Spoken Digit Recognition using the k-Nearest-Neighbor method and Mel Frequency Cepstral Coefficients

Sorin MURARU, Cătălina-Lucia COCIANU
Bucharest University of Economic Studies, Bucharest, Romania
sor.muraru@gmail.com, catalina.cocianu@ie.ase.ro

This study investigates the utilization of the k-nearest-neighbor algorithm within the framework of machine learning for speech recognition applications. The AudioMNIST dataset is used for performing the evaluations in which the model predicts the spoken digit, namely from 0 to 9. Two different training-to-test percentage splits of the dataset are used, 70%-30% and 80%-20%, while the k parameter ranges from 1 to 12. To better adapt the prediction model, the Mel-frequency cepstrum coefficients are extracted from each audio sample, and the 13 filters are averaged over 25 ms frame windows with 10 ms frame overlap. In both training-to-test configurations the value for the k parameter that obtained the highest accuracy (> 95%) is k=5, while the easiest to predict digits was "7". These findings underscore the efficacy of k-nearest-neighbor in speech recognition tasks and highlight the importance of parameter selection and feature extraction techniques in optimizing model performance. Further exploration of kNN's applicability in diverse speech recognition contexts holds promise for advancing the field's understanding and practical implementations.

Keywords: *k-Nearest-Neighbor, Machine learning, MFCC, Speech recognition, Natural language processing*
DOI: 10.24818/issn14531305/28.2.2024.01

1 Introduction

Automatized recognition of vocal commands or voice authentication using specific key phrases is becoming increasingly widespread in today's society. Be it for bank services, operating instructions, or simply virtual assistant commands, among many others, it is of particular importance for the speech recognition process to be highly accurate and precise, with little room for error [1]. Machine learning and deep learning techniques play a vital role in the development of the field [2], with consistent research existing in areas such as security and identity verification [3], sentiment analysis [4] or text summarization [5]. Methods such as convolutional neural networks [6] or long short-term memory [7] have been proven adequate when trying to perform such tasks. Concerning vocal input, the characteristics of this type of data present specific challenges that reside primarily in the human way of hearing. Many promising methodologies have something in common: employing the Mel-frequency cepstral coefficients (MFCC) for the description of the analyzed speech excerpt [8].

Extracting the MFCCs specific to an audio sample involves several distinct stages. Explicitly, the MFCCs rely on splitting the audio piece into a handful of frequency windows that are meant to mimic the human way of hearing, essentially creating a logarithmic scale of frequency bands [9]. Although many classifying methods are compatible with the use of MFCCs, this paper employs the use of a straightforward k-Nearest-Neighbours (kNN) technique to predict the digit spoken in the audio excerpt from the AudioMNIST database [10]. This is a versatile method based on the hypothesis that data points with similar characteristics tend to be close to one another in a multidimensional space. As such, the shorter the distance between an analyzed data point and another already existing one, the stronger the chance of them being similar in characteristics. On this basis, extracting, processing, and classifying MFCC feature vectors using the kNN method proved very effective for the task of predicting the spoken digit from the AudioMNIST database.

Lately, machine learning (ML) and deep learning (DL) methods have been employed in

the wider field of natural language processing (NLP) with varying degrees of efficacy. Researchers have explored diverse methodologies within ML, ranging from traditional statistical approaches to cutting-edge deep learning techniques, to enhance NLP applications. Key themes include the development of sophisticated neural network architectures such as transformers, attention mechanisms, and recurrent neural networks, which have demonstrated remarkable success in tasks like language modeling, sentiment analysis, and machine translation. Additionally, there is a growing emphasis on domain-specific adaptations of ML models for diverse applications, including healthcare, finance, and cybersecurity. Challenges such as ethical considerations, interpretability, and the need for more robust and explainable models are actively discussed, highlighting the evolving landscape and the continuous pursuit of refining ML and NLP techniques for broader and more impactful applications. Notably, Devlin et al. [11] have shown that significantly accurate results can be achieved by pre-training models on large-scale machines for various NLP tasks. This implies an unsupervised training initial stage for learning general language representations and patterns, prior to the actual supervised training main stage. In addition to advancements in NLP, ML and DL techniques have significantly impacted vocal recognition specifically, an area explored in various academic works. Lample and Conneau [12] show that it is possible to attain multilanguage recognition by performing cross-lingual pretraining beforehand. Malik et al. [13] describe automatic speech recognition through various ML techniques, offering insights into the evolution and future directions of speech recognition systems. Soares et al. [14] use a support-vector-machine and random-forest classifier to determine the presence of queen bees in hives based on audio samples out of which MFCCs are extracted. Siyad and George [15] compare the effectiveness of random forests and that of kNN for identifying spoken Indian language using MFCCs and vowel onset points. Utilizing MFCCs feature extraction, Jayadi et al. [16] find the kNN technique to be a reliable classifier (80% accuracy) for identifying flu based on cough sounds. Arpitha, Madhumathi, and Balaji [17] make use of kNN and support vector machines for analyzing the spectrogram of an ECG signal converted into acoustic series using MFCCs and calculating the obtained mean values. Furthermore, Lahmiri et al. [18] analyzed the cry of newborn babies through various methods, including MFCC feature extraction, and used the obtained coefficients to classify health-related issues through the kNN technique and support vector machines. Speaker identification was performed by Yerramreddy et al. [19] using the spectral features extracted from MFCCs and comparing the efficacy of multiple classifier models, including kNN. The use of both kNN and the spectral features extracted by calculating the MFCCs shows potential for good results in a multitude of audio sample classifying tasks.

