applied TDFs were the following:

- *MAV*
- *MAX*
- *MIN*
- *NSSC*
- *NZC*
- *RMS*
- *Standard deviation (STD)*: The standard deviation in a processing window can be defined as in (3.18).

$$STD = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N-1}}, \quad (3.18)$$

where \bar{x} is the average amplitude value in the processing window.

- WAMP
- WL

3.7.2 Applied frequency-domain features

FDFs are computed using the amplitude spectrum of the obtained signals, thus, they require transformation to frequency-domain. The transformation in embedded systems is usually performed using Fast Fourier Transform (FFT). Their main disadvantage compared to TDFs is that they require higher computational costs, and thus more energy, which is a very important factor in wearable devices. The used FDFs are the next:

- *Spectral Energy (ENE)*: The spectral energy can be computed as in (3.19).

$$ENE = \sum_{i=1}^M Y_i^2, \quad (3.19)$$

where M is the number of points in the frequency axis, and Y_i is the value of the amplitude spectrum at the given frequency index.

- *Median Frequency (MDF)*: The frequency which divides the spectrum into two regions with equal amplitude.

$$\sum_{i=1}^{MDF} Y_i = \sum_{i=MDF}^M Y_i = \frac{1}{2} \sum_{i=1}^M Y_i \quad (3.20)$$

- *Mean Frequency (MNF)*: This frequency is also called central frequency (f_c), and it can be calculated using (3.21).

$$MNF = \sum_{i=1}^M f_i Y_i / \sum_{i=1}^M Y_i, \quad (3.21)$$

where f_i is the frequency at the given index.

- *Mean Power (MNP)*: The average power of the spectrum.

$$MNP = \sum_{i=1}^M Y_i / M \quad (3.22)$$

- *Peak Magnitude (PKM)*: The highest amplitude in the spectrum.
- *Peak Frequency (PKF)*: The frequency at which the PKM occurs.
- *Variance of the Central Frequency (VCF)*: The variance of the central frequency can be defined using (3.23).

$$VCF = \frac{1}{\sum_{i=1}^M Y_i} \sum_{i=1}^M Y_i (f_i - f_c)^2 \quad (3.23)$$

3.7.3 Performance evaluation

Three different processing window widths were applied: 64, 128, and 256 measurements. These window sizes are different than the previously tested ones, since the FFT algorithm uses power of two number of samples. Using the applied 125Hz sampling frequency, this means nearly 0.5s, 1s, and 2s. In the case of all three window widths the size of the window shift was 32 samples.

The classification was performed using only Multi-Layer Perceptron (MLP) classifiers. The training of the classifiers was tested with 10, 15, 20, 25, and 30 hidden layer neurons. The distribution of data for training, validation, and testing was identical as described in Subchapter 3.5, but the aggregation-based reduction mode was not tested in this comparison.

In the further comparison, the results where the highest recognition rates were achieved on unknown samples were utilized.

The achieved recognition efficiencies on training and validation data per used sensor types and feature extraction modes are summarized in Table 3.9 and Table 3.10, respectively.

The obtained results on the training data show, that the highest classification rates can reach almost 100% using TDFs in the largest processing windows. The recognition rates show a rising tendency for all used sensor and extraction combinations when the window width is increased. In the case of almost all datasets the TDFs provided higher efficiencies than FDFs, and the average difference was around 2% with a $\pm 2\%$ standard deviation. The lowest efficiencies, under 80%, were achieved with the smallest processing windows in the case when the data from accelerometer and the gyroscope sensors were used alone, and the vector length-based feature extraction was applied.

Analyzing the results on the validation data, it can be concluded that the TDFs provide higher efficiencies in most of the setups. The highest obtained classification rates on the unknown data were 91.74% using TDFs, and 88.51% utilizing FDFs. At most of the datasets, a rising tendency can be noticed with the increasing of the window width, but in some cases, usually when the data from the magnetic sensors are applied, the classification rate can even decrease. The most significant improvements can be noticed using FDFs when the magnitude-based extraction is performed on accelerometer signals. When this sensor type is used alone, the recognition rates increase from 78.75% to 88.51%. The highest achieved classification efficiencies applying the smallest window width were 86.72% and 85.45%, using TDFs and FDFs, respectively. Both were obtained utilizing the raw data from the gyroscopes and accelerometers. Adding the extracted features from the magnetometer signals slightly decreases the recognition rates. The lowest efficiencies, under 70%, were achieved using only the data from the magnetic sensors. The time-domain extraction provides higher rates using the computed magnitudes of the gyroscope than utilizing the raw measurements, while the results are opposite applying FDFs. Using only the accelerometer data, the recognition rates are higher using raw measurements in smaller processing windows, while the magnitude-based extraction provides higher efficiencies using the largest window. Adding the magnetometer-based features both improves the recognition rates achieved with the gyroscope and the accelerometer. The classification efficiencies obtained using the gyroscopes and the accelerometers together provide higher recognition rates when the raw measurements are utilized during feature extraction, but the difference decreases for both TDFs and FDFs when the size of the processing window is increased. This is a significant result, since the magnitude-based extraction results in more than three-

times less features than using raw measurements, thus, its implementation requires lower memory space and the classification requires less processing time.

Table 3.9: Achieved recognition efficiencies (%) on the training data using different feature sets.

			Used Sensors									
			Extraction mode									
			MAG	GYR	GYR	ACC	ACC	MAG, GYR	MAG, ACC	GYR, ACC	GYR, ACC	MAG, GYR, ACC
			SEP	SEP	VL	SEP	VL	SEP	SEP	SEP	VL	SEP
Processing window width	64	TDF	93.88	85.88	79.24	91.71	77.43	97.57	96.89	96.10	87.56	98.70
		FDf	88.62	85.29	76.03	88.12	80.90	94.26	95.77	94.00	83.33	98.47
	128	TDF	97.79	94.69	87.78	97.04	90.13	99.39	99.83	99.08	93.60	99.88
		FDf	96.24	89.16	84.15	92.99	85.04	98.91	98.55	97.79	90.86	99.61
	256	TDF	98.72	99.44	92.11	99.85	95.62	99.90	99.95	99.90	99.39	100
		FDf	98.69	98.59	86.56	97.49	93.14	99.72	99.64	99.64	97.88	99.97

Table 3.10: Achieved recognition efficiencies in (%) on the validation data using different feature sets.

			Used Sensors									
			Extraction mode									
			MAG	GYR	GYR	ACC	ACC	MAG, GYR	MAG, ACC	GYR, ACC	GYR, ACC	MAG, GYR, ACC
			SEP	SEP	VL	SEP	VL	SEP	SEP	SEP	VL	SEP
Processing window width	64	TDF	63.46	72.52	74.26	79.01	76.27	77.95	81.70	86.72	81.67	84.50
		FDf	55.82	79.28	70.46	77.30	78.75	78.25	77.78	85.45	79.58	83.56
	128	TDF	70.18	78.21	79.30	85.12	83.33	77.00	81.70	85.18	86.39	83.61
		FDf	65.97	80.36	76.49	82.64	82.67	82.85	85.85	87.49	83.58	85.97
	256	TDF	67.73	82.23	82.81	82.52	83.61	85.60	85.92	91.74	90.88	88.12
		FDf	62.90	78.91	80.95	84.48	88.51	75.64	84.60	87.42	86.01	86.17

3.8 Conclusion

In this chapter, a new movement classification system is presented, which can classify different movements of the body and the arms based on measurements of 9DoF sensor boards attached to the wrists of the subject.

The proposed prototype system uses accelerometers, gyroscopes and magnetometers mounted on WSN motes. The measurement system can be realized in a wristwatch-like unit for real life applications. Data acquisition was done with the help of multiple subjects for eleven movement classes.

Two online classification algorithms are also proposed based on the collected measurements. To reduce computation costs, only time-domain features with low memory requirements were applied. The accelerometer, the gyroscope, and the magnetometer were tested separately, in pairs, and altogether, to investigate the impact of the sensors in the application, and to prevent unnecessary usage of memory and hardware resources, and of course, it can lower the overall cost and power consumption of the system.

The first algorithm uses together the features extracted on the two units during classification. To explore the optimal cost, power consumption, and efficiency, 340 datasets were constructed based on different feature extraction modes, sampling frequencies, processing window sizes, and sensor combinations.

The results show that the recognition rates achieved, using only simple time-domain features, are not affected significantly by the sampling rate, and only slight improvements can be noticed when it is increased. The tested millisecond range processing windows prove to be usable, since above 77% percent efficiency can be reached on unknown data even with the smallest, 80ms, window width, while almost 90% can be achieved with the 800ms window size. It can be concluded from the achieved efficiencies, that the movements of different subjects show high correlation, since the training and validation datasets were constructed of data from different persons. The classification rates on training data can be almost 100%, which is also a very important factor if the application should be trained and used for one person.

The magnetic sensor itself provides very low, 40-67%, recognition rates, but it can significantly improve the performance of the gyroscope and the accelerometer if they are used together. The two inertial sensors alone can provide around 80% applying the largest processing window. The highest efficiencies were achieved when the data from the three sensor types were applied together, but the impact of the magnetometer is very small, since it only increases the recognition rates by 1-2%, while it largely increases the cost, the energy consumption, and the required feature number. For some datasets, adding the features extracted from the magnetometer data can even lower the classification efficiency.

The highest efficiencies were achieved when the separately computed features were used, but they require three-times more used features than the aggregation- and magnitude-based datasets. The inertial sensors can provide 86.17% using the VL-based extraction in the 800ms processing window, but for the smaller window sizes the proposed aggregation-based feature extraction provides higher classification rates.

Seven popular classification methods, the MLP, the NCC, the NBC, the OvA SVM, the OvO SVM and the k-NN, were tested with and without LDA-based dimension reduction. The classifiers were compared by efficiency, training time, and memory requirement for implementation. The obtained results show that the LDA can lower memory consumption and/or training time, but it can also increase classification efficiency of some classifiers. It can be concluded that the highest efficiency can be achieved using the MLP classifier, but

the use of the LDA-MLP is also reasonable due to the slightly lower efficiency, lower memory requirements in the case of high feature numbers, and significantly lower training time. The CT can effectively classify the training data, but its performance is significantly lower for the unknown samples. The very popular k-NN and SVM-based methods showed to be unsuitable for use in this application due to their low efficiencies and high hardware requirements.

The proposed second online algorithm uses a hierarchical-distributed approach during classification, since both units determine their own class, and one of the devices combines the two results to get the class of the entire body and arms. Two movement hierarchies were tested for various feature sets constructed based on different combinations of used sensor types, feature extraction modes, and processing window sizes.

The obtained results show, that the two hierarchies provide similar results, but the non-distributed approach gives much higher results. The average difference is around 25% compared to both hierarchical approaches for both training and validation data. The highest classification efficiencies on validation data, 75.76% using the first and 72.05% applying the second hierarchy, can be achieved using the three sensor types together. The second movement hierarchy has high memory requirements for implementation, but this could be reduced by using simpler classifiers or less features in the distributions.

A comparison was also presented, where the recognition efficiency was compared of datasets constructed based on only time-domain analysis and only frequency-domain analysis. The obtained results show, that the TDFs provide higher recognition rates in most of the setups. The highest recognition rates for all three tested processing window widths were achieved using the raw data from the gyroscopes and accelerometers. The overall highest efficiencies were obtained using the largest window size, 91.74% using TDFs, and 88.51% utilizing FDFs.

The future goals of this research include testing different weights for the aggregation-based feature extraction, finding the features with the most influence, and developing a more complex method for the class combination phase in the hierarchical-distributed approach. The proposed system could also be easily expanded for the detection of falls and epileptic seizures.

Thesis Group 1. I developed novel online algorithms for the classification of human body and arm movements using wrist-mounted accelerometers, gyroscopes and magnetic sensors. The algorithms are implementable on resource constrained embedded systems.

Thesis 1.1. I developed a prototype measurement system to record human movements and collected measurement data for training and validation of the classifiers. I used two nine degrees of freedom (9DoF) sensor boards mounted on WSN motes, which were attached to the wrists of the subjects.

Thesis 1.2. I developed a new online movement classification algorithm where the features extracted on multiple units are used together in the classification stage.

I showed that the use of only time-domain features (TDFs) with low memory requirements is sufficient to obtain high classification efficiencies in movement recognition systems using inertial and magnetic sensors.

I showed that there is a high correlation between the movements of different subjects, since the classifiers provided very high recognition rates on measurements of subjects, which were not used during the training.

I showed that millisecond domain in processing window widths is sufficient for efficient recognition of human body and arm movements.

I proposed an aggregation-based feature reduction method for sensors with multiple measurement axes, which results in three-times less features than when the features computed for each sensor axis are used as inputs of the classifiers. The aggregation is done by calculating a linear combination of the feature values computed for each axis for a specific feature type. The aggregation method can provide useful classification efficiencies. This reduction method can also help the system to be less sensitive to differences in orientations of the sensors on the arms.

I showed that adding a new sensor type can in some cases only slightly increase or even decrease the overall recognition rate of the system. With testing the possible sensor types in different combinations, an optimal configuration can be found, which can lower the overall cost and power consumption of the system.

I showed that the Multi-Layer Perceptron (MLP) neural networks are the optimal solution for online algorithms in similar applications, and that some popular methods (SVM, k-NN) are not suitable for online use in resource constrained embedded systems.

I showed that the use of LDA-based dimension reduction on the feature sets can improve recognition efficiency, training time, and memory requirements for implementation of different classifiers.

I showed that using only simple TDFs even higher recognition rates can be achieved for some setups than when only features extracted using frequency-domain analysis are applied.

I showed that the proposed aggregation-based feature reduction can be also effectively applied in the case of frequency-domain features (FDFs).

Thesis 1.3. I developed a hierarchical-distributed algorithm for online human movement classification, where the multiple units determine their own movement classes, and one unit combines the resulted classes to determine the movement class of the entire body. This algorithm requires less computation because of the smaller classifiers and less wireless communication, which improves energy efficiency. The addition of new units to the system would be also easier using this approach. I applied two different movement hierarchies, which provide similar results.

Chapter 4

Real-time vehicle classification using a single magnetometer

4.1 Introduction

Automatic vehicle detection technologies play an important role in Intelligent Transportation Systems (ITS). They provide traffic counting, presence detection, speed monitoring, headway measurement, vehicle classification, and weigh-in-motion data. Vehicle count and classification data are important inputs for traffic modelling, pavement design, transportation planning, emission/pollution estimation, and traffic control. These systems can be also useful in toll systems and parking lots.

