Authentication of Smartphone Users Based on the Way They Walk Using k-NN Algorithm

Claudia Nickel, Tobias Wirtl, Christoph Busch
Hochschule Darmstadt (CASED)
Darmstadt, Germany
Email: claudia.nickel@h-da.de

***Abstract*—Accelerometer-based biometric gait recognition offers a convenient way to authenticate users on their mobile devices. Modern smartphones contain in-built accelerometers which can be used as sensors to acquire the necessary data while the subjects are walking. Hence, no additional costs for special sensors are imposed to the user. In this publication we extract several features from the gait data and use the k-Nearest Neighbour algorithm for classification. We show that this algorithm yields a better biometric performance than the machine learning algorithms we previously used for classification, namely Hidden Markov Models and Support Vector Machines. We implemented the presented method on a smartphone and demonstrate that it is efficient enough to be applied in practice.**

***Keywords*-biometrics; gait recognition; smartphone; accelerometer**

I. Introduction

The focus of this publication is the unobtrusive authentication on smartphones via accelerometer-based biometric gait recognition. Such an authentication method enables the mobile phone to recognize its owner based on the way he is walking. There are two main advantages of this approach. First, gait can be captured via acceleration sensors, which are already integrated into smartphones. Hence, there are no additional hardware costs for deploying this method. Second, gait recognition does not require explicit user interaction during verification as the phone does it literally “on-the-go”. These two factors make accelerometer-based biometric gait recognition a very user friendly method, which does not require extra interaction time.

This overcomes the problems of conventional authentication methods on mobile devices. Currently, most phones offer only authentication via PIN. Breitinger et al. [1] present results of a survey, showing that most owners of mobile devices do not activate the PIN authentication. The main reason for this is the low user-friendliness because of a high time-consumption.

Current statistics further emphasize the relevance of this topic. The number of mobile phone subscribers increased to more than five billion in 2010 [2] and nearly 70% of the consumers are using smartphones [3]. In addition, the computational power of smartphones is steadily increasing, resulting in a wide variety of different applications. Some of these applications, like online banking, require the user to authenticate himself. Other applications store various kinds of data in the phone. Because this data often contains sensitive private or business information the phone itself should be protected by an authentication method. Mobile phones can be easily lost or forgotten and they are an facile target for thefts. When no authentication is required, the attacker can effortlessly analyze the data on the phone or use stored passwords to access e-mail or social network accounts.

Research on accelerometer-based biometric gait recognition was started by Ailisto et al. resulting in the first publication on this topic in 2005 [4]. The technique was further developed and analyzed by Gafurov [5]. In these initial publications, dedicated prototype accelerometers were used, which were attached to the hip, leg, arm or ankle of the subjects. With the fast development of smartphones containing in-built accelerometers, those were used as sensors for data collection [6], [7], [8]. Nevertheless, the data processing and classification is in nearly all cases performed on a PC.

In a previous publication [9] we showed, that current smartphones have enough computational power to perform the extraction and comparison of gait cycles directly on the phone. Because machine learning algorithms yielded lower error rates than cycle-based approaches [10], the next step is to see how these perform on the phone. The contribution of this paper is twofold. The k-nearest Neighbour (k-NN) algorithm is comprehensively evaluated on a database collected using a off-the-shelf smartphone. The biometric performance is compared to the one obtained when evaluating Support Vector Machines (SVMs) and Hidden Markov Models (HMMs) on the same database [11]. In addition, the algorithm is implemented on a smartphone and it is shown that the complexity of feature extraction and comparison is low enough to be applicable in practice.

The remaining part of the paper is structured as follows. The data collection is described in the following section. Afterwards the preprocessing and feature extraction steps are presented. Section V presents the evaluation. The results are compared to the ones obtained using further classifiers in section VI. The feasibility of executing the proposed method completely on a smartphone is shown in section VII, followed by conclusions and proposals of future work in section VIII.

II. Data Collection

Gait data of 36 subjects was collected in two different sessions. Age and gender distribution are given in Table I. These two sessions were on average 24 days apart (min = 1, max = 125, median = 10.5). At each session the subjects had to

978-0-7695-4712-1/12 $26.00 © 2012 IEEE
DOI 10.1109/IIH-MSP.2012.11

walk up and down a hallway on flat carpet. The data collected between starting to walk and stopping at the other end of the hall is called *walk*. In each session, 12 normal walks, 16 fast walks and again 12 normal walks were collected of each subject, later referred to as setting 1, 2, and 3. This resulted in around 15 minutes of walking data for each subject in each session and a total amount of nearly 20 hours of gait data.