This paper aims to explore the effectiveness of applying the kNN method with different levels of k to identify the spoken digits in the AudioMNIST database by making use of MFCCs. The values of k range from 1 to 13 and the code is written in Python 3.10, making use of well-known libraries in the field. Firstly, the dataset and the working method are described in detail. Subsequently, the accuracy of each scenario is calculated, and the results are displayed through a comparative overview to identify the best predictive configuration. Moreover, the outcome and its essential characteristics are discussed in an ensuing section. Lastly, conclusions are formed and provided succinctly.

2 Materials and Methods

The proposed methodology was applied to a 30,000 audio recordings dataset. The AudioMNIST contains samples of spoken digits from 0 to 9 of 60 different speakers [20]. Each .wav file was processed and depending on its inclusion in the training or prediction subset, was labeled or had its label predicted, respectively. Other than the spoken digit no metadata or otherwise easily identifiable aspect of the recordings was used, such as accent of the speaker, gender of the speaker, age of

the speaker, or recording room. All audio samples were pooled together and used as originally found in the dataset, without any cropping or enhancements applied to the audio signals.

Each of the 30,000 audio recordings was processed as follows, in the stages preliminary to applying the kNN algorithm:

1. Each .wav file was first read into the system using the SciPy library and the respective sample rate and amplitude were stored per recording.
2. The MFCCs were extracted for each audio file using the found parameters in the previous step and setting the FFT number to 1200.
3. All the obtained coefficients were then saved in plain .txt files.

Similar to Yusuf and Hidayat [21], the extracted MFCCs were the standard 13 filters. The process can be shortly explained as follows:

1. The audio signal undergoes pre-emphasis to enhance high-frequency components, followed by segmentation into short frames with overlapping intervals.
2. Each frame is multiplied by a windowing function to reduce spectral leakage and frame boundary artifacts.
3. Fast Fourier Transform (FFT) is applied to obtain the power spectrum of each frame.
4. The power spectrum is passed through a Mel filterbank, which simulates human auditory perception by grouping spectral energy into overlapping triangular filters on the Mel scale.
5. The logarithm of the filterbank energies is computed to approximate the non-linear response of the human ear.
6. Discrete Cosine Transform (DCT) is applied to the log filterbank energies to separate the coefficients and reduce dimensionality.
7. The subset of the first 13 DCT coefficients is retained as MFCC features, capturing the most relevant spectral information.

The obtained processed data is now composed of 30,000 data points in a multi-dimensional space represented by the mean values of each of the 13 feature vectors. It is now possible to use the kNN algorithm to perform the classification task. The kNN method is one of the most commonly used ML techniques developed to perform both classification and regression tasks. Its popularity is mainly due to the ease of interpretation and the low complexity. The idea underlying the kNN algorithm is that similar data points usually produce similar outcomes (labels or continuous values), that is similarity means closeness [22]. The closeness degree can be expressed in many ways, using distance metrics. In the case of continuous variables, the algorithm usually computes Euclidian distances to evaluate and classify data points. Alternatively, one can use many other distance functions, such as Minkowski distance, Manhattan distance, Chebyshev measure, cosine transform, and so on. For categorical variables, the most commonly used choice is the Hamming distance [23]. The kNN algorithm that solves the classification problem is briefly described as follows. We denote by $\aleph = \{x_1, x_2, \dots, x_n\}$ the training dataset and let x_{new} be the input data. Each sample belonging to $\aleph$ is assigned to a known class label and x_{new} is going to be labeled by the following procedure.