Vehicle detection systems can be classified into intrusive and non-intrusive technologies [67]. Intrusive detector technologies include inductive loops, pneumatic road tubes, and piezoelectric detectors. They are mounted directly on the pavement surface, in saw-cuts or holes in the road surface, by tunneling under the surface, or by anchoring directly to the pavement surface. Non-intrusive detector technologies include active and passive infrared, microwave radar, ultrasonic, passive acoustic, and Video Image Processing (VIP)-based technologies. These systems are installed above the roads or on the side of a roadway, and they can monitor one or multiple lanes. Magnetic sensor-based units can be included into both technologies, since they can be mounted into the pavement surface or on the side of the roads.

The most commonly used technologies are inductive loops and cameras. VIP systems can monitor multiple lanes, and are easily configurable, but their installation and maintenance are costly, and their performance is affected by many factors, such as fog, rain, snow, vehicle shadows, and day to night transition [68-70]. Inductive loops provide presence detection using the metallic content of the vehicles when they are passing above the sensor. Two loops are needed for speed measurement, heading direction estimation, and vehicle classification. Classification is usually done using the estimated axle-count and vehicle length, which is computed using the measured speed and the detection length [71]. Some methods also exist which use only a single loop for classification [72-73]. Their main disadvantages are that the installation requires pavement cut, the mounting and the

maintenance requires lane closure, and deformations of the road surface affect their performance.

Magnetic sensor-based detection systems offer an alternative technology to inductive loops. They measure the changes in the Earth's magnetic field, and are immune to environmental factors, such as rain, snow, or fog. These systems require less pavement cut, and they are less susceptible to stresses of traffic than loops. Some models use batteries and form WSNs [67, 74], which can further decrease the installation cost and pavement damage, since there is no need to install cables to the detector. Magnetometer-based technologies can also effectively work when they are mounted on the side of the roads.

The direction of motion in one direction can be used for detecting vehicles driving in the wrong lane, for example, while overtaking. Another advantage of the magnetometer-based technology is, that even a single sensor unit can effectively estimate the heading direction. This can be done by using only the axis pointing in the direction of movement [75], or by utilizing a two-axis sensor, where the axes are parallel with the ground, for more precise results [76-78].

In recent years, the use of accelerometers for traffic measurement was also reported in the literature. These sensors measure vibrations caused by vehicles passing near the sensors. They can be used alone [79] or fused with magnetometers [80-82]. In these systems, magnetometers are utilized to determine vehicle arrival, departure, and speed, while the data of the accelerometers are used to identify axle locations based on peak detection algorithms.

Vehicle detection can be used for vehicle counting, but it is a very important part of classification algorithms, and it can be also used as inputs for other technologies, like cameras [83].

The magnetic sensor measurements are affected by temperature drifts, which can cause problems in traffic surveillance systems, due to the many changes in the temperature of the pavement [67]. Beside this complicating factor, magnetometer-based systems are also applicable for parking occupancy detection [84-86].

4.2 Related work and motivation

4.2.1 Vehicle detection using magnetic sensors

Various algorithms exist to detect vehicle presence both for in-pavement and roadside technologies. The reported results show, that almost 100% detection efficiency can be reached, while incorrect detections mainly occur by vehicles passing in the neighboring lanes. Some algorithms use fixed threshold values, but the problem is that they cannot deal with the variations of the environment. This can be solved by using adaptive thresholds, which change with the drifts during time. Mostly, a sequence of measurements is considered to be above/under thresholds for declaring both vehicle arrival and departure, since this can filter some false detections. Utilizing only one sensor axis is sufficient to detect vehicle

presence [87-89], but some authors also consider the other axes in their algorithms [67, 90]. The computed magnetic field magnitude [74, 91-93] or the magnetic field angle between sensor axes [94] can be also effectively used for detection purposes. In [74, 91, 94], threshold-based algorithms were applied, while authors of [92] and [93] utilized signal variances.

4.2.2 Vehicle classification using magnetic sensors

Classifying vehicles using magnetic signatures is a very challenging task, since waveforms exhibit large variability even among vehicles of the same class, and the signature also depends from the vehicle's position from the sensor [95].

Many works utilize two or more measurement units for vehicle classification [87, 90, 96-97]. Since the application of multiple sensors enables the computation of vehicle speed and length, the reported methods mostly rely on the measured vehicle length during classification.

Other approaches apply only a single unit, which can decrease the cost of the system. Table 4.1 summarizes the placement of the sensors, the used data types and feature extraction modes during feature extraction, the number of applied vehicle classes and samples, the used classification methods, and the achieved efficiencies in related researches dealing with vehicle classification utilizing only a single unit. The used feature types are shown in Table 4.2. Both roadside [76, 92, 98] and in-pavement [67, 91, 93, 99-101] installation was considered in the literature, and even the usage of a unit fixed atop of the roadway was proposed in [95]. Authors of [100] and [101] applied single units with two sensors, which enables the estimation of vehicle speed and length, but in the rest of the works, the algorithms rely only on different features extracted from the time-series to determine the class of the passing vehicle. Features were extracted using the waveforms measured on one or more sensor axes (the orientation of the X , Y , and Z axes is the same as described in Subchapter 4.3), but the magnetic field magnitude (F) was also widely used. Various time- and frequency-domain features were applied, while in [95], FFT-based components with PCA-based dimension reduction were used to construct the feature set. In related researches, vehicles were classified into two to five classes, but most of the works applied a very small number of samples for classification. The most widely used classification methods were the CT and the SVM, but the k-NN algorithm and the Direct Hill-Pattern Matching classifier were also utilized.

Table 4.1: Summary of relevant works.

Related work	Sensor placement	Data type	Feature extraction	Vehicle classes	Sample number	Classifier	Efficiency
[76]	side of the road	X	TDFs	3 (heavy tracked vehicle; light tracked vehicle; light wheeled vehicle)	93	SVM	86.27%
[92]	side of the road	X, Y	TDFs	5 (motorcycle; hatchback; sedan; SUV; bus)	100	CT	97%
[98]	side of the road	X, Y, Z, F	TDFs, FDFs	4 (hatchback; sedan; bus; multi-purpose vehicle)	300	k-NN	95.46%
[67]	middle of the lane	X, Z	TDFs	5 (passenger vehicle; SUV; van; pickup; bus)	37	Direct Hill-Pattern Matching	63%
[99]	middle of the lane	F	TDFs	2 (car; bus)	542	CT	97.0% on training data, 88.9% on validation data
[91]	middle of the lane	F	TDFs	4 (car; minibus; bus; truck)	100	CT	95%
[93]	middle of the lane	Z	TDFs	4 (sedan and SUV; van and seven-seat; light and medium truck; heavy truck and semi-trailer)	4507	CT	80.55% on validation data
[100]	middle of the lane (1 board with 2 sensors)	Z	TDFs	4 (motorcycle; car; van; pickup)	130	CT	81.69%
[101]	middle of the lane (1 board with 2 sensors)	F	TDFs FDFs	3 (bus; small and medium truck; large truck)	460	SVM	92.8%
[95]	atop of the roadway	F	FDFs	3 (passenger car, 2-axle single-unit vehicles; 2-axle and 3-axle single-unit trucks; 5-axle single-trailer trucks)	1985	SVM	86.85%

Table 4.2: Used feature types in related works.

Related work	Feature types
[76]	concavity area, convexity area, the angle of concave part, the angle of convex part of the waveform
[92]	vehicle signal duration, signal energy, average energy, ratio of positive and negative energy
[98]	position of the maximum, position of the minimum, detection length, peak-to-peak value, mean value, standard deviation, number of extremes, the sign of the first extreme, the number of zero-crossings, energy of the detected signal, average energy, ratio of the energy of the signals on the sensor's axis to the energy of the F signal, first non-zero samples of the frequency spectrum
[67]	Hill-patterns
[99]	number of peaks, maximum peak time ratio, minimum trough time ratio, mean value, the standard deviation, the maximum peak amplitude, the minimum trough amplitude, maximum peak/trough amplitude ratio
[91]	magnetic signature length
[93]	statistical features: magnetic length, mean, variance, maximum and minimum, position of the maximum and minimum, number of local maxima and minima, crossing mean counts, energy, mean energy
[100]	signal length, relative vehicle length, Hill-pattern peaks, three differential energy parameters
[101]	number of local maxima, local minima, extreme points, and negative local minima, relative time of minimum and maximum, penultimate minimum/minimum, maximum value, minimum value, sum of value, average value, max value/min value, max average value/average value, standard deviation, on-time speed, highest spectrum power and the corresponding frequency
[95]	FFT + PCA

4.2.3 Motivation

The goal of this research was to develop an online vehicle classification method utilizing a single, magnetometer-based unit. In-pavement mounting was chosen, since the measurement signals are more detailed and carry more information, than when the sensors are mounted on the side of the roads (described in Subchapter 4.3).

First, a vehicle detection algorithm is proposed, which can deal with the environmental changes and is mainly designed for vehicle classification purposes, but can be also used for other applications.

The developed vehicle classification system classifies vehicles into nine classes, which is the main contribution compared to reported methods, where vehicles were classified into two to five classes. Drawbacks of reported studies are, that most works applied only a small number of collected samples, and many do not consider unknown data to validate their developed algorithms. In this work, a very high number of samples were used to construct

training and validation datasets. Related works mainly do not deal with the implementability of the algorithms. In the proposed system, all parts of the vehicle classification algorithm are realized on the used microcontroller-based hardware. To achieve online and real-time operation, the used inputs were only time-domain features, which require less computation and memory resources than necessary for frequency-domain analysis, which were also widely used in other studies. Other disadvantages of many works include, that they utilize the length of the detection as one of the inputs in classification stage. This feature provides valuable information about the length of the vehicle, but it obviously has a negative effect on recognition efficiency if the unit is placed to a different location where the speed is different than on the location where the training samples were collected. The effect of this feature in the proposed system is also investigated. To minimize the number of used features and the possible cost of the system, different feature extraction modes and combinations of used sensor axes were also examined.

To further increase the efficiency of the system, a rule-based false detection filtering method is also proposed, which can decrease the number of false detections caused by vehicles with high metallic content passing in the neighboring lane.

4.3 Measurement system and data acquisition

Magnetometers are passive sensors that measure the strength of the Earth's magnetic field at a given point. Metallic objects, like vehicles, cause local distortions, which can be measured by these sensors. Vehicle presence can be determined using these distortions. Vehicles have different ferromagnetic composition, and thus, they have different magnetic fingerprints.

Magnetic sensor-based systems can be installed both in the pavement surface and on the roadside, but the magnetic field readings are much stronger when the sensors are mounted below the passing vehicles. Also, the signals are more uniform when roadside detectors are used, since many different ferromagnetic parts pass near the sensors when they are mounted into the pavement [87]. Another disadvantage of roadside installation is that these sensors can only detect vehicles in the adjacent lane.

4.3.1 Measurement unit

The used hardware is an HMC5843-based unit, which was developed by Selma Ltd., Subotica, Serbia. The Honeywell HMC5843 is a small (4mm × 4mm × 1.3mm), surface mount, multi-chip module designed for low field magnetic sensing in three axes. The sensor utilizes Anisotropic MagnetoResistive (AMR) technology, and features precision in-axis sensitivity and linearity, and very low cross-axis sensitivity. The ASIC also contains an I2C serial bus interface. The highest sampling frequency of the magnetometer is 50Hz, and it can measure up to ±650μT in 12-bit resolution.

The hardware is also equipped with an 8-bit microcontroller (Silicon Labs C8051F930), which can communicate with the sensor via I2C interface. The hardware is installed into the pavement in a plastic box (8cm × 8cm × 6cm). The unit also contains an RS-485 interface which enables communication over long distances. The supplying and the communication are realized using a cable.

4.3.2 Data acquisition

A single measurement unit was installed for data acquisition. The location is on the main road of Subotica, Serbia, and it is between two intersections (150-200m from both), where the vehicles are moving with almost constant speed when they are passing above the sensor, since they have enough time from the intersection to accelerate and reach their desired speed. The speed limit is 50km/h, but the speed of the passing vehicles varies greatly, and many drivers even exceed the limitation. The location is very advantageous, since it is a frequented road with various vehicle classes passing.

In-pavement mounting was chosen, since the measurement signals are more detailed and carry more information, than when the sensors are mounted on the side of the roads. The detector was mounted into the middle of the outer lane, as shown in Figure 4.1. The sensor's X axis points to the movement direction, the Y axis points to the neighboring lane, and Z is orthogonal with the pavement surface. The unit was installed into the pavement, 5cm beneath the surface.

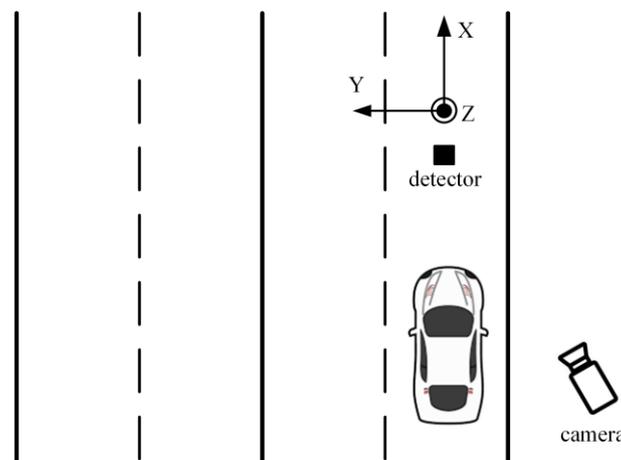


Figure 4.1: Sensor placement for data acquisition.

The developed vehicle detection algorithm (described in Subchapter 4.4) was implemented on the installed measurement unit. The applied sampling frequency was 50Hz. The software of the device sent the measurement data and the value of the detection flag to a server via the RS-485 interface with a baud rate of 115200bps. A data acquisition software on the server side received and stored the measurement data to a database. To validate the detections, and to determine the class for each recorded sample, a camera was installed

beside the road. The data acquisition software saved camera images at every falling edge of the detection flag.

Data acquisition was done for multiple months in various weather conditions, and altogether more than 30000 samples were collected. Figure 4.2 shows the measurement signals on the three axes and the corresponding vehicle images when a car and a bus pass above the sensor.