The subjects carried a standard cell phone (Motorola Milestone using Android OS) in a pouch attached to the right hip. The phone contains built-in accelerometers which measure accelerations in three directions (x-, y- and z-acceleration). The phone was positioned in such a way that the vertical acceleration is measured on the x-axis, the forward-backward acceleration on the y-axis and the z-axis points sideways. An Android application was written to access the accelerometer data and write it to text files.

	20-24	25-29	30-39	40-49	50+	total
male	3	18	7	1	0	29
female	1	4	2	0	0	7
total	4	22	9	1	0	36

TABLE I
AGE AND GENDER DISTRIBUTION OF PARTICIPANTS OF DATA COLLECTION.

III. PREPROCESSING

Before gait features can be extracted, the data has to be preprocessed applying the following steps. Walks have to be extracted, because the stored files contain also data were the subjects are not walking. An automatic extraction is followed by a visual inspection, and the start and end of the walk are manually corrected if necessary. A product implementation should incorporate activity recognition to determine the relevant data where the subject is walking.

The data is linearly interpolated to a fixed sampling rate. The collected data does not have such, due to a limitation of the Android API. Acceleration values can only be obtained on sensor changes, which are potentially delayed by background processes. The mean sampling rate of our databases is around 127 data values per second. Different sampling rates (upsampling and downsampling) are evaluated. The reason for upsampling was to prevent loss of too many data values. E.g. peaks might disappear if their original time stamp is between two time values used for interpolation. Downsampling has the advantage of data reduction.

Because the mobile phone is not well calibrated, the acceleration it measures in a stable position (no movement) is not exactly zero or gravity, as it should be in vertical direction, and it is not stable over time. To reduce the influence of this phenomenon, the data is centered around zero. This is done by subtracting the mean of the acceleration values of the walk from the data.

Afterwards, the data is divided into segments of a fixed time length. Segment lengths between 3 and 7.5 s are used in this evaluation, the segments overlap by 50%. Table II shows the median number of resulting segments per setting as well as the mean duration of each walk.

	Setting		
	1	2	3
walk [s]	27.1	19.9	26.3
3 s	16	12	15
5 s	9	6	8
7.5 s	6	4	5

TABLE II
MEAN DURATION OF A WALK IN SECONDS FOR EACH SETTING AND THE MEDIAN NUMBER OF SEGMENTS AFTER SEGMENTATION TO 3, 5, AND 7.5 S.

IV. FEATURE EXTRACTION

For each of these segments, several features are extracted for x-, y- and z-direction as well as the magnitude vector $s_m = \sqrt{\bar{s}_x^2 + \bar{s}_y^2 + \bar{s}_z^2}$. These features are mean, minimum, maximum, standard deviation of the segment and

Diff Difference of maximal and minimal value of the segment

Bin Relative histogram distribution in linear spaced bins between the minimum and the maximum acceleration in the segment. Five and ten bins were used.

RMS Square root of the mean of the squares of the acceleration values of the segment:
$\text{rms} = \sqrt{\frac{\bar{s}_a^2(1)+\bar{s}_a^2(2)+\ldots+\bar{s}_a^2(n)}{n}}$, where n is the number of data points in the segment, $a \in \{x, y, z, m\}$.

Cross Number of sign changes in the segment

MFCC Mel-frequency cepstral coefficients [window size = 1.44s, window hop time = 0.048s, max freq.= 10Hz]

BFCC1 Bark-frequency cepstral coefficients [window size = 1.12s, window hop time = 0.032s, max freq.= 7.5Hz]

BFCC2 Bark-frequency cepstral coefficients [window size = 1.92s, window hop time = 0.048s, max freq.= 8.75Hz]

More information about cepstral coefficients can be found in [12]. During the evaluation, single features will be combined to feature vectors. The selection of features which compose the feature vectors is done in two different ways. Mainly, the performance of the single features is evaluated and best performing features are combined. Alternatively, features are selected based on their discriminative potential score (DPS). This score was introduced by Mrácek et al. [13]. In their work anatomical features of 3D face images are computed and the most suitable features are determined based on their discriminative potential score. The idea is to identify features which have a low intra-class variability and a high inter-class variability. For features consisting of multiple entries, like MFCC, each entry is considered as a separate feature. The mean, standard deviation and maximum deviation of the value ranges of each feature are used to identify low intra-class variability. The correlation of the feature values to the uniform probability density function is used to identify a high inter-class variability. Using these values, the DPS score is calculated. Based on this score, different feature vectors can be created by setting a threshold for the minimal required DPS.