Step 1. Initialize *k*, the number of elements (neighbors) considered for classification

Step 2. For each $x_i \in \aleph$ compute $D_i = Distance(x_i, x_{new})$

Step 3. Sort $\aleph$ ascending based on $\{D_1, D_2, \dots, D_n\}$ and get $\aleph_{new}^{sort}$

Step 4. Select the first k elements of $\aleph_{new}^{sort}$, $\aleph_{new}^{k}$, and get the most frequent class assigned to the data points in $\aleph_{new}^{k}$

Note that the accuracy of kNN essentially depends on *k*.

For this work, the distance between any two points i and j with coordinates $vi_{1\text{-}13}$ and $vj_{1\text{-}13}$, can be computed using the Euclidean distance:

$$Distance_{ij} = \sum_{no=1}^{13} (vi_{no} - vj_{no})^2$$

The lower the obtained distance, the higher the potential similarity between the two data points.

The dataset was split into different proportions for training and testing. The first configuration consisted of the proportions: 70% of the data used for training and 30% of the data used for testing, resulting in 21.000 and 9.000 samples, respectively. The second configuration had a larger base used for training, namely 24.000 samples or 80% of the total, and the rest were used for testing. To preserve the generality character of the study, each of the multiple simulations per k-level ran with randomized selections of the audio samples to be included in the training and testing sets. This was the applied procedure for all runs, irrespective of their number or k level, leading to complete independence between any two runs and any two k-levels.

3 Results

For each k-level configuration, the simulations consisted of 50 runs, at the end of which the accuracy was calculated using the formula:

$$Accuracy\ (\%) = \frac{Number\ of\ predictions\ correctly\ labelled}{Number\ of\ data\ points\ in\ test\ set}\ (\%)$$

The obtained statistics for the 70%-30% configuration according to the ANOVA test can be seen in **Figure 1**.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	0.00667	12	0.00056	143.62	3.95658e-172
Error	0.00247	637	0		
Total	0.00914	649			

Fig. 1. The ANOVA results table of the simulations at 70%-30% training-to-test split.

The function above computes the p-value for a balanced one-way analysis of variance (ANOVA) and provides relevant statistical outputs. It assesses the null hypothesis that the samples in y are drawn from populations with identical means, contrasting with the alternative hypothesis suggesting differences in population means. This evaluation is fundamental in discerning potential variations among the groups under study. The ANOVA table, generated alongside the p-value, offers insights into the partitioning of variance, delineating between-group variation (Columns) and within-group variation (Error). Notable components within the ANOVA table include the sum of squares (SS) and the degrees of freedom (df). The total degrees of freedom is derived from the total number of observations minus one. Furthermore, the degrees of freedom for between-groups and within-groups are computed as the number of groups minus one and the difference between total and between-groups degrees of freedom, respectively. These metrics aid in dissecting the sources of variance within the dataset, facilitating informed interpretations of group differences. The mean squared error (MS) represents the variance within each source of variation and is calculated as the SS divided by df. The F-statistic, derived from the ratio of mean squared errors, serves as a measure to assess the significance of differences among group means. The p-value associated with the F-statistic indicates the probability of observing a test statistic greater than or equal to the computed value under the null hypothesis. A low

p-value suggests statistical significance, implying substantial disparities among column means. Furthermore, the obtained accuracies for each level of the parameter k for the 70%-30% configuration can be viewed in **Figure 2**.