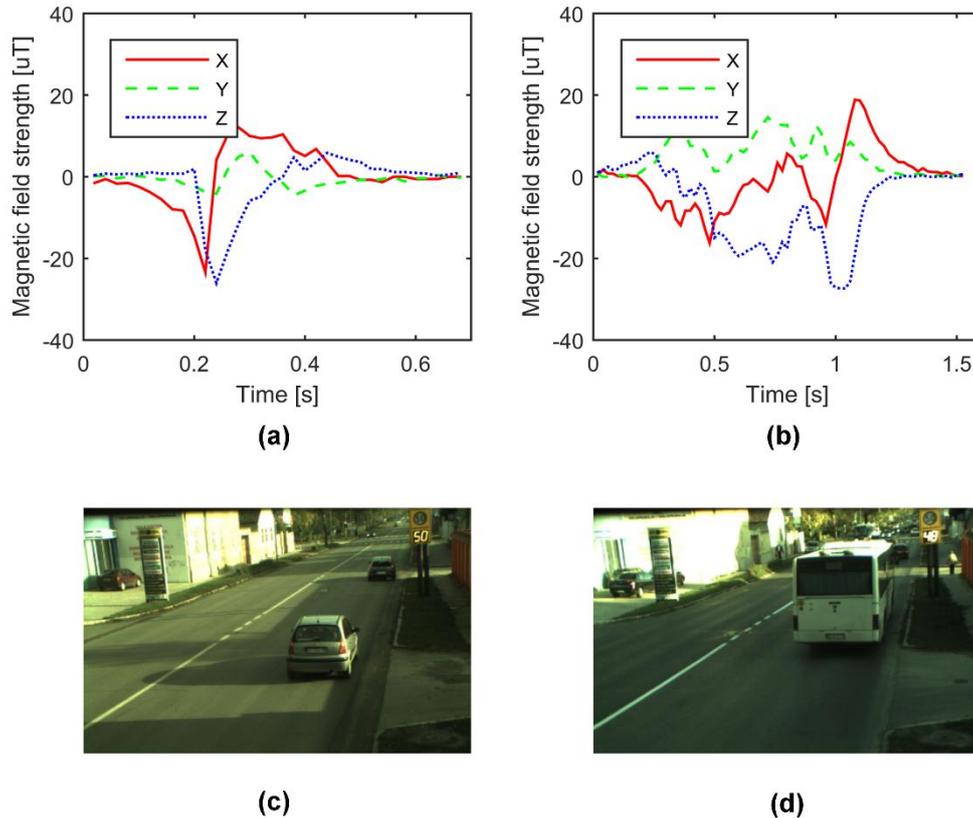


Figure 4.2: Measured signals on the three sensor axes when a car (a) and a bus (b) are passing above the sensor, and the images of the vehicles (c, d).

4.4 Sensor calibration and vehicle detection algorithm

Vehicle detection is a very important part of the classification process, since it should follow the environmental changes, and act the same way for different vehicles. These environmental changes cause slow variations in the offsets on the three sensor axes.

4.4.1 Algorithm description

When the unit is turned on, the calibration process is run. This process uses a calibration range size, which must be slightly larger than the peak-to-peak noise level, because it should follow the slow changes in the environment. This noise level should be estimated after the installation of the unit in the current location, when no vehicles are near to the sensor. During

calibration the highest and lowest measurement values are monitored in a measurement window for all three axes. The difference between the highest and lowest values must be smaller than the calibration range size. The calibration is restarted if the difference exceeds the range size at any time in the window at any of the axes. This filters the factors which affect the magnetic field near the sensor. The size of the measurement window should be at least 1s.

If the calibration is successful, the upper and lower calibration and detection thresholds are calculated for all axes. This is done by equally stretching the range determined by the highest and lowest values to the width defined by the range sizes.

Vehicle presence is declared if the measurement values exceed the detection threshold at both of the X and Z axes. The detection flag is cleared if the measurement values on both X and Z axes are between the calibration thresholds for a previously defined number of measurements. The algorithm should be suitable for detecting vehicles with trailers, so the used length should be calculated using the potential speed on the location and the possible distance between the vehicle and the trailer.

A recalibration is attempted always when vehicle presence is not declared. The process is the same as during calibration, but the measured highest and lowest measurements must fit into the previous calibration range. This enables the following of the environmental changes.

4.4.2 Experimental results

A one-hour test has been done to test the efficiency of the algorithm. The results, divided by basic vehicle classes, are shown in Table 4.3. The overall efficiency of the detection algorithm is 94.15%. The algorithm detects all vehicles passing above the sensor, only motorcycles can cause failures if they are not passing near to the detector. False detections cause the rest of the failures. These are generated by vehicles with high metallic content passing in the neighboring lane. To achieve fast reaction when the vehicles are approaching the sensor, which enables the processing of the entire magnetic signature of the vehicle in the feature extraction stage, the detection range size is set to be as small as possible. Less false detections could be achieved if the thresholds would be raised, but this could also lower the detection rate of motorcycles, and important parts of the magnetic signature could also be lost.

Table 4.3: Efficiency of the detection algorithm during a one-hour test.

Vehicle class	Passed	Detected	Rate
Motorcycle	4	3	75%
Car	168	168	100%
Van	10	10	100%
Truck	15	15	100%
Bus	6	6	100%
Other	2	2	100%
False detections caused by vehicles passing in the neighboring lane		13	
Sum	205	217	94,15%

4.5 Vehicle classification algorithm

The used vehicle classification algorithm consists of three main parts: vehicle detection algorithm, feature extraction, and classification. To perform real-time and online vehicle classification, all parts should be easily implementable on the microcontroller of the used hardware.

The flow chart of the vehicle classification algorithm for one measurement cycle is shown in Figure 4.3.

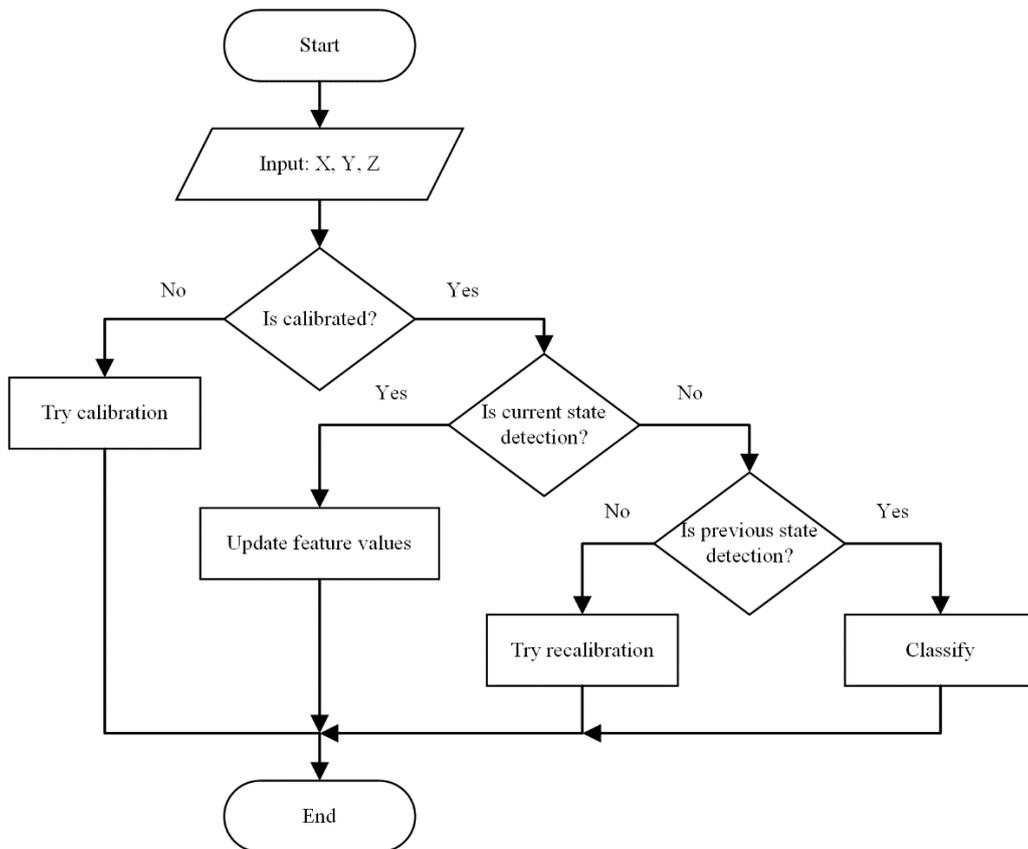


Figure 4.3: The flowchart of the proposed vehicle classification algorithm.

4.5.1 Feature extraction

The extracted features on the time-series data form the inputs in the classification stage. Features are extracted in the detection window, but the last measurements, which are used for clearing the detection flag, are not utilized. The chosen feature types are based on only time domain analysis, because they need little computation, and they also do not require the storage of all measurements in the window. This speeds up the processing time and decreases the necessary memory. The values are updated after every new measurement, and they can be immediately passed to the classifier when the detection is cleared.

Feature types

Feature types were chosen by their potential ability to detect vehicle axles and overall metallic content. It is also important that the features should be immune to speed variations. Some features were previously not applied in related works for vehicle classification purposes, but are very popular in other pattern recognition applications. The used feature types were the following:

- *Detection length (DL)*: The number of measurements in the detection window.
- *Highest value (MAX) and lowest value (MIN)*: The highest and lowest measurement values in the detection window.
- *Place of the highest and lowest values (PlaceMax, PlaceMin)*: The indexes of the measurements where the MAX and MIN points were found, both divided with the detection length.
- *Range changes (RCH)*: The calibration thresholds define three ranges in the signal values, one above the upper threshold, one under the lower, and one between them. This feature measures how many times has the signal switched ranges.
- *Number of local maxima and minima (NumLocMax, NumLocMin)*: The number of local maxima in the range above the upper threshold, and the number of local minima in the range under the lower threshold. A point is considered as a local maximum (minimum) if it has local minima (maxima) before and after it, and the differences in the amplitude are higher than the peak-to-peak noise value.
- *MAV*
- *Mean value (MV)*: The mean amplitude value.
- *NSSC*
- *NZC*
- *Average waveform length (AWL)*: The length of the waveform over the detection window divided by the number of samples in the window.

$$AWL = \frac{1}{DL} \sum_{i=1}^{DL-1} |x_{i+1} - x_i| \quad (4.1)$$

- RMS
- WAMP

Extraction modes

In related works, the raw signal values on the X, Y, and Z axes were utilized for feature computation. In this work, various extraction modes were tested, which can possibly increase efficiency and/or decrease the required feature number. Feature extraction is done using raw sensor measurements and aggregated data, which are computed using the raw measurement data. The tested aggregation modes were motivated by the fact that both the strength and the direction of the magnetic field at the sensor will change when a vehicle passes by. Lower input number in the classification process can decrease the necessary computation time and the required memory space for the implementation of the classifier. The following extraction modes were applied:

- *Measurement axes (X, Y, Z)*: The features were computed using the raw measurement values on each axis. The calibrated offsets were subtracted from the measured values.
- *Absolute values (X_{abs}, Y_{abs}, Z_{abs})*: The computed absolute values were applied on each measurement axis, which were calculated after the offsets were subtracted from the measurement values.
- *Magnitude from the origin (XY_O, XZ_O, YZ_O, XYZ_O)*: Magnitude values were computed in three dimensions and in two dimensions using different combination of the axes. The magnitude of the calibration point was subtracted from computed magnitudes.

$$XY_O = \sqrt{X^2 + Y^2} - \sqrt{X_{cal}^2 + Y_{cal}^2} \quad (4.2)$$

$$XZ_O = \sqrt{X^2 + Z^2} - \sqrt{X_{cal}^2 + Z_{cal}^2} \quad (4.3)$$

$$YZ_O = \sqrt{Y^2 + Z^2} - \sqrt{Y_{cal}^2 + Z_{cal}^2} \quad (4.4)$$

$$XYZ_O = \sqrt{X^2 + Y^2 + Z^2} - \sqrt{X_{cal}^2 + Y_{cal}^2 + Z_{cal}^2} \quad (4.5)$$

where X_{cal} , Y_{cal} , and Z_{cal} define the calibration point, which are the middle points between the upper and lower calibration thresholds on each sensor axis.

- *Angles (XY_A, XZ_A, YZ_A)*: The difference between the angle computed for a measurement point and the angle of the calibration point. The angles were determined for all three planes defined by the sensor axes using (4.6-4.8).

$$XY_A = \tan^{-1}(Y/X) - \tan^{-1}(Y_{cal}/X_{cal}) \quad (4.6)$$

$$XZ_A = \tan^{-1}(Z/X) - \tan^{-1}(Z_{cal}/X_{cal}) \quad (4.7)$$

$$YZ_A = \tan^{-1}(Z/Y) - \tan^{-1}(Z_{cal}/Y_{cal}) \quad (4.8)$$

- *Magnitude from the calibration point* (XY_C , XZ_C , YZ_C , XYZ_C): Magnitudes were computed using the differences between the measurement points and the calibration point. The computation was done also for two and three dimensions as given in (4.9-4.12).

$$XY_C = \sqrt{(X - X_{cal})^2 + (Y - Y_{cal})^2} \quad (4.9)$$

$$XZ_C = \sqrt{(X - X_{cal})^2 + (Z - Z_{cal})^2} \quad (4.10)$$

$$YZ_C = \sqrt{(Y - Y_{cal})^2 + (Z - Z_{cal})^2} \quad (4.11)$$

$$XYZ_C = \sqrt{(X - X_{cal})^2 + (Y - Y_{cal})^2 + (Z - Z_{cal})^2} \quad (4.12)$$

Since the extraction modes based on the absolute values and the magnitudes calculated from the calibration point provide always positive values, the next features types were not computed in these cases: MIN, PlaceMin, NumLocMin, NZC, and MAV.

4.5.2 Classification

Three-layer MLP neural networks were applied for classification in the proposed algorithm. The number of input neurons was equal with the size of the feature vector, while in the output layer a neuron was assigned to each class. The hyperbolic tangent sigmoid transfer function was used in the hidden layer, while the neurons in the output layer were created using the linear transfer function. The training of the MLPs was done offline using the LM algorithm.

4.5.3 Experimental results

Vehicle classes

Taking into account the capabilities of magnetic sensors, nine possibly separable vehicle classes were defined. The vehicle types and the number of axles for each class are shown in Table 4.4.

Table 4.4: Used vehicle classes.

Class number	Vehicle types	Number of axles
1	motorcycle	2
2	car	2
3	car with trailer	2 + 1
4	van, mini bus	2
5	truck	2-3
6	truck with trailer	2-3 + 2-3
7	tractor trailer	2 + 3
8	bus	2
9	articulated bus	3

It is very important to use nearly equal number of samples for all classes during the training of the MLPs, so 130 samples were utilized for each class. The training of the MLP networks was done using 80 (61.54%) samples, while the remaining 50 (38.46%) were used as unknown inputs for the validation of the trained classifiers. In the training process, the 70% of the training data were used as training inputs, and 30% as validation inputs.