Using all previously described features results in a feature vector of length 244. By setting the thresholds to values between -0.4 to 0.7, only a limited number of features remain, resulting in feature vectors of length between 241 and 12.

V. Evaluation

We apply the k-NN algorithm for classification, which is a simple instance based learning algorithm [14]. For these algorithms, the stored model consists of a set of instances together with their corresponding class labels and the used distance function. The Euclidean distance is applied, which is most common. During classification, the distances between the probe vector and the stored instances are computed. The classes of the k closest vectors are analyzed. The class of the majority of the k neighbours is assigned to the probe vector. In case k is even, the genuine class is assigned. The implementation of the WEKA library is used [15]. Each subject is once used as the genuine subject. The impostor training set consists always of data of 20 subjects, the impostor testing set consists of data from the remaining 15 subjects. The used setting, session and number of walks per subject are described below in the specific test description.

A. Single Features

Table III shows the half total error rate (HTER) for all settings. The HTER is half of the sum of False Match Rate (FMR) and False Non Match Rate (FNMR). These error rates state the proportion of segments which are falsely classified as genuine or impostor, respectively. MFCC and BFCC were computed for different axes-combinations and obtained using different values for the number of considered neighbours (k). The chosen sampling rate is 100 and the segments are of length 7500 ms. There is hardly any influence of the number of considered neighbours, therefore it is set to 3. The combination of all three axes and the magnitude vector gives best results for all three features when normal walk data (from setting 1 or 3) is used. Best feature set is BFCC2_xyzm obtaining an HTER of 13.09 for setting 1. In general, the results of setting 3 are slightly better because subjects got used to the data collection and walked more stable. When fast walk data is used (setting 2, using all 16 walks for training) the combination of x-axis and magnitude vector for BFCC1 gives best results.

The influence of the sampling rate and segment length is shown in Table IV for setting 1. The influence of both parameters is low. A sampling rate of 100 Hz and a segment length of 7500 ms is used for the following evaluation.

B. Combined Features

The statistical features have not been considered as single features but they are included in the DPS feature sets. Table V gives the results for the feature vectors using different thresholds for the minimal required DPS score (same walks are used for training and testing as before). The best results using the DPS features are a FMR of 4.24% at a FNMR of 23.44%, which is obtained by using 0.1 as a threshold to create the feature set. The corresponding feature vector DPS0.1 has 186 entries, hence only 58 features are not considered. The deleted features are mainly statistical ones, like *Diff*, but also entries at the end of the cepstral coefficient vectors are in some cases removed.

	Setting 1				Setting 2	Setting 3
k	1	2	3	4	3	3
BFCC1_xyzm	17.48	16.83	17.56	17.17	15.13	18.74
BFCC2_xyzm	13.46	12.62	13.09	12.61	14.38	12.25
MFCC_xyzm	15.14	14.27	14.98	14.23	14.88	12.34
BFCC1_xm	18.36	16.70	18.39	17.29	11.83	14.84
BFCC2_xm	14.63	14.09	14.85	14.47	13.26	13.24
MFCC_xm	16.79	15.61	16.53	16.02	13.25	15.51
BFCC1_x	23.13	20.76	22.74	21.54	21.11	24.73
BFCC2_x	22.35	20.48	22.15	21.06	22.11	18.80
MFCC_x	24.58	21.98	24.59	23.00	23.26	20.31
BFCC1_m	26.11	25.07	26.68	25.82	20.53	30.67
BFCC2_m	23.36	21.56	22.75	21.44	20.27	25.62
MFCC_m	23.19	21.32	23.49	22.23	19.31	28.21

TABLE III
HTER for BFCCs and MFCC using different values for the number of considered neighbours (k).

The result of DSP0.1 is similar to the so far obtained best result of a FMR of 3.97% at a FNMR of 22.22%. Because feature vector DSP0.1 is much longer and its creation requires the calculation of several different features, using only BFCC2_xyzm is preferred.