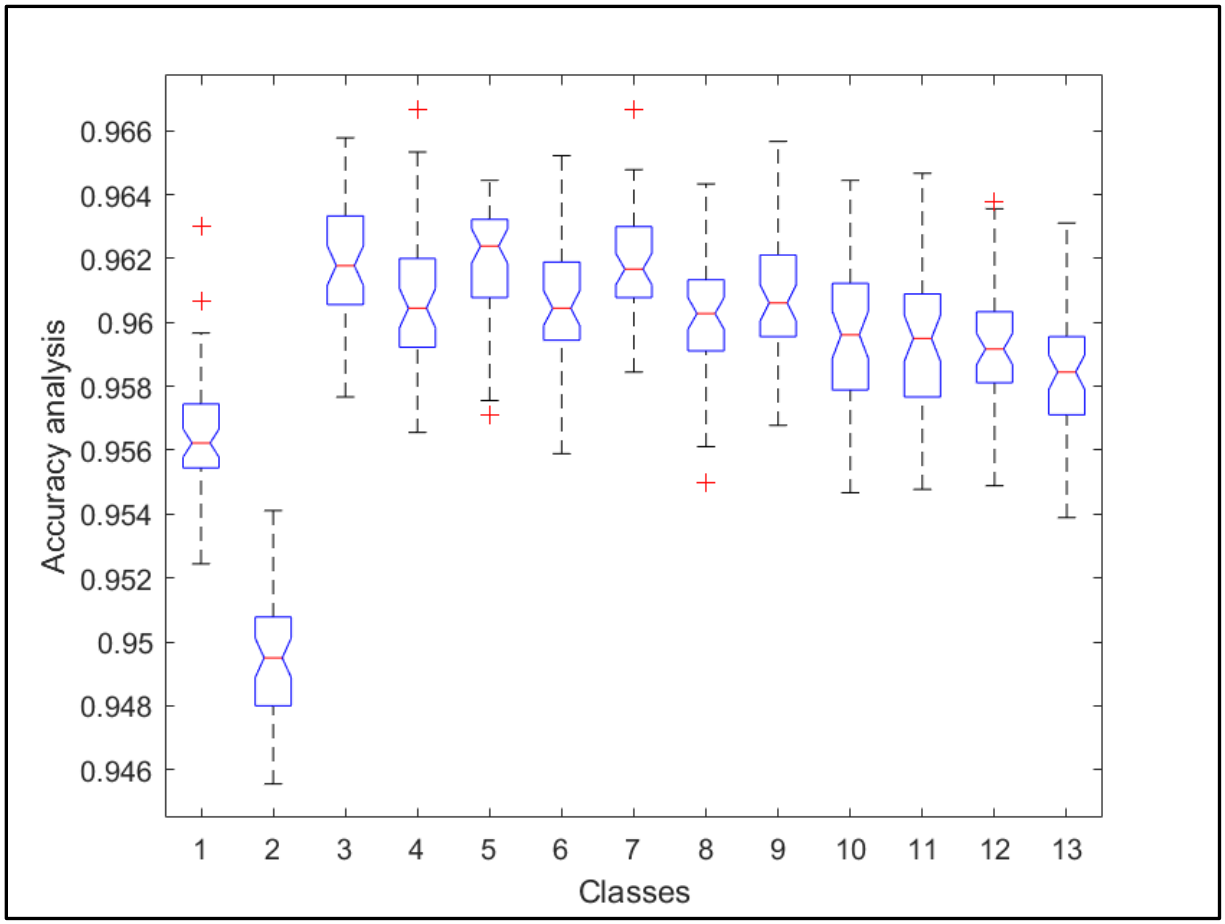


Fig. 2. Box-plot graph displaying the obtained accuracy for each k parameter at 70%-30% training-to-test split.

Overall, a consistent tendency emerges as a pattern, placing the k=3 level as slightly more accurate than the rest, however, overtaken by the k=5 when solely considering the mean accuracy values. The length of the box plots has only slight differences, showcasing that the distributions of the predictions are somewhat similar. In addition, the occasional outliers do not go very far away from the extremes of the box plots, and their very few numbers show consistency in the model. The exact mean values of accuracy can be seen in **Table 1**.

Table 1. The mean values of accuracy for each k-level at 70%-30% training-to-test split.

1	**2**	**3**	**4**	**5**
95.645%	94.940%	96.194%	96.069%	96.200%
6	**7**	**8**	**9**	**10**
96.061%	96.167%	96.022%	96.078%	95.962%
11	**12**	**13**	-	-
95.937%	95.923%	95.843%	-	-

Although the mean of k=5 is higher than k=3, it is only slightly so by 0.006%. Moreover, given that the latter had achieved few cases with higher accuracy, it is worth pinpointing both as

favorable parameters. The obtained statistics for the 80%-20% configuration according to the ANOVA test can be seen in **Figure 3**.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	0.00657	12	0.00055	100.87	1.79989e-138
Error	0.00345	637	0.00001		
Total	0.01002	649			

Fig. 3. The ANOVA results table of the simulations at 80%-20% training-to-test split.