Datasets

Since the classification algorithm should be implemented on the microcontroller-based system, it is important to minimize the number of inputs. The waveforms on the X and Z axes should carry most of the useful information. The Y axis can give valuable information about the position of passing vehicle, but it can be also largely affected by vehicles passing in the neighboring lane. The aggregation of the measurement data can further decrease the number of features, and possibly increase the classification efficiency. To find the optimal configuration, it was necessary to test different combinations of used axes. Using the described five feature extraction modes, altogether 18 different combinations were defined:

- *Measurement axes:*
 1. X, Y, and Z
 2. X
 3. Z
 4. X and Z
- *Absolute values:*
 1. X_{abs} , Y_{abs} , and Z_{abs}
 2. X_{abs}
 3. Z_{abs}
 4. X_{abs} and Z_{abs}
- *Magnitudes from the origin:*
 1. XY_O , XZ_O , YZ_O , and XYZ_O
 2. XZ_O
 3. XY_O , XZ_O , YZ_O
 4. XYZ_O
- *Angles:*
 1. XY_A , XZ_A , YZ_A
 2. XZ_A
- *Magnitude from the calibration point:*
 1. XY_C , XZ_C , YZ_C , and XYZ_C
 2. XZ_C
 3. XY_C , XZ_C , YZ_C
 4. XYZ_C

The detection length feature provides vehicle length information when the speed of the vehicles on the observed location is similar. Since other chosen features should be potentially immune to speed differences, it was reasonable to explore the effect of this feature on classification efficiency. This is important, because not using the detection length in the feature set could enable the system to work efficiently on other locations where the potential speed of the vehicles is different. By adding this feature to all defined combinations, altogether 36 different datasets were tested.

Performance evaluation

To provide more valid results, five different training and validation sets were generated for all dataset types by randomly distributing the 130 samples. All 5 datasets for the 36 input combinations were tested with 5, 10, 15, 20, 25, and 30 neurons in the hidden layer. Since the training of the neural networks largely depends from the initial weights, for all setups 5 trainings were done. In the further comparison, the results from the five trainings where the highest efficiency on validation data were achieved were used.

Table 4.5 summarizes the required feature numbers and average recognition rates on training and validation samples for each of the 36 datasets. The computation of the average efficiencies was done using the efficiencies when the best results on unknown samples were obtained for each of the 5 random sets. It can be observed, that the highest efficiencies can be achieved using the feature extraction mode where the magnitudes from the origin were utilized. The highest recognition rate on validation data, 74.67%, was obtained using the magnitude changes in three dimensions (XYZ_o) together with the DL feature. The number of features in this dataset is 15, which is also acceptable. Without applying the information about the length of the detection, the highest result was 73.73%, achieved using features extracted in the XZ -plane (XZ_o), which is a significant result taking into account the number of classes and achieved classification efficiencies in related works. The number of features in this dataset is 14. This feature set with the DL provided 74.44%, which is only 0.23% lower than with the XYZ_o -based extraction. This shows that the Y axis does not carry any additional information if the X and Z axes are used. The extraction mode applying the magnitudes from the calibration point requires smaller number of features compared to the other magnitude-based mode, but the classification efficiencies are also lower. The differences are mostly higher than 3%. Between 70-72% recognition rate can be achieved using the features extracted from the waveforms of the measurement axes. Applying only the X or the Z axis provides very similar results, 71.33% and 71.24% without the detection length, and 72.22% and 72.36% with the DL, respectively. This shows that notable results can be achieved using even a single axis sensor, which can largely decrease the cost of the system. Utilizing the features from the two axes together even decreases the average recognition rate, besides that it requires double number of features. The average efficiencies are 70.13% without and 71.64% with the DL. The extraction mode utilizing the absolute

values requires less features, but the recognition rates are smaller for 1-3% for the same axis combinations. Applying the changes in angles for feature computation can also provide high classification rates. Using only the signals in the XZ-plane can result in 71-72% efficiency on validation samples. The highest classification accuracies on training data can be achieved using the datasets with the highest feature numbers. In more datasets around 82-84% can be reached, but the efficiencies on unknown data are smaller in these cases than when the setups with lower feature numbers are utilized. In the case of the datasets which provide the best results on unknown data (above 72%), around 80-81% efficiency was obtained on training samples. Analyzing the effect of adding the DL feature to the feature vector, it can be stated, that it does not increase the classification efficiencies significantly. The recognition rates are increased in average by 1.28% \pm 1.63% for training data, and 0.80% \pm 1.37% for unknown samples. This shows that the used speed independent feature types are sufficient to extract the necessary information from the waveforms.

Table 4.5: Feature numbers and average recognition efficiencies on training and validation data for different setups.

Used axes					
Feature number					
Efficiency on training data (%)			Efficiency on validation data (%)		
X, Y, Z 42	X, Y, Z, DL 43	X 14	X, DL 15	Z 14	Z, DL 15
82.11 67.69	81.50 66.58	78.83 71.33	80.11 72.22	77.92 71.24	81.36 72.36
X, Z 28	X, Z, DL 29	$X_{abs}, Y_{abs}, Z_{abs}$ 27	$X_{abs}, Y_{abs}, Z_{abs}, DL$ 28	X_{abs} 9	X_{abs}, DL 10
81.61 70.13	81.97 71.64	81.00 71.20	81.81 69.73	73.86 68.18	76.14 70.04
Z_{abs} 9	Z_{abs}, DL 10	X_{abs}, Z_{abs} 18	X_{abs}, Z_{abs}, DL 19	$XY_O, XZ_O, YZ_O,$ XYZ_O 56	$XY_O, XZ_O, YZ_O,$ XYZ_O, DL 57
74.31 69.07	75.56 69.82	78.97 69.96	80.08 71.38	82.20 70.18	82.81 68.18
XZ_O 14	XZ_O, DL 15	XY_O, XZ_O, YZ_O 42	$XY_O, XZ_O, YZ_O,$ DL 43	XYZ_O 14	XYZ_O, DL 15
80.75 73.73	80.14 74.44	81.03 69.82	84.06 71.51	77.97 72.76	81.39 74.67
XY_A, XZ_A, YZ_A 42	$XY_A, XZ_A, YZ_A,$ DL 43	XZ_A 14	XZ_A, DL 15	$XY_C, XZ_C, YZ_C,$ XYZ_C 36	$XY_C, XZ_C, YZ_C,$ XYZ_C, DL 37
79.00 66.53	79.97 65.78	80.92 71.20	79.69 71.87	79.08 66.36	81.75 69.11
XZ_C 9	XZ_C, DL 10	XY_C, XZ_C, YZ_C 27	$XY_C, XZ_C, YZ_C,$ DL 28	XYZ_C 9	XYZ_C, DL 10
72.78 68.49	76.53 70.53	80.83 68.36	79.28 68.27	73.61 68.93	75.69 71.38

The average misclassification rates on the 5 random validation sets utilizing the XZ_0 -based features with and without the DL feature are presented in Table 4.6. This extraction mode provided the highest results on unknown data. It can be observed, that the motorcycles can be recognized with more than 96% in both cases. Cars are correctly classified for 90.0% of the samples with, and 87.6% without the DL feature. Most of the improperly classified samples are assigned as vans (4.8% and 6.8%). Buses can be recognized with misclassification rates of 11.2% and 13.6%. Articulated buses have lower classification rates, and the DL feature even lowers the efficiency. The recognition rates are 76.4% with and 83.2% without the DL. Most of the misclassifications occur as trucks with trailer (8.8% and 6.8%) and buses (11.6% and 5.6%). The most positive effect of the added DL feature can be noticed at vans, which are classified with the highest misclassification rates (44.8% and 52%). 24.0% of the samples are classified as trucks for both datasets, and 7.6% and 8.4% as cars with trailer. The improvements between the two sets can be noticed at cars, since using the DL the misclassification rates decrease from 17.6% to 10.4%. This result is reasonable, since both classes have 2 axles and do not differ significantly in metallic content. The rest of the vehicle classes, trucks, trucks with trailer, and tractor trailers can be recognized with similar rates, around 55-60%. Trucks are mostly misclassified as vans (10.8% and 8.4%) and buses (12.4% and 18.0%), trucks with trailer as tractor trailers (16.8% and 21.6%) and articulated buses (13.2% and 11.6%), while tractor trailers as trucks with trailer (13.6% and 10.8%), buses (12.8% and 10.8%), and articulated buses (8.4% and 9.6%).

The required number of hidden layer neurons for achieving 97% convergence in efficiency using the mean values from the 5 random sets can be seen in Table 4.7. Analyzing the results, it can be observed, that mostly 10 (58.33%) and 15 (30.56%) neurons are required for validation samples. To reach convergence on training data, more, between 15 and 25 neurons are necessary.

Table 4.6: Average misclassification rates on validation data using the XZ_0 -based features with (upper row) and without (lower row) using the detection length feature.

		Output class									Sum
		1	2	3	4	5	6	7	8	9	
Target class	1		1.6%		0.4%	1.2%					3.2%
			2.8%		0.4%						3.2%
	2	2.0%		2.8%	4.8%	0.4%					10.0%
		2.0%		2.4%	6.8%	1.2%					12.4%
	3		4.8%		7.2%	10.8%		1.2%	1.2%		25.2%
			7.2%		10.0%	10.8%	0.4%	0.4%	1.2%		30.0%
	4	0.8%	10.4%	7.6%		24.0%		0.4%	1.6%		44.8%
		0.8%	17.6%	8.0%		24.0%			1.6%		52.0%
	5		0.8%	6.0%	10.8%		0.8%	2.8%	12.4%	1.2%	34.8%
			1.2%	4.8%	8.4%		0.8%	2.8%	18.0%	0.4%	36.4%

	6			0.8%	0.4%	2.4%		16.8%	4.4%	13.2%	38.0%
				2.0%		1.6%		21.6%	4.0%	11.6%	40.8%
	7			1.6%	0.4%	2.4%	13.6%		12.8%	8.4%	39.2%
				0.4%		2.0%	16.4%		10.8%	9.6%	39.2%
	8			0.8%	3.2%	1.2%	3.6%		2.4%	11.2%	
				0.4%	4.0%	0.8%	4.0%		4.4%	13.6%	
	9					0.4%	8.8%	2.8%	11.6%		23.6%
				0.4%		6.8%	4.0%	5.6%			16.8%

Table 4.7: Required number of hidden layer neurons for reaching 97% convergence on the 36 datasets.

Hidden layer neuron number	5	10	15	20	25	30
Percentage of convergence for training data	0.00%	2.78%	19.44%	30.56%	38.89%	8.33%
Percentage of convergence for validation data	0.00%	58.33%	30.56%	8.33%	2.78%	0.00%

Implementation

The implementation of MLPs requires the storage of three parameter types: input ranges, weights, and biases. Input ranges are used for the normalization of input values, and the matrices consist the highest and lowest values for all inputs. Two weight matrices are required, one connecting the input layer with the hidden layer, and the second connecting the hidden layer and the output layer. The dimensions of the matrices can be given using the sizes of the layers. Bias values are used in all neurons of the hidden and the output layer.

The necessary memory for different feature and hidden layer neuron numbers can be seen in Figure 4.4. The required memory spaces are given in bytes. Since all parameter types are floating-point numbers, during computation 4 bytes were counted for each parameter.

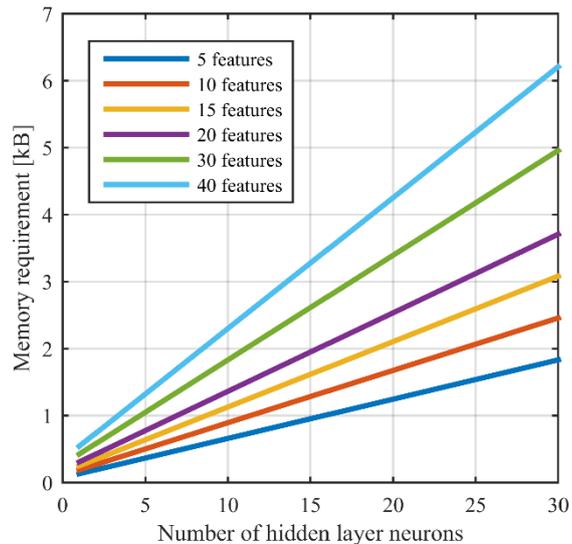


Figure 4.4: Memory requirements for the implementation of MLP networks.

The testing shows that after the detection is cleared, the vehicle class can be determined in 1-2 measurement cycles (20-40ms), without affecting the other parts of the algorithm. The required time depends of the size of the network.

4.6 Rule-based false detection filtering algorithm

The overall efficiency of the classification algorithm can be largely affected by the false detections, thus, it is important to minimize their number. The proposed false detection filtering method is based on different rules, which are constructed using different data types and relations.

4.6.1 Used data in rule construction

Finding optimal parameters and relations was tested with different data types. Beside the measurement values on the three sensor axes, aggregated data computed using the measurement values on the three axes were also tested. The axis pointing to the neighboring lane should carry the most information for false detection filtering, but other sensor axes were also considered. The following data types were utilized:

- *Measurement axes* (X, Y, Z),
- *Magnitudes from the origin* (XY_O, XZ_O, YZ_O),
- *Angles* (XY_A, XZ_A, YZ_A).

For all samples the following value types were determined in the detection window for all data types:

- Highest value,
- Lowest value,
- Highest value in the case of absolute values of the measurements.

Since a vehicle with high metallic content passing in the neighboring lane can generate larger differences in the signals than a vehicle with lower metallic content passing above the sensor, it is reasonable to test ratios computed using different data, e.g. the ratio of the highest values on the X and Z axes. Ratios were calculated for all combinations in different data types in the case of all value types. Reciprocal combinations were not computed, e.g. if X/Z is calculated then there is no need to compute Z/X .

4.6.2 Optimization

Genetic algorithms were used for parameter optimization. Evolutionary algorithm-based techniques are effective and widely used methods in optimization tasks. In the constructed fitness functions, the goal was to minimize the fitness value.

To find the parameter types which can be effectively used for the filtering of false detections, first all combinations of data types and value types were tested separately. The mistakes in the case of correct detections is allowed, but the rate of both correctly determined good and false detections are used in the fitness function. The rates are weighted, since it is less important to detect false detections than to lose correct ones. The used fitness function is the following:

$$fitval = w_c - w_c \cdot n_c/N_c + w_f - w_f \cdot n_f/N_f \quad (4.13)$$

where $fitval$ is the fitness value, n_c is the number of found correct detections, N_c is the number of all correct detections, n_f is the number of found false detections, N_f is the number of all false detections, w_c is the weight used at correct detections, and w_f is the weight used at false detections.

After finding the most useful parameters, it is important to combine these parameters in more complex rules for more accurate results. The rules were combined in two different modes, since both the “AND” and “OR” logical operators between the separate rules were tested during the examination of the samples.

4.6.3 Performance evaluation

Both “<” and “>” relations were tested during the optimization of single parameters types, and it shows when is a detection declared as false. Using all data, value and relation types, altogether 108 combinations were tested.

Optimization was done using 600 samples of correct detections and 300 samples of false detections. The detections included measurements of various types of vehicle classes, including cars, vans, trucks, and buses. Every optimization was done in 5000 generations with a population size of 50.