	Threshold	Feature Length	FMR	FNMR	HTER
DPS-0.4	-0.4	241	4.54	25.42	14.98
DPS-0.3	-0.3	240	4.55	25.38	14.97
DPS-0.2	-0.2	238	4.58	25.30	14.94
DPS-0.1	-0.1	230	4.55	24.66	14.60
DPS0	0	217	4.43	23.51	13.97
DPS0.1	0.1	186	4.24	23.44	13.84
DPS0.2	0.2	160	4.18	23.67	13.93
DPS0.3	0.3	119	4.18	27.97	16.08
DPS0.4	0.4	77	4.49	33.57	19.03
DPS0.5	0.5	44	4.34	40.09	22.22
DPS0.6	0.6	24	4.28	49.81	27.05
DPS0.7	0.7	12	5.48	61.51	33.50

TABLE V
Error rates obtained by using feature sets consisting of all features with a DPS above the given threshold (setting 1).

Table VI shows the results obtained for different combinations of BFCC1, BFCC2, and MFCC, using the maximal amount of training data available for each setting. Only the best axes-combinations are presented for each setting. This has been the combination of all three axes and the magnitude vector for setting 1, and the combination of x-axis and magnitude vector for setting 2. The combinations yield slightly worse results than the single features BFCC2_xyzm and BFCC1_xm alone for setting 1 and 2, respectively.

C. Varying the Amount of Training Data

Figure 1 states the results for varying amounts of training data. For setting 1 the results are given for feature set BFCC2_xyzm, for setting 2 the progress for several good feature vectors is depicted. Increasing the number of walks for training has the same impact to all given feature vectors. Adding more walks significantly decreases the HTER up to

	3000			5000			7500		
	FMR	FNMR	HTER	FMR	FNMR	HTER	FMR	FNMR	HTER
50	3.78	26.75	15.26	3.67	24.72	14.20	3.96	23.59	13.77
100	3.95	27.15	15.55	3.78	24.46	14.12	3.97	22.22	13.09
200	4.15	27.02	15.59	3.88	24.47	14.17	4.11	22.26	13.18

TABLE IV
ERROR RATES FOR FEATURE SET BFCC2_XYZM FOR DIFFERENT SAMPLING RATES AND SEGMENT LENGTHS (SETTING 1).

	Setting 1				Setting 2			
Features	Axes	FMR	FNMR	HTER	Axes	FMR	FNMR	HTER
BFCC1MFCC	x,y,z,m	4.09	26.52	15.31	x,m	4.10	22.49	13.29
BFCC2MFCC	x,y,z,m	3.90	24.28	14.09	x,m	4.26	22.53	13.39
BFCC1BFCC2	x,y,z,m	4.24	24.77	14.51	x,m	4.44	21.83	13.14
BFCC1BFCC2MFCC	x,y,z,m	4.01	23.63	13.82	x,m	4.32	21.83	13.08

TABLE VI
ERROR RATES FOR DIFFERENT COMBINATIONS OF BFCCS AND MFCCS FOR SETTING 1 AND 2.

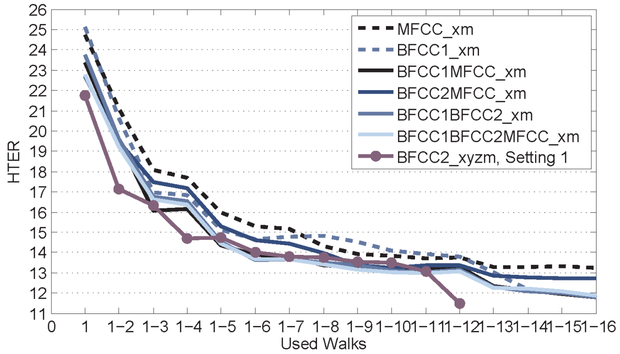

Fig. 1. Influence of the amount of training data for different feature vectors for setting 1 and 2.

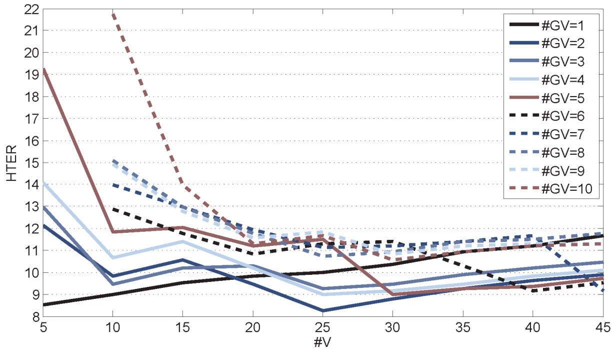

Fig. 2. Voting results for feature set BFCC2_xyzm (setting1).

using a set of six walks. For setting 2 and feature vector BFCC1_xm the initial HTER of 25.13 is lowered to 14.64. Using the same amount of training data for setting 1 decreases the HTER from 21.73 to 14.01. Afterwards, the decrease is slower. For each feature vector, the lowest error rate is obtained when using the maximal available amount of 12 or 16 normal or fast walks for training, respectively.

D. Voting Results

In all cases, the FNMR is still very high, while the FMR is low. To get a more balanced situation a voting algorithm is applied. So far, one classification result is obtained per segment; hence it is based on a very short walking duration of less than eight seconds. To get more reliable results, one can group several consecutive classification results and convert them to a single result. The idea is inspired by a petition quorum. Each of the single classification results votes either for genuine or for impostor. A defined number of $\#V$ votes is analysed and the number of votes for genuine $\#V_g$ is computed. If this number is larger or equal to the pre-selected threshold $\#GV$ the whole quorum votes for genuine, otherwise for impostor. If a low threshold is chosen, a higher weight is given to the votes for genuine. This is decreasing the FNMR while increasing the FMR. The evaluation showed that the decrease of FNMR is stronger than the increase of FMR, resulting in a lower HTER.

Voting results for setting 1 are given in Figure 2. $\#V$ is chosen to be between 5 and the maximal number of 45 available votes. At least 20 votes should be grouped (around 80 seconds of data) to achieve low error rates. The best HTER of 8.24 is obtained for $\#V = 25$ (around 100 seconds of data) and $\#GV = 2$.

VI. BENCHMARK TO OTHER CLASSIFIERS

In a previous publication [11] we evaluated the biometric performance of SVMs and HMMs on the same database as described in section II. Table VII gives an overview over the most important facts of the evaluated algorithms. The optimal feature vector and parameter settings for each algorithm have been determined. The length of the best feature vector lies between 52 and 104 values. The authentication should be based on 1.7 to 3.2 minutes of walk data which are used for voting. The lowest EER/HTER is between 8.24 and 8.85. A walking duration of five minutes for enrolment was identified to be best for all algorithms. Based on the error rates, no clearly best stochastic classifier can be identified. A disadvantage of SVMs was that outliers occurred training, which resulted in very high FNMRs. If this is happening during enrolment, the created classifier is unsuitable and the enrolment has to be repeated, resulting in a lower usability. k-NN gives the lowest HTER after voting, but has also the lowest error rate before voting. Hence, it gives already good results when classification is based on a short walk duration. Therefore, k-NN is identified to be most suitable.

Algorithm	Length of best feature vector	Authentication based on ... minutes	Lowest EER/HTER
HMM	104	2.5	8.75
SVM	52	2.5	8.85
k-NN	52	1.7	8.24

TABLE VII
COMPARISON OF THE DIFFERENT EVALUATED ALGORITHMS.

VII. IMPLEMENTATION ON A SMARTPHONE

The k-NN algorithm has been implemented on a smartphone using the previously identified best settings and feature set BFCC2_xyzm. During enrolment the subject is requested to walk for five minutes. Afterwards the classifier is constructed using the enrolment data of the genuine subject and data of 20 subjects from the database described in section II (setting 1, session 1) as impostor data. Using pre-computed feature vectors from the impostor data, training takes around 1.5 minutes on a Motorola Defy smartphone running Android 2.2. This is a short enough time, because enrolment is only rarely performed. The effort for the user during enrolment is restricted to five minutes of walking. Authentication is currently based on 30 seconds walk data. The data is segmented and features are extracted as described previously. If at least one of the segments is classified as belonging to the genuine user, the authentication is successful. The whole process including segmentation, feature extraction and classification takes around seven seconds. This is a short enough duration to implement a continuous classification. By iteratively collecting data and performing the classification a current classification results is available when it is required. Therefore, an authentication consists only of retrieving the last authentication result from the system, which can be done without temporal delay.