The testing of single parameter types was done with 0.75 weight for correct detections and 0.25 for false detections. The results in the case of single parameters can be seen in Table 4.8 for measurement parameter types and in Table 4.9 for rate parameter types where the

achieved fitness value was lower than 0.1 or nearly this value. The indices at data types represent the applied value type, and the used abbreviations are the following: h – highest value, l – lowest value, a – highest value in the case of absolute values of measurements.

Table 4.8: Achieved results in the case of single measurement parameter types.

Data type	Relation	Fitness value	Correct good detection rate (%)	Correct false detection rate (%)
X_h	<	0.043	99.00	85.67
X_a	<	0.047	98.50	85.67
Z_a	<	0.013	99.50	96.33
XY_{Oh}	<	0.076	96.50	80.00
XZ_{Oh}	<	0.011	99.67	96.67
YZ_{Oh}	<	0.011	99.67	96.67
XY_{Al}	<	0.095	95.33	76.00
XZ_{Al}	<	0.038	99.00	87.67
XY_{Aa}	<	0.079	93.83	87.00
XZ_{Aa}	<	0.039	99.00	87.33

Table 4.9: Achieved results in the case of single ratio parameter types.

Data type	Relation	Fitness value	Correct good detection rate (%)	Correct false detection rate (%)
X_l/Z_L	>	0.031	98.83	91.00
Y_l/Z_L	>	0.041	98.17	89.00
Y_A/Z_A	>	0.065	97.83	80.33
XY_{Oh}/XZ_{Oh}	<	0.104	98.33	63.33
XY_{Oh}/YZ_{Oh}	<	0.102	98.83	62.67
XZ_{Oh}/YZ_{Oh}	<	0.062	97.50	82.67

The obtained results using measurement parameter types show that with even a single parameter almost 97% of the false detections can be filtered with losing around 0.3% of good detections. It can be also observed, that despite that the axis pointing to neighboring lane (Y) should carry most information in the case of false detections, the other two axes provide more reliable parameters. The lowest fitness value using measurements on the sensor axes, 0.013, was achieved using the highest absolute measurement values on the Z axis. Utilizing the highest values or the highest values on absolute measurement values in the case of the X axis also provides fitness values under 0.05. The best results were obtained using aggregated data. A fitness value of 0.011 was achieved using highest values on XZ_O and YZ_O magnitudes.

Highest absolute values in the case of these magnitudes provided identical results, so they were not included in Table 4.8. Computed angles can also give useful results.

The results achieved using computed ratios show that these parameters have lower false detection filtering capability. The lowest fitness value, 0.031, was obtained using X_l/Z_l , which means 91% filtering of false detections with losing a little more than 1% of correct detections. Slightly higher fitness value can be achieved using the rate Y_l/Z_l . The best result using rates of computed magnitudes was obtained utilizing XZ_{Oh}/YZ_{Oh} , which gave a fitness value of 0.062.

Parameter types with adequate relations which showed that are capable of filtering false detections were combined in more complex rules. Altogether six combinations were defined using groups of similar parameter types: measurement values on the sensor axes, magnitudes, angles, ratios computed using measurement data of the sensor axes, ratios using magnitudes, ratios using angles. Both the “AND” and “OR” logical operators were tested. The applied weights were 0.9 for correct good detections and 0.1 for correct false detections. Table 4.10 summarizes the obtained results.

Table 4.10: Achieved results in the case of different parameter combinations.

Data types and relations	Logical operator	Fitness value	Correct good detection rate [%]	Correct false detection rate [%]
$X_a (<), Y_a (<), Z_a (<)$	AND	0.008	99.67	95.33
$X_a (<), Y_a (<), Z_a (<)$	OR	0.008	99.67	95.33
$XY_{Oh} (<), XZ_{Oh} (<), YZ_{Oh} (<), XY_{Ol} (<)$	AND	0.006	99.67	96.67
$XY_{Oh} (<), XZ_{Oh} (<), YZ_{Oh} (<), XY_{Ol} (<)$	OR	0.006	99.67	97.00
$XY_{Al} (<), XZ_{Al} (<), XY_{Aa} (<), XZ_{Aa} (<)$	AND	0.018	99.83	83.33
$XY_{Al} (<), XZ_{Al} (<), XY_{Aa} (<), XZ_{Aa} (<)$	OR	0.021	99.50	83.67
$X_h/Y_h (<), X_h/Z_h (<), X_l/Z_l (>), Y_l/Z_l (>), X_a/Z_a (>),$ $Y_a/Z_a (>)$	AND	0.018	99.50	86.33
$X_h/Y_h (<), X_h/Z_h (<), X_l/Z_l (>), Y_l/Z_l (>), X_a/Z_a (>),$ $Y_a/Z_a (>)$	OR	0.008	99.50	96.33
$XY_{Oh}/XZ_{Oh} (<), XY_{Oh}/YZ_{Oh} (<), XZ_{Oh}/YZ_{Oh} (<),$ $XY_{Oa}/XZ_{Oa} (>), XY_{Oa}/YZ_{Oa} (>)$	AND	0.028	99.50	76.33
$XY_{Oh}/XZ_{Oh} (<), XY_{Oh}/YZ_{Oh} (<), XZ_{Oh}/YZ_{Oh} (<),$ $XY_{Oa}/XZ_{Oa} (>), XY_{Oa}/YZ_{Oa} (>)$	OR	0.035	98.50	79.00
$XY_{Al}/XZ_{Al} (<), XY_{Al}/YZ_{Al} (<), XZ_{Al}/YZ_{Al} (<),$ $XY_{Aa}/YZ_{Aa} (<)$	AND	0.057	100.00	43.33
$XY_{Al}/XZ_{Al} (<), XY_{Al}/YZ_{Al} (<), XZ_{Al}/YZ_{Al} (<),$ $XY_{Aa}/YZ_{Aa} (<)$	OR	0.059	99.83	42.67

The results show that using measurement values and magnitudes the results cannot be improved compared to results obtained with single parameters. It can be also seen, that the

“AND” and “OR” logical operators provide almost identical results. The best results were achieved using magnitude values, which can filter 97% of the false detections with 99.67% of good detections remaining correct. Complex rules constructed using measurement values on the sensor axes provided even lower false detection filtering rates (95.33%) compared to results achieved with using only the Z_a parameter (96.33%). In the case of computed ratios of measurements on sensor axes, combining the parameters with the “OR” logical operator can significantly improve the results. The false detection filtering rate is 96.33%, while with single rules only 91% can be achieved. Applying the “AND” operator between single rules in this case can provide only 10% lower filtering rates. Combining ratios computed from magnitudes can lead to a false detection filtering ratio of only 76.33% when the “AND” operator is applied, and 79% in the case of the “OR” operator. Complex rules constructed using ratios of angle parameters provide only around 43% filtering.

4.7 Conclusion

In this chapter, a new, real-time vehicle classification system was presented, which utilizes only a single tri-axial magnetic sensor.

Data acquisition was done for multiple months in real environment using a unit mounted in the pavement. A camera was also installed beside the road, and images were saved for every detection to determine the vehicle class for each sample. Vehicle detection was realized using the proposed detection algorithm, which applies adaptive thresholds to follow the environmental changes. The accuracy of the proposed algorithm is 94.15%, with errors mainly caused vehicles passing in the neighboring lane.

The online classification applies three-layer MLP neural networks. Nine vehicle classes were defined, and 130 samples per class were used for the training and validation of the classifiers. The utilized features for classification were extracted in the detection windows. The chosen feature types are based on only time domain analysis and require low computation and memory. Since the classification algorithm should be implemented on the microcontroller-based system, it is important to minimize the applied feature number. Various feature extraction modes and used sensor axes were tested to find the optimal configuration. The effect of the detection length was also investigated, since it depends from the speed of the vehicle. Altogether 36 different datasets were constructed and tested.

The obtained results show, that using aggregated data in the form of angles or magnitudes, can even increase the classification efficiencies besides decreasing the number of inputs. The highest recognition rates on validation data were achieved using magnitudes computed from the origin in the XZ-plane, 74.67% with, and 73.73% without applying the detection length. Using only the X or the Z axis can provide 71-72%, which shows that even a single axis sensor can effectively classify vehicles into multiple classes. The length of the detection in average increases the classification efficiencies only by 1.28% \pm 1.63% for training data, and 0.80% \pm 1.37% for unknown samples, which shows that the used features

can effectively extract the information from the waveforms. The necessary feature number in the best sets was 14-15, while convergence on unknown samples can be noticed with 10-15 neurons.

The testing shows that the proposed algorithm can be implemented on the microcontroller of the unit, and the vehicle class can be determined in 1-2 measurement cycles (20-40ms) after the detection is cleared. The solution could be also effectively used in WSN applications, since the classification algorithm runs online, and there is no need to transmit measurement data to a server.

A false detection filtering algorithm was also proposed, since decreasing the number of false detections can improve the overall efficiency of the classification system. The filtering is based on different rules constructed using different parameter types extracted from the signals. All 108 combinations of data, value, and relation types were tested separately to find the most useful parameters. These were later combined in more complex rules to achieve higher filtering rates. The optimization of the parameter values was achieved using genetic algorithms. During the computation of the fitness values, correctly declared good detections were used with higher weights than correctly determined false detections.

The achieved results show that 97% of false detections can be filtered with losing only nearly 0.3% of good detections. The highest results can be obtained using computed magnitudes, and slightly lower rates can be achieved using measurements on sensor axes. A single parameter can be also sufficient for the filtering of the false detections. It can be also noticed that despite that the axis pointing to neighboring lane (Y) should carry most information for the filtering of false detections, the other two axes provide more reliable parameters.

The future goals of this research include improving the recognition efficiency with adding more features which could discriminate different trucks and buses, and finding the features with most influence using different feature selection algorithms to reduce the required number of inputs in the classification stage.

Thesis Group 2. I developed a new real-time vehicle classification system based on a single magnetic sensor mounted into the surface of the pavement.

Thesis 2.1. I developed a prototype measurement system with a single magnetometer-based unit installed into the pavement surface. I also used a camera mounted on the side of the road to validate the detections and to determine the class of the samples. I collected a very large database of samples using the measurement system.

Thesis 2.2. I developed a vehicle detection algorithm based on adaptive thresholds. The algorithm can deal with the environmental changes and is mainly designed for vehicle classification purposes. The algorithm estimates the offsets on each sensor axis using a

calibration method after the unit is turned on, and follows the environmental changes by continuous recalibration. The algorithm is also capable of detecting vehicles with trailers.

Thesis 2.3. I developed a new online vehicle classification algorithm, which can classify vehicles into nine vehicle classes using MLP neural networks. The algorithm uses TDFs, which can potentially detect vehicle axles and overall metallic content, are immune to speed variations, and require low memory usage.

I showed that a single magnetometer-based system is sufficient for classifying vehicles into nine vehicle classes.

I showed that using aggregated data in the form of angles or magnitudes, can even increase the classification efficiencies besides decreasing the number of inputs.

I showed that adding the detection length to the feature set, which is a speed-dependent feature, only slightly increases the classification efficiencies. This shows that the used features can effectively extract the information from the waveforms.

Thesis 2.4. I developed a rule-based algorithm for the filtering of false detections caused by vehicles with high metallic content passing in the neighboring lane. This can further increase the overall efficiency of the proposed vehicle classification system.

I showed that using proper rules, a very high percentage of false detections can be filtered with losing only a few good detections.

I showed that despite that the axis pointing to neighboring lane should carry most information for the filtering of false detections, the other two axes provide more reliable parameters.

I showed that aggregated data in the form of magnitudes can provide higher filtering rates than parameters extracted from the signals on the sensor axes.

Publications of the author

Publications related to the thesis

- [S1] P. Sarcevic, Vehicle classification using neural networks with a single magnetic detector. *Issues and Challenges of Intelligent Systems and Computational Intelligence, Studies in Computational Intelligence*, vol. 530, pp. 103-115, 2014.
- [S2] P. Sarcevic, and S. Pletl, Vehicle classification and false detection filtering using a single magnetic detector based intelligent sensor. In: *Proc. of International Conference on Information Society and Technology (ICIST)*, Kopaonik, Serbia, pp. 144-149, 2014.
- [S3] P. Sarcevic, Z. Kincses, and S. Pletl, Wireless sensor network based movement classification using wrist-mounted 9DOF sensor boards. In: *Proc. of IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, Budapest, Hungary, pp. 85-90, 2014.
- [S4] P. Sarcevic, Z. Kincses, and S. Pletl, Comparison of different classifiers in movement recognition using WSN-based wrist-mounted sensors. In: *Proc. of IEEE Sensors Applications Symposium (SAS)*, Zadar, Croatia, pp. 446-451, 2015.
- [S5] P. Sarcevic, Z. Kincses, S. Pletl, and L. Schaffer, Distributed movement recognition algorithm based on wrist-mounted wireless sensor motes. In: *Proc. of European Wireless (EW)*, Budapest, Hungary, pp. 130-135, 2015.
- [S6] P. Sarcevic, L. Schaffer, Z. Kincses, and S. Pletl, Hierarchical-distributed approach to movement classification using wrist-mounted wireless inertial and magnetic sensors. *Infocommunications Journal* 7(3): 33-41, 2015.
- [S7] P. Sarcevic, Z. Kincses, and S. Pletl, Online human movement classification using wrist-worn wireless sensors. *Journal of Ambient Intelligence and Humanized Computing*, 2017. <https://doi.org/10.1007/s12652-017-0606-1>
- [S8] P. Sarcevic, Z. Kincses, and S. Pletl, Comparison of time- and frequency-domain features for movement classification using data from wrist-worn sensors. In: *Proc. of IEEE International Symposium on Intelligent Systems and Informatics (SISY)*, Subotica, Serbia, pp. 261-265, 2017.

- [S9] P. Sarcevic, and S. Pletl, False detection filtering method for magnetic sensor-based vehicle detection systems. In: *Proc. of IEEE International Symposium on Intelligent Systems and Informatics (SISY)*, Subotica, Serbia, pp. 277-281, 2018.
- [S10] P. Sarcevic, and S. Pletl, Real-time vehicle classification system using a single magnetometer. *Acta Polytechnica Hungarica* (submitted).

Publications not related to the thesis

- [S11] P. Sarcevic, S. Pletl, and Z. Kincses, Evolutionary algorithm based 9DOF sensor board calibration. In: *Proc. of IEEE International Symposium on Intelligent Systems and Informatics (SISY)*, Subotica, Serbia, pp. 187-192, 2014.
- [S12] P. Sarcevic, S. Pletl, and Z. Kincses, Examining the number of required stationary orientations for efficient accelerometer calibration. In: *Proc. of International Conference and Workshop Mechatronics in Practice and Education (MECHEDU)*, Subotica, Serbia, pp. 38-41, 2017.