VIII. CONCLUSION

We propose to use the k-NN algorithm for classification of gait feature vectors. The recognition performance is better than the previously presented results of SVMs and HMMs. Several features have been extracted from the gait data and were combined to feature vectors. We showed that a selection of relevant features based on the DPS gives reasonable results. But considering the amount of necessary features, a feature vector consisting only of feature BFCC, computed for all three acceleration axes and the magnitude vector, is preferred for normal walk. We implemented all steps necessary for enrolment and authentication on a standard smartphone. Low processing times are achieved, showing that the proposed approach is applicable in practice.

Future work will include the evaluation of the proposed method using a more demanding data set. So far, it has been shown that the algorithm performs well in a controlled environment when subjects were walking on flat floor. The influence of covariates like clothes, backpacks, surfaces needs to be analyzed. In addition, it is desirable to allow for an unconstrained position of the phone. It is shown that different feature vectors should be used for the different velocities. Therefore, the implemented authentication method on the smartphone has to be augmented by activity recognition to identify when and in which velocity the subject is walking.

ACKNOWLEDGMENT

This work was supported by CASED (www.cased.de). The authors would like to thank the numerous volunteers participating in the data collection.

REFERENCES

[1] F. Breitinger and C. Nickel, “User Survey on Phone Security,” in *BIOSIG 2010 - Proceedings of the Special Interest Group on Biometrics and Electronic Signatures*, 2010.

[2] International Telecommunication Union, “Measuring the Information Society,” Tech. Rep., 2011.

[3] ORACLE, “Opportunity Calling: The Future of Mobile Communications – Take Two,” 2011.

[4] H. J. Ailisto, M. Lindholm, J. Mäntyjärvi, E. Vildjiounaite, and S.-M. Mäkelä, “Identifying people from gait pattern with accelerometers,” *Biometric Technology for Human Identification II*, vol. 5779, no. 1, pp. 7–14, 2005, vTT Electronics, Finland. [Online]. Available: http://link.aip.org/link/?PSI/5779/7/1

[5] D. Gafurov, “Performance and Security Analysis of Gait-based User Authentication,” Ph.D. dissertation, University of Oslo, 2008. [Online]. Available: http://urn.nb.no/URN:NBN:no-19706

[6] M. O. Derawi, C. Nickel, P. Bours, and C. Busch, “Unobtrusive User-Authentication on Mobile Phones using Biometric Gait Recognition,” in *6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2010.

[7] J. Frank, S. Mannor, and D. Precup, “Activity and Gait Recognition with Time-Delay Embeddings,” in *AAAI Conference on Artificial Intelligence*, 2010. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1666

[8] J. Kwapisz, G. Weiss, and S. Moore, “Cell phone-based biometric identification,” in *4th IEEE International Conference on Biometrics: Theory Applications and Systems*, 2010, pp. 1 –7.

[9] C. Nickel, M. O. Derawi, P. Bours, and C. Busch, “Scenario Test of Accelerometer-Based Biometric Gait Recognition,” in *3rd International Workshop on Security and Communication Networks*, 2011.

[10] C. Nickel and C. Busch, “Classifying accelerometer data via hidden markov models to authenticate people by the way they walk,” in *IEEE International Carnahan Conference on Security Technology*, October 2011, pp. 1–6.

[11] C. Nickel, H. Brandt, and C. Busch, “Benchmarking the Performance of SVMs and HMMs for Accelerometer-Based Biometric Gait Recognition,” in *Proceedings of the 11th IEEE International Symposium on Signal Processing and Information Technology*, 2011.

[12] B. Logan, “Mel Frequency Cepstral Coefficients for Music Modeling,” in *International Symposium on Music Information Retrieval*, 2000.

[13] S. Mracek, C. Busch, R. Dvorak, and M. Drahansky, “Inspired by Bertillon – Recognition Based on Anatomical Features from 3D Face Scans,” in *Proceedings IEEE International Workshop on Security and Communication Networks*, 2011.

[14] D. W. Aha, D. Kibler, and M. K. Albert, “Instance-Based Learning Algorithms,” in *Machine Learning*, vol. 6, 1991, pp. 37–66.

[15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA Data Mining Software: An Update,” in *ACM Special Interest Group on Knowledge Discovery and Data Mining Explorations*, vol. 11, 2009.