Bibliography

- [1] J. W. Judy, Microelectromechanical systems (MEMS): fabrication, design and applications. *Smart Materials and Structures* 10(6): 1115-1134, 2001.
- [2] G. J. Fouche, and R. Malekian, Drone as an autonomous aerial sensor system for motion planning. *Measurement* 119: 142-155, 2018.
- [3] C. A. Brooks, and K. Iagnemma, Vibration-based terrain classification for planetary exploration rovers. *IEEE Transactions on Robotics* 21(6): 1185-1191, 2005.
- [4] M.S. Safizadeh, and S.K. Latifi, Using multi-sensor data fusion for vibration fault diagnosis of rolling element bearings by accelerometer and load cell. *Information Fusion* 18: 1-8, 2014.
- [5] S. El-Zahab, E. M. Abdelkader, and T. Zayed, An accelerometer-based leak detection system. *Mechanical Systems and Signal Processing* 108: 276-291, 2018.
- [6] A. Silva, A. Zarzo, J. M. Munoz-Guijosa, and F. Miniello, Evaluation of the continuous wavelet transform for detection of single-point rub in aeroderivative gas turbines with accelerometers. *Sensors* 18(6): 1931, 2018.
- [7] J. Lenz, and A. S. Edelstein, Magnetic sensors and their applications. *IEEE Sensors Journal* 6(3): 631-649, 2006.
- [8] W. Li, and J. Wang, Magnetic sensors for navigation applications: an overview. *The Journal of Navigation* 67(2): 263-275, 2014.
- [9] S. Gao, Y. Zhong, X. Zhang, and B. Shirinzadeh, Multi-sensor optimal data fusion for INS/GPS/SAR integrated navigation system. *Aerospace Science and Technology* 13(4-5): 232-237, 2009.
- [10] H. Zhao, and Z. Wang, Motion measurement using inertial sensors, ultrasonic sensors, and magnetometers with extended Kalman filter for data fusion. *IEEE Sensors Journal* 12(5): 943-953, 2012.
- [11] L. Marin, M. Valles, A. Soriano, A. Valera, and P. Albertos, Multi sensor fusion framework for indoor-outdoor localization of limited resource mobile robots. *Sensors* 13(10): 14133-14160, 2013.

- [12] J. Simanek, M. Reinstein, and V. Kubelka, Evaluation of the EKF-based estimation architectures for data fusion in mobile robots. *IEEE/ASME Transactions on Mechatronics* 20(2): 985-990, 2015.
- [13] K. Liu, C. Chen, R. Jafari, and N. Kehtarnavaz, Fusion of inertial and depth sensor data for robust hand gesture recognition. *IEEE Sensors Journal* 14(6): 1898-1903, 2014.
- [14] Y.-L. Hsu, C.-L. Chu, Y.-J. Tsai, and J.-S. Wang, An inertial pen with dynamic time warping recognizer for handwriting and gesture recognition. *IEEE Sensors Journal* 15(1): 154-163, 2015.
- [15] T. de Quadros, A. E. Lazzaretti, and F. K. Schneider, A movement decomposition and machine learning-based fall detection system using wrist wearable device. *IEEE Sensors Journal* 18(12): 5082-5089, 2018.
- [16] P. Tsinganos, and A. Skodras, On the comparison of wearable sensor data fusion to a single sensor machine learning technique in fall detection, *Sensors* 18(2): 592, 2018.
- [17] Q. Gu, Z. Li, and J. Han, Linear discriminant dimensionality reduction. *Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science*, vol. 6911, pp. 549-564, 2011.
- [18] A. Martinez, and A. Kak, PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(2): 228-233, 2001.
- [19] T. Mitchell, *Machine learning*. McGraw-Hill, New York, 1997.
- [20] M. Moller, A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* 6(4): 525-533, 1993.
- [21] M. Hagan, and M. Menhaj, Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks* 5(6): 989-993, 1994.
- [22] J. P. Varkey, D. Pompili, and T. Walls, Human motion recognition using a wireless sensor-based wearable system. *Personal and Ubiquitous Computing* 16(7): 897-910, 2012.
- [23] K. Altun, B. Barshan, and O. Tuncel, Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition* 43(10): 3605-3620, 2010.
- [24] C. Li, M. Lin, L. Yang, and C. Ding, Integrating the enriched feature with machine learning algorithms for human movement and fall detection. *The Journal of Supercomputing* 67(3): 854-865, 2014.

-
- [25] H. Martin, A. Bernardos, J. Iglesias, and J. Casar, Activity logging using lightweight classification techniques in mobile devices. *Personal and Ubiquitous Computing* 17(4): 675-695, 2013.
- [26] J. McCall, Genetic algorithms for modelling and optimisation. *Journal of Computational and Applied Mathematics* 184(1): 205-222, 2005.
- [27] T. Back, D. B. Fogel, and Z. Michalewicz, *Handbook of evolutionary computation*. IOP Publishing, Bristol, 1997.
- [28] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar, A survey on activity detection and classification using wearable sensors. *IEEE Sensors Journal* 17(2): 386-403, 2017.
- [29] H. Alemdar, and C. Ersoy, Wireless sensor networks for healthcare: A survey. *Computer Networks* 54(15): 2688-2710, 2010.
- [30] F. Aiello, F.L. Bellifemine, G. Fortino, S. Galzarano, and R. Gravina, An agent-based signal processing in-node environment for real-time human activity monitoring based on wireless body sensor networks. *Engineering Applications of Artificial Intelligence* 24(7): 1147-1161, 2011.
- [31] R. Gravina, P. Alinia, H. Ghasemzadeh, and G. Fortino, Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges. *Information Fusion* 35: 68-80, 2017.
- [32] H. Ghasemzadeh, S. Ostadabbas, E. Guenterberg, and A. Pantelopoulos, Wireless medical-embedded systems: A review of signal-processing techniques for classification. *IEEE Sensors Journal* 13(2): 423-437, 2013.
- [33] S. Preece, J. Y. Goulermas, L. Kenney, and D. Howard, A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE Transactions on Biomedical Engineering* 56(3): 871-879, 2009.
- [34] M.-W. Lee, A. M. Khan, and T.-S. Kim, A single tri-axial accelerometer-based real-time personal life log system capable of human activity recognition and exercise information generation. *Personal and Ubiquitous Computing* 15(8): 887-898, 2011.
- [35] C. Zhu, and W. Sheng, Motion- and location-based online human daily activity recognition. *Pervasive and Mobile Computing* 7(2): 256-269, 2011.
- [36] D. Fuentes, L. Gonzalez-Abril, C. Angulo, and J.A. Ortega, Online motion recognition using an accelerometer in a mobile device. *Expert Systems with Applications* 39(3): 2461-2465, 2012.
- [37] S. Chernbumroong, S. Cang, and H. Yu, A practical multi-sensor activity

- recognition system for home-based care. *Decision Support Systems* 66: 61–70, 2014.
- [38] F. Attal, M. Dedabrishvili, S. Mohammed, F. Chamroukhi, L. Oukhellou, and Y. Amirat, Physical human activity recognition using wearable sensors. *Sensors* 15(12): 31314-31338, 2015.
- [39] I. Suarez, A. Jahn, C. Anderson, and K. David, Improved activity recognition by using enriched acceleration data. In: *Proc. of ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, Osaka, Japan, pp. 1011-1015, 2015.
- [40] J. Korpela, T. Maekawa, J. Eberle, D. Chakraborty, and K. Aberer, Tree-structured classifier for acceleration-based activity and gesture recognition on smartwatches. In: *Proc. of IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, Sydney, Australia, 2016.
- [41] R. Ugolotti, F. Sassi, M. Mordonini, and S. Cagnoni, Multi-sensor system for detection and classification of human activities. *Journal of Ambient Intelligence and Humanized Computing* 4(1): 27-41, 2013.
- [42] A. Yang, R. Jafari, S. Sastry, and R. Bajcsy, Distributed recognition of human actions using wearable motion sensor networks. *Journal of Ambient Intelligence and Smart Environments* 1(2): 103-115, 2009.
- [43] M. Field, D. Stirling, Z. Pan, M. Ros, and F. Naghdy, Recognizing human motions through mixture modeling of inertial data. *Pattern Recognition* 48(8): 2394-2406, 2015.
- [44] L. Bao, and S. S. Intille, Activity recognition from user-annotated acceleration data. *Pervasive Computing, Lecture Notes in Computer Science*, vol 3001, pp. 1-17, 2004.
- [45] S. Gonzalez, J. Sedano, J. Villar, E. Corchado, A. Herrero, and B. Baruaque, Features and models for human activity recognition. *Neurocomputing* 167: 52-60, 2015.
- [46] T. Maekawa, Y. Kishino, Y. Sakurai, and T. Suyama, Activity recognition with hand-worn magnetic sensors. *Personal and Ubiquitous Computing* 17(6): 1085-1094, 2013.
- [47] Y.-S. Lee, and S.-B. Cho, Layered hidden Markov models to recognize activity with built-in sensors on Android smartphone. *Pattern Analysis and Applications* 19(4): 1181–1193, 2016.
- [48] G. Cohn, S. Gupta, T.-J. Lee, D. Morris, J. Smith, M. Reynolds, D. Tan, and S.

- Patel, An ultra-low-power human body motion sensor using static electric field sensing. In: *Proc. of ACM Conference on Ubiquitous Computing (UbiComp)*, Pittsburgh, USA, pp. 99-102, 2012.
- [49] Y. Guo, W. He, and C. Gao, Human activity recognition by fusing multiple sensor nodes in the wearable sensor systems. *Journal of Mechanics in Medicine and Biology* 12(5): 1250084, 2012.
- [50] D. Nemeč, A. Janota, M. Hrubos, and V. Simak, Intelligent real-time MEMS sensor fusion and calibration. *IEEE Sensors Journal* 16(19): 7150-7160, 2016.
- [51] S. Poddar, V. Kumar, and A. Kumar, A comprehensive overview of inertial sensor calibration techniques. *Journal of Dynamic Systems, Measurement, and Control* 139(1): 011006, 2016.
- [52] Q. Cai, N. Song, G. Yang, and Y. Liu, Accelerometer calibration with nonlinear scale factor based on multi-position observation. *Measurement Science and Technology* 24(10): 105002, 2013.
- [53] G. Secer, and B. Barshan, Improvements in deterministic error modeling and calibration of inertial sensors and magnetometers. *Sensors and Actuators A: Physical* 247: 522-538, 2016.
- [54] V. Renaudin, M. H. Afzal, and G. Lachapelle, Complete triaxis magnetometer calibration in the magnetic domain. *Journal of Sensors* 2010: 967245, 2010.
- [55] M. Glueck, D. Oshinubi, P. Schopp, and Y. Manoli, Real-time autocalibration of MEMS accelerometers. *IEEE Transactions on Instrumentation and Measurement* 63(1): 96-105, 2014.
- [56] T. Beravs, J. Podobnik, and M. Munih, Three-axial accelerometer calibration using Kalman filter covariance matrix for online estimation of optimal sensor orientation. *IEEE Transactions on Instrumentation and Measurement* 61(9): 2501-2511, 2012.
- [57] Y. Li, X. Niu, Q. Zhang, H. Zhang, and C. Shi, An in situ hand calibration method using a pseudo-observation scheme for low-end inertial measurement units. *Measurement Science and Technology* 23(10): 105104, 2012.
- [58] H. Zhang, Y. Wu, W. Wu, M. Wu, and X. Hu, Improved multi-position calibration for inertial measurement units. *Measurement Science and Technology* 21(1): 015107, 2010.
- [59] M. Sipos, P. Paces, J. Rohac, and P. Novacek, Analyses of triaxial accelerometer calibration algorithms. *IEEE Sensors Journal* 12(5): 1157-1165, 2012.
- [60] M. Gietzelt, K.-H. Wolf, M. Marschollek, and R. Haux, Performance comparison of accelerometer calibration algorithms based on 3D-ellipsoid fitting methods.

- Computer Methods and Programs in Biomedicine* 111(1): 62-71, 2013.
- [61] S. Stancin, and S. Tomazic, Time- and computation-efficient calibration of MEMS 3D accelerometers and gyroscopes. *Sensors* 14(8): 14885-14915, 2014.
- [62] W. T. Fong, S. K. Ong, and A. Y. C. Nee, Methods for in-field user calibration of an inertial measurement unit without external equipment. *Measurement Science and Technology* 19(8): 085202, 2008.
- [63] Y. Wu, and L. Pei, Gyroscope calibration via magnetometer. *IEEE Sensors Journal* 17(16): 5269–5275, 2017.
- [64] Z. Wu, and W. Wang, Magnetometer and gyroscope calibration method with level rotation. *Sensors* 18(3): 748, 2018.
- [65] J. F. Vasconcelos, G. Elkaim, C. Silvestre, P. Oliveira, and B. Cardeira, Geometric approach to strapdown magnetometer calibration in sensor frame. *IEEE Transactions on Aerospace and Electronic Systems* 47(2): 1293-1306, 2011.
- [66] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, Feature reduction and selection for EMG signal classification. *Expert Systems with Applications* 39(8): 7420-7431, 2012.
- [67] S. Cheung, and P. Varaiya, *Traffic surveillance by wireless sensor networks: Final report*. University of California, Berkeley, 2007.
- [68] Y.-L. Chen, B.-F. Wu, H.-Y. Huang, and C.-J. Fan, A real-time vision system for nighttime vehicle detection and traffic surveillance. *IEEE Transactions on Industrial Electronics* 58(5): 2030-2044, 2011.
- [69] Z. Sun, G. Bebis, and R. Miller, On-road vehicle detection: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(5): 694-711, 2006.
- [70] S. Sivaraman, and M. M. Trivedi, Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Transactions on Intelligent Transportation Systems* 14(4): 1773-1795, 2013.
- [71] J. Gajda, and M. Stencel, A highly selective vehicle classification utilizing dual-loop inductive detector. *Metrology and Measurement Systems* 21(3): 473–484, 2014.
- [72] S. Meta, and M. Cinsdikici, Vehicle-classification algorithm based on component analysis for single-loop inductive detector. *IEEE Transactions on Vehicular Technology* 59(6): 2795–2805, 2010.
- [73] J. Gajda, and M. Mielczarek, Automatic vehicle classification in systems with single inductive loop detector. *Metrology and Measurement Systems* 21(4): 619–

- 630, 2014.
- [74] W. Balid, H. Tafish, and H. Refai, Intelligent vehicle counting and classification sensor for real-time traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems* 19(6): 1784-1794, 2018.
- [75] D. Liu, X. Xu, C. Fei, W. Zhu, X. Liu, G. Yu, and G. Fang, Direction identification of a moving ferromagnetic object by magnetic anomaly. *Sensors and Actuators A: Physical* 229: 147-153, 2015.
- [76] J. Lan, Y. Xiang, L. Wang, and Y. Shi, Vehicle detection and classification by measuring and processing magnetic signal. *Measurement* 44(1): 174–180, 2011.
- [77] N. Wahlstrom, R. Hostettler, F. Gustafsson, and W. Birk, Classification of driving direction in traffic surveillance using magnetometers. *IEEE Transactions on Intelligent Transportation Systems* 15(4): 1405–1418, 2014.
- [78] Q. Zhu, G. Tong, B. Li, and X. Yuan, A practicable method for ferromagnetic object moving direction identification. *IEEE Transactions on Magnetics* 48(8): 2340–2345, 2012.
- [79] J. Smidla, and G. Simon, Accelerometer-based event detector for low-power applications. *Sensors* 13(10): 13978–13997, 2013.
- [80] R. Hostettler, and P. Djuric, Vehicle tracking based on fusion of magnetometer and accelerometer sensor measurements with particle filtering. *IEEE Transactions on Vehicular Technology* 64(11): 4917–4928, 2015.
- [81] W. Ma, D. Xing, A. McKee, R. Bajwa, C. Flores, B. Fuller, and P. Varaiya, A wireless accelerometer-based automatic vehicle classification prototype system. *IEEE Transactions on Intelligent Transportation Systems* 15(1): 104–111, 2014.
- [82] D. Kleyko, R. Hostettler, W. Birk, and E. Osipov, Comparison of machine learning techniques for vehicle classification using road side sensors. In: *Proc. of IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Las Palmas, Spain, pp. 572–577, 2015.
- [83] R. Wang, L. Zhang, K. Xiao, R. Sun, and L. Cui, EasiSee: real-time vehicle classification and counting via low-cost collaborative sensing. *IEEE Transactions on Intelligent Transportation Systems* 15(1): 414-424, 2014.
- [84] Z. Zhang, M. Tao, and H. Yuan, A parking occupancy detection algorithm based on AMR sensor. *IEEE Sensors Journal* 15(2): 1261–1269, 2015.
- [85] H. Zhu, and F. Yu, A vehicle parking detection method based on correlation of magnetic signals. *International Journal of Distributed Sensor Networks* 2015: 361242, 2015.

- [86] H. Zhu, and F. Yu, A cross-correlation technique for vehicle detections in wireless magnetic sensor network. *IEEE Sensors Journal* 16(11): 4484–4494, 2016.
- [87] S. Taghvaeeyan, and R. Rajamani, Portable roadside sensors for vehicle counting, classification, and speed measurement. *IEEE Transactions on Intelligent Transportation Systems* 15(1): 73-83, 2014.
- [88] E. Sifuentes, O. Casas, and R. Pallas-Areny, Wireless magnetic sensor node for vehicle detection with optical wake-up. *IEEE Sensors Journal* 11(8): 1669-1676, 2011.
- [89] V. Markevicius, D. Navikas, M. Zilyys, D. Andriukaitis, A. Valinevicius, and M. Cepenys, Dynamic vehicle detection via the use of magnetic field sensors. *Sensors* 16(1): 78, 2016.
- [90] G. Burrelli, and R. Giorgi, A field experience for a vehicle recognition system using magnetic sensors. In: *Proc. of Mediterranean Conference on Embedded Computing (MECO)*, Budva, Montenegro, pp. 178-181, 2015.
- [91] S. Vancin, and E. Erdem, Implementation of the vehicle recognition systems using wireless magnetic sensors. *Sadhana* 42(6): 841–854, 2017.
- [92] B. Yang, and Y. Lei, Vehicle detection and classification for low-speed congested traffic with anisotropic magnetoresistive sensor. *IEEE Sensors Journal* 15(2): 1132-1138, 2015.
- [93] H. Dong, X. Wang, C. Zhang, R. He, L. Jia, and Y. Qin, Improved robust vehicle detection and identification based on single magnetic sensor. *IEEE Access* 6: 5247-5255, 2018.
- [94] Q. Wang, J. Zheng, H. Xu, B. Xu, and R. Chen, Roadside magnetic sensor system for vehicle detection in urban environments. *IEEE Transactions on Intelligent Transportation Systems* 19(5): 1365-1374, 2018.
- [95] H. Tafish, W. Balid, and H. Refai, Cost effective vehicle classification using a single wireless magnetometer. In: *Proc. of International Wireless Communications and Mobile Computing Conference (IWCMC)*, Paphos, Cyprus, pp. 194-199, 2016.
- [96] W. Zhang, G. Tan, N. Ding, Y. Shang, and M. Lin, Vehicle classification algorithm based on binary proximity magnetic sensors and neural network. In: *Proc. of IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Beijing, China, pp. 145-150, 2008.
- [97] V. Velisavljevic, E. Cano, V. Dyo, and B. Allen, Wireless magnetic sensor network for road traffic monitoring and vehicle classification. *Transport and Telecommunication* 17(4): 274-288, 2016.

- [98] C. Xu, Y. Wang, X. Bao, and F. Li, Vehicle classification using an imbalanced dataset based on a single magnetic sensor. *Sensors* 18(6): 1690, 2018.
- [99] H. Li, H. Dong, L. Jia, and M. Ren, Vehicle classification with single multi-functional magnetic sensor and optimal MNS-based CART. *Measurement* 55: 142-152, 2014.
- [100] S. Kaewkamnerd, J. Chinrungrueng, R. Pongthornseri, and S. Dumnin, Vehicle classification based on magnetic sensor signal. In: *Proc. of International Conference on Information and Automation (ICIA)*, Harbin, China, pp. 935-939, 2010.
- [101] Y. He, Y. Du, L. Sun, and Y. Wang, Improved waveform-feature-based vehicle classification using a single-point magnetic sensor. *Journal of Advanced Transportation* 49(5): 663-682, 2015.

Acknowledgments

First of all, I would like to express my gratitude to my supervisor, Dr. Szilveszter Pletl, for his professional support throughout my studies, and providing me with useful comments regarding this thesis. Furthermore, I would like to thank my co-authors, Dr. Zoltán Kincses and László Schäffer, for their contributions to the achieved results.

I would also like to thank my friends and colleagues for their support and encouragement over the years.

A am very grateful to Selma Ltd., Subotica, Serbia for the technical resources and support during the research in vehicle classification.

Last, but not least, I would like to thank my wife, Mónika, and my mother, Éva, for their continuous support and constant encouragement without which I would have never been able to complete my thesis or achieve my goals.

Parts of this work were supported by the European Union and co-funded by the European Social Fund. Project title: "Telemedicine-focused research activities on the field of Mathematics, Informatics and Medical sciences" Project number: TÁMOP-4.2.2.A-11/1/KONV-2012-0073.

Summary

Due to the advances in technology, accelerometers, gyroscopes, and magnetometers became widely used in a large variety of applications.

These sensors are most commonly used in Inertial Navigation Systems (INS) to obtain information about the position, the orientation, and the speed of a moving object. All three sensor types can be also found in consumer electronics, such as smartphones, tablets, or smartwatches, due to their small size, light weight, low cost, and low power consumption.

Inertial and magnetic sensors are widely used in different pattern recognition applications. Often, they are utilized in these applications in the form of Wireless Sensor Networks (WSN). The necessary computation in these systems can be done offline or online. An external unit is applied in the case of offline computation, while in online applications, the algorithms run on the used embedded systems. The design of online applications is a challenging task, since they require appropriate usage of energy, memory, and processing.

In this thesis, I presented my research results achieved in two pattern recognition applications of inertial and magnetic sensors. The developed algorithms are easily implementable on the used microcontroller-based embedded systems, thus, they are capable of running online.

The first application deals with human motion classification. Such systems are widely used in different industrial, entertainment, health, and medical applications.

In the proposed system, accelerometers, gyroscopes, and magnetometers are attached to the wrists of the subjects and are used to classify different movements of the body and the arms. The proposed algorithms utilize features extracted from the signals, which are generated by the changes in the position and the orientation of the sensors. I applied only time-domain features (TDFs) with low memory requirements, and extracted features in processing windows of fixed width. I tested various datasets, which were generated based on different sampling frequencies, processing window widths, feature extraction modes, and used sensor types, to explore the capabilities of the proposed system, and to find the optimal configuration. All defined processing window sizes were in millisecond domain. Features were extracted using the measurements on the three sensor axes separately and the computed magnitude. I also tested a proposed aggregation-based feature reduction method. The three sensor types were tested separately, in pairs, and altogether, to investigate the impact of the sensors on recognition efficiency, and to prevent unnecessary usage of memory and hardware resources. Decreasing the number of applied sensor types can also lower the

overall cost and power consumption of the system. I examined the recognition efficiency, the memory consumption, and the training time of seven classification methods with and without using Linear Discriminant Analysis (LDA)-based dimension reduction before the classifiers. The applied classification methods were the following: Nearest Centroid Classifier (NCC), Multi-Layer Perceptron (MLP), Naïve Bayes Classifier (NBC), “one-versus-one” (OvO) and “one-versus-all” (OvA) Support Vector Machines (SVM), k-Nearest Neighbor (k-NN), Classification Tree (CT). I also compared the capabilities of TDFs and FDFs in the proposed system.

The second application utilizes magnetometers to achieve vehicle classification. Automatic vehicle detection technologies play an important role in Intelligent Transportation Systems (ITS). These systems provide vehicle count and classification data, which are very important inputs for traffic modelling, pavement design, transportation planning, emission/pollution estimation, and traffic control. The most widely used technologies are inductive loops and Video Image Processing (VIP)-based systems, but magnetometer-based vehicle detectors have many advantages compared to these technologies.

Vehicle classification can be realized by using two sensor units, which allows the computation of vehicle speed and length, or by applying only one sensor unit, but these systems must rely only on magnetic signatures in the classification process.

The proposed system uses measurements from only a single magnetometer to classify vehicles into multiple classes. The sensor is mounted into the pavement surface, thus, it is in stationary position. The distortions in the measured magnetic field caused by vehicles passing above the sensor are used during the extraction of features. The system uses a proposed adaptive threshold-based vehicle detection algorithm. The utilized features for classification were extracted in the detection windows. I tested different datasets based on different feature extraction modes and combinations of used sensor axes to minimize the number of used features and the possible cost of the system. Beside using features computed using the signals on the sensor axes, signals of absolute values and different aggregation modes were also tested. The applied aggregation modes include magnitudes from the origin, angles, and magnitudes from the calibration point. All three types were tested in the planes defined by the sensor axes, while the magnitudes were also computed in three-dimensional space. The effect of the detection length in the feature set was also examined. Since vehicles with high metallic content passing in the neighboring lanes can cause false detection, a rule-based false detection filtering algorithm was also proposed. I tested different parameter types in single and complex rules. The applied parameters were the highest and lowest values extracted in the detection window using different signals. The applied signal types were the signals on the three measurements axes, and magnitudes and angles in the planes defined by the sensor axes. The optimization of the parameter values was realized using genetic algorithms.

Thesis summary

Thesis Group 1. I developed novel online algorithms for the classification of human body and arm movements using wrist-mounted accelerometers, gyroscopes and magnetic sensors. The algorithms are implementable on resource constrained embedded systems.

Thesis 1.1. I developed a prototype measurement system to record human movements and collected measurement data for training and validation of the classifiers. I used two nine degrees of freedom (9DoF) sensor boards mounted on WSN motes, which were attached to the wrists of the subjects.

Thesis 1.2. I developed a new online movement classification algorithm where the features extracted on multiple units are used together in the classification stage.

I showed that the use of only time-domain features (TDFs) with low memory requirements is sufficient to obtain high classification efficiencies in movement recognition systems using inertial and magnetic sensors.

I showed that there is a high correlation between the movements of different subjects, since the classifiers provided very high recognition rates on measurements of subjects, which were not used during the training.

I showed that millisecond domain in processing window widths is sufficient for efficient recognition of human body and arm movements.

I proposed an aggregation-based feature reduction method for sensors with multiple measurement axes, which results in three-times less features than when the features computed for each sensor axis are used as inputs of the classifiers. The aggregation is done by calculating a linear combination of the feature values computed for each axis for a specific feature type. The aggregation method can provide useful classification efficiencies. This reduction method can also help the system to be less sensitive to differences in orientations of the sensors on the arms.

I showed that adding a new sensor type can in some cases only slightly increase or even decrease the overall recognition rate of the system. With testing the possible sensor types in different combinations, an optimal configuration can be found, which can lower the overall cost and power consumption of the system.

I showed that the Multi-Layer Perceptron (MLP) neural networks are the optimal solution for online algorithms in similar applications, and that some popular methods (SVM, k-NN) are not suitable for online use in resource constrained embedded systems.

I showed that the use of LDA-based dimension reduction on the feature sets can improve recognition efficiency, training time, and memory requirements for implementation of different classifiers.

I showed that using only simple TDFs even higher recognition rates can be achieved for some setups than when only features extracted using frequency-domain analysis are applied.

I showed that the proposed aggregation-based feature reduction can be also effectively applied in the case of frequency-domain features (FDFs).

Thesis 1.3. I developed a hierarchical-distributed algorithm for online human movement classification, where the multiple units determine their own movement classes, and one unit combines the resulted classes to determine the movement class of the entire body. This algorithm requires less computation because of the smaller classifiers and less wireless communication, which improves energy efficiency. The addition of new units to the system would be also easier using this approach. I applied two different movement hierarchies, which provide similar results.

Thesis Group 2. I developed a new real-time vehicle classification system based on a single magnetic sensor mounted into the surface of the pavement.

Thesis 2.1. I developed a prototype measurement system with a single magnetometer-based unit installed into the pavement surface. I also used a camera mounted on the side of the road to validate the detections and to determine the class of the samples. I collected a very large database of samples using the measurement system.

Thesis 2.2. I developed a vehicle detection algorithm based on adaptive thresholds. The algorithm can deal with the environmental changes and is mainly designed for vehicle classification purposes. The algorithm estimates the offsets on each sensor axis using a calibration method after the unit is turned on, and follows the environmental changes by continuous recalibration. The algorithm is also capable of detecting vehicles with trailers.

Thesis 2.3. I developed a new online vehicle classification algorithm, which can classify vehicles into nine vehicle classes using MLP neural networks. The algorithm uses TDFs, which can potentially detect vehicle axles and overall metallic content, are immune to speed variations, and require low memory usage.

I showed that a single magnetometer-based system is sufficient for classifying vehicles into nine vehicle classes.

I showed that using aggregated data in the form of angles or magnitudes, can even increase the classification efficiencies besides decreasing the number of inputs.

I showed that adding the detection length to the feature set, which is a speed-dependent feature, only slightly increases the classification efficiencies. This shows that the used features can effectively extract the information from the waveforms.

Thesis 2.4. I developed a rule-based algorithm for the filtering of false detections caused by vehicles with high metallic content passing in the neighboring lane. This can further increase the overall efficiency of the proposed vehicle classification system.

I showed that using proper rules, a very high percentage of false detections can be filtered with losing only a few good detections.

I showed that despite that the axis pointing to neighboring lane should carry most information for the filtering of false detections, the other two axes provide more reliable parameters.

I showed that aggregated data in the form of magnitudes can provide higher filtering rates than parameters extracted from the signals on the sensor axes.

Összefoglaló

A technológia fejlődésének köszönhetően a gyorsulásmérők, giroszkópok, és magnetométerek különböző alkalmazásokban való széleskörű felhasználása egyre elterjedtebbé vált.

A leggyakoribb felhasználása ezen szenzoroknak az inerciális navigációs rendszerek, amelyekben pozíció, orientáció, és sebesség meghatározásához szolgáltatnak információt. A kis méretüknek, illetve alacsony tömegüknek, áruknak, és fogyasztásuknak köszönhetően mindhárom szenzor típus megtalálható számos fogyasztói elektronikákban, mint az okostelefonok, táblagépek, vagy okosórák.

Az inerciális és mágneses szenzorok széles körben használtak különböző mintafelismerési alkalmazásokban. Gyakran vezeték nélküli szenzorhálózatok (WSN) formájában kerülnek felhasználásra az alkalmazásokban. A szükséges számítások ezekben a rendszerekben történhet offline vagy online. Offline esetben egy külső egység végzi a számításokat, míg az online alkalmazások esetén a felhasznált beágyazott rendszereken futnak az algoritmusok. Az online algoritmusok tervezése egy kihívást jelentő feladat, mivel ügyelni kell a megfelelő felhasználásra energia, memória, és feldolgozás terén.

Munkámban bemutattam az inerciális és mágneses szenzorok két mintafelismerési alkalmazásában elért eredményeimet. A kifejlesztett algoritmusok könnyen implementálhatóak a felhasznált mikrovezérlő alapú beágyazott rendszerekre, tehát képesek online működésre.

Az első alkalmazás emberi mozgások osztályozásával foglalkozik. Ilyen rendszerek széles körben felhasználhatóak különböző ipari, szórakoztató, egészségügyi és orvosi alkalmazásokban.

A javasolt rendszerben az alanyok csuklóihoz rögzített gyorsulásmérőket, giroszkópokat és magnetométereket használtam a test és karok különböző mozgásainak osztályozására. A javasolt algoritmusok a szenzorok pozíció és orientáció változásainak következtében keletkezett jelekből kinyert jellemzőket használnak fel. Kizárólag alacsony memóriagényű időtartománybeli jellemzőket alkalmaztam, amelyeket fix szélességű ablakokban nyertem ki. Annak érdekében, hogy megvizsgáljam a javasolt rendszer képességeit és megtaláljam az optimális konfigurációt, különböző adathalmazokat teszteltem, amelyek különböző mintavételezési frekvenciák, ablakméretek, jellemző kinyerési módszerek, és felhasznált szenzor típusok alapján lettek létrehozva. Minden alkalmazott ablakméret milliszekundum tartományú volt. A jellemzők kinyeréséhez felhasználásra kerültek a mérési értékek az egyes

tengelyeken, illetve a kiszámított vektor hosszak. Teszteltem egy javasolt aggregáció alapú módszert is, amely segítségével csökkenthető a jellemzők száma. Annak érdekében, hogy megvizsgáljam az érzékelők hatását a felismerési hatékonyságra és megelőzzem a memória és a hardver erőforrások felesleges használatát, a három szenzor típust teszteltem külön-külön, párokban, és együtt. Az érzékelő típusok csökkentésével egyben csökkenthetjük a rendszer árát és fogyasztását. Megvizsgáltam hét osztályozási módszer felismerési hatékonyságát, memóriafelhasználását, és tanítási idejét lineáris diszkriminancia-analízis alapú dimenziócsökkentés alkalmazásával és anélkül az osztályozók előtt. A felhasznált osztályozási módszerek a következők: legközelebbi centroid osztályozó, többretegű perceptron, naiv Bayes módszer, "egy az egy ellen" és "egy mindenki ellen" szupport vektor gépek, k-legközelebbi szomszéd módszer, osztályozási fák. A javasolt rendszerben egy összehasonlítást is végeztem az idő- és frekvenciatartománybeli jellemzők képességei között.

A második alkalmazás magnetométereket használ fel járműosztályozáshoz. Az automatikus járműérzékelési technológiák fontos szerepet játszanak az intelligens közlekedési rendszerekben. Ezek a rendszerek adatokat biztosítanak a járművek számáról és osztályáról, amelyek nagyon fontos bemeneti szempontjai a forgalom modellezésnek, az útburkolat kialakításának, a közlekedéstervezésnek, a szennyezés becslésének, és a forgalomirányításnak. A leggyakrabban alkalmazott technológiák az induktív hurkok és a képfeldolgozó detektorok, viszont a mágneses szenzor alapú járműérzékelőknek számos előnyük van ezekkel a technológiákkal szemben.

A járműosztályozás megvalósítható két szenzor alkalmazásával, amely lehetővé teszi a jármű sebességének és hosszának meghatározását, vagy csak egy eszköz felhasználásával, viszont ezek a rendszerek csak a jelalakokra hagyatkozhatnak a klasszifikálási folyamat során.

A javasolt rendszer csak egy magnetométert alkalmaz a járművek több osztályba történő osztályozására. Az érzékelő az úttest felületébe van szerelve, tehát stacionárius pozícióban van. A jellemzők kinyerése során a mért mágneses mezőben kialakuló disztorziók vannak felhasználva, amelyek a járművek szenzor feletti mozgása miatt jönnek létre. A rendszer egy javasolt adaptív küszöbököt alkalmazó járműdetektáló algoritmust használ. A klasszifikálás során felhasznált jellemzők az érzékelési ablakokban lettek kinyerve. A felhasznált jellemzők és a rendszer lehetséges árának minimalizálása érdekében különböző adathalmazokat teszteltem, amelyek különböző jellemző kinyerési módszerek és alkalmazott szenzor tengelyek alapján lettek generálva. A szenzorok tengelyein mért jeleken kívül az abszolút értékek jeleit és különböző aggregációs módokat is teszteltem. Az alkalmazott aggregációs módok a következők: origótól számított vektor hosszak, szögek, kalibrációs ponttól számított vektor hosszak. Mindhárom típus a szenzorok tengelyei által meghatározott síkokban lett tesztelve, illetve a vektor hosszak három dimenziós térben is meg lettek határozva. Az érzékelés hosszának az adathalmazokra gyakorolt hatását is

megvizsgáltam. Mivel a szomszédos sávban elhaladó magas fémtartalmú járművek fals detekciókat okozhatnak, egy szabályalapú fals detekció szűrő módszert is javasoltam. Különböző paramétereket teszteltem egyetlen valamint összetett szabályok esetén is. A detekciós ablakokban különböző jelek esetén kinyert legmagasabb és legalacsonyabb mérési értékeket használtam fel paraméterekként. Az alkalmazott jeltípusok a mérési értékek a szenzor három tengelyén és a szenzorok tengelyei által meghatározott síkokban kiszámított vektor hosszak és szögek voltak. A paraméter értékek optimalizálása genetikus algoritmusokkal lett megvalósítva.

Tézisek összefoglalása

1. Téziscsoport Újszerű online algoritmusokat fejlesztettem emberi test és karok mozgásának osztályozásához csuklóra szerelt gyorsulásmérők, giroszkópok, és magnetométerek felhasználásával. Az algoritmusok implementálhatóak korlátos erőforrásokkal rendelkező beágyazott rendszerekre.

1.1. Tézis Kifejlesztettem egy prototípus mérési rendszert emberi mozgások rögzítésére, és mérési adatokat gyűjtöttem az osztályozók tanításához és validálásához. Két WSN motekra szerelt kilenc szabadságfokú szenzorlapot alkalmaztam, melyek az alanyok csuklóihoz voltak rögzítve.

1.2. Tézis Kifejlesztettem egy új online mozgásosztályozó algoritmust amely a különböző egységeken kinyert jellemzőket együtt használja fel az osztályozási fázisban.

Megmutattam, hogy kizárólag alacsony memóriai igényű időtartománybeli jellemzők alkalmazása elegendő magas klasszifikálási hatékonyság eléréséhez inerciális és mágneses szenzorokat alkalmazó mozgásosztályozó rendszerekben.

Megmutattam, hogy különböző alanyok mozgásaiban magas a korreláció, mivel az osztályozók magas felismerési arányokat szolgáltatnak olyan alanyok adatai esetén amelyek nem voltak alkalmazva tanítás során.

Megmutattam, hogy milliszekundum tartományú ablakszélességek elegendőek az emberi test és karok mozgásainak hatékony felismeréséhez.

Javaslatot tettem egy aggregáció alapú módszerre, amellyel több tengelyű szenzorok esetén csökkenthető a jellemzők száma. A módszer alkalmazása háromszor kevesebb jellemzőt eredményez annál az esetnél, amikor a szenzor egyes tengelyein kinyert jellemzők kerülnek az osztályozók bemeneteire. Az aggregáció az egyes tengelyeken kinyert jellemző értékek lineáris kombinációjának kiszámításával történik adott jellemzőtípus esetén. Az aggregációs módszer alkalmazható osztályozási hatékonyságokat biztosít. Ez a redukciós módszer segíthet a rendszernek abban is, hogy kevésbé legyen érzékeny a szenzorok kisebb orientációbeli eltérésére.

Megmutattam, hogy új szenzortípus hozzáadása a rendszerhez egyes esetekben csak kis mértékben javítja vagy akár csökkenti a rendszer felismerő képességét. A lehetséges

szenzortípusok különböző kombinációkban történő tesztelésével egy optimális konfiguráció található, amely csökkentheti a rendszer árát és fogyasztását.

Megmutattam, hogy hasonló alkalmazások esetén a többrétegű perceptron neurális hálózatok a legalkalmasabb osztályozók online algoritmusok megvalósításához, illetve, hogy néhány népszerű módszer (szupport vektor gépek, k-legközelebbi szomszéd módszer) nem alkalmas online felhasználásra korlátos erőforrású beágyazott rendszerek esetén.

Megmutattam, hogy a lineáris diszkriminancia-analízis alapú dimenziócsökkentés alkalmazása a jellemzőket tartalmazó vektoron javíthatja különböző osztályozók felismerési hatékonyságát, tanítási idejét, és az implementáláshoz szükséges memóriaigényüket.

Megmutattam, hogy csak egyszerű időtartománybeli jellemzők alkalmazásával egyes adathalmazok esetén magasabb klasszifikációs arány érhető el mint csak frekvenciatartomány alapú analízis esetén kinyert jellemzők felhasználásával.

Megmutattam, hogy a javasolt aggregáció alapú csökkentése a jellemzők számának frekvenciatartománybeli jellemzők esetén is hatékonyan alkalmazható.

1.3. Tézis Kifejlesztettem egy hierarchikus-elosztott algoritmust emberi mozgások online osztályozására, amelyben a több felhasznált eszköz meghatározza a saját mozgásoztályát, majd egy eszköz ezen osztályokat kombinálva meghatározza a teljes test mozgásoztályát. Ez az algoritmus kevesebb számítást igényel az osztályozók kisebb mérete miatt, illetve kevesebb vezeték nélküli kommunikációt igényel, amely az energiahatékonyság szempontjából jelentős. Ez a módszer alkalmazása leegyszerűsíti az új egységek rendszerhez való hozzáadását is. Két különböző mozgáshierarchiát alkalmaztam, amelyek hasonló hatékonyságokat eredményeztek.

2. Téziscsoport Kifejlesztettem egy valós idejű járműosztályozó rendszert, amely egy darab úttestbe szerelt mágneses szenzort alkalmaz.

2.1. Tézis Kifejlesztettem egy prototípus mérési rendszert, amelyben egy magnetométer alapú egység lett szerelve az úttest felszínébe. Az érzékelések validációja és az egyes minták osztályának megállapítása érdekében egy kamerát is alkalmaztam, amely az út szélére lett telepítve. A mérési rendszer segítségével nagyon nagy adatbázist alkottam az összegyűjtött mintákból.

2.2. Tézis Kifejlesztettem egy adaptív küszöbököt alkalmazó járműérzékelési algoritmust. Az algoritmus kezelni tudja a környezeti változásokat, és főleg járműosztályozási alkalmazásokhoz lett tervezve. Az algoritmus egy kalibrációs módszert alkalmaz bekapcsolás után, amely segítségével becsli az eltolásokat az egyes szenzortengelyeken, majd állandó újralibrációval követi a környezetben fellépő változásokat. Az algoritmus alkalmas utánfutós járművek detektálására is.

2.3. Tézis Kifejlesztettem egy új online járműosztályozó algoritmust, amely többrétegű neurális hálózatok használatával képes a járműveket kilenc osztályba besorolni. Az

algoritmus időtartománybeli jellemzőket alkalmaz, amelyek potenciálisan érzékelni tudják a járművek fémtartalmát és tengelyeinek számát, immunisak a sebességváltozásokra, és alacsony a memóriahasználatuk.

Megmutattam, hogy egy egyetlen magnetométeren alapuló rendszer is elegendő a járművek kilenc osztályba sorolásához.

Megmutattam, hogy vektor hosszak és szögek formájában aggregált adatok segítségével akár a felismerési arány is növelhető a bemenetek számának csökkentése mellett.

Megmutattam, hogy a detekció hosszának, amely egy sebességfüggő jellemző, hozzáadása a jellemzőkből álló vektorhoz csak kis mértékben növeli az osztályozási hatékonyságot. Ez mutatja, hogy a felhasznált jellemzők képesek az információ hatékony kinyerésére a jelalakokból.

2.4. Tézis Kifejlesztettem egy szabályalapú algoritmust a szomszédos sávban elhaladó nagy fémtartalmú járművek által okozott fals detekciók szűrésére. Ennek az alkalmazása tovább növelheti a javasolt járműosztályozó rendszer általános hatékonyságát.

Megmutattam, hogy megfelelő szabályok alkalmazásával nagy százalékban kiszűrhetőek a fals érzékelések csupán néhány helyes detekció elvesztésével.

Megmutattam, hogy annak ellenére, hogy a szomszédos sáv felé mutató tengely kellene, hogy a legtöbb információt hordozza a fals érzékelések szűréséhez, a másik két tengely sokkal megbízhatóbb paramétereket biztosít.

Megmutattam, hogy a vektor hosszak alakjában aggregált adatok sokkal magasabb szűrési hatékonyságot biztosító paramétereket szolgáltatnak mint az érzékelő tengelyeinek jelein kinyert paraméterek.