In terms of statistical relevance, the tables show that both formats for training and testing, either 70-30 or 80-20, gain significant relevance as the obtained means show a high degree of independence. The obtained accuracies for each level of the parameter k for the 80%-20% configuration can be viewed in **Figure 4**.

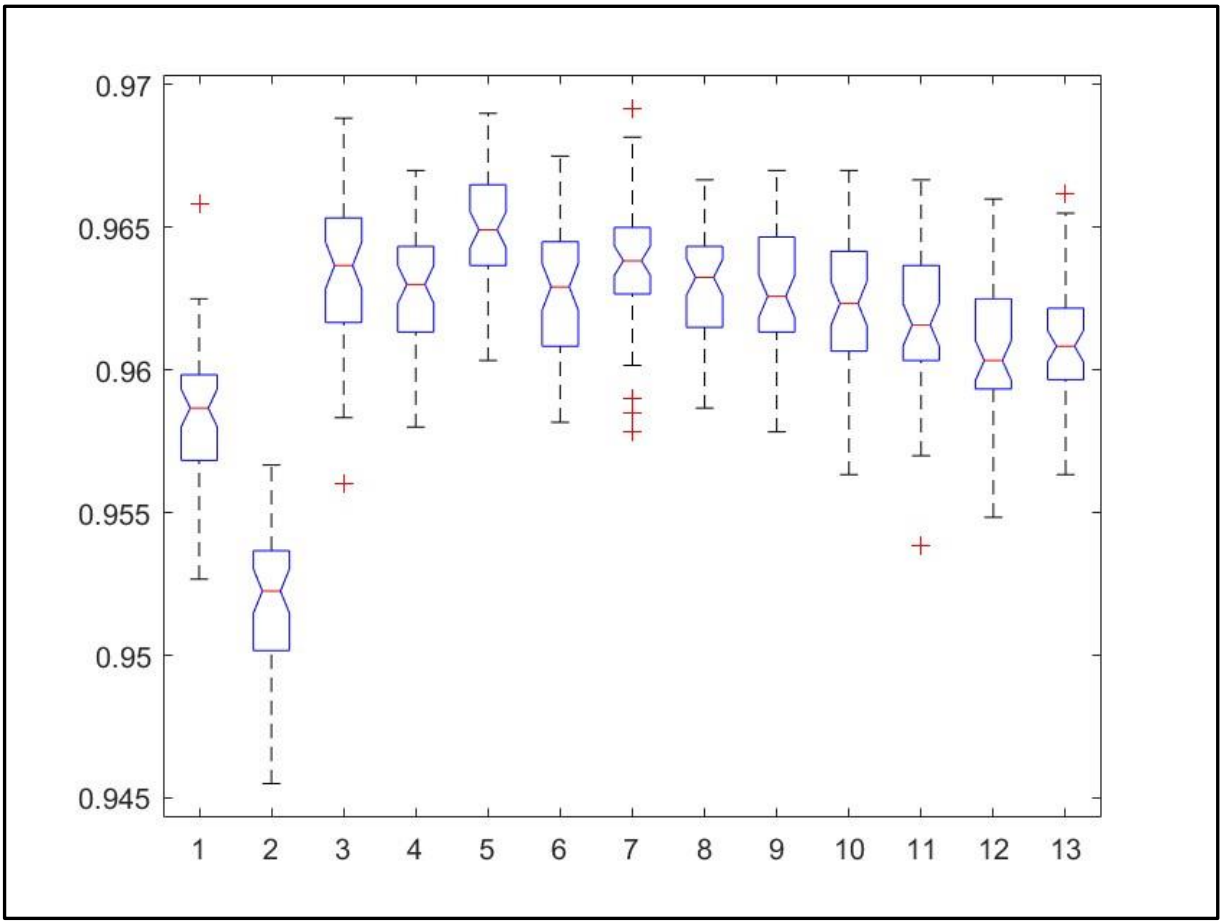


Fig. 4. Box-plot graph displaying the obtained accuracy for each k parameter at 80%-20% training-to-test split.

Similarly to the 70%-30%, it becomes obvious from the graph that the k=3 and k=5 parameters gained the most accuracy for their models. However, the latter of the two seems to have both a higher mean, and a slightly higher extreme for its top end. This would place k=5 case as more favorable the higher the percentage allocated to training data. The outliers are few in number, and the length of the boxplots kept a somewhat similar extent. The case k=7 would seem to also have high accuracy, with respect to the k=3 case, closely

followed by k=6 case. The k=2 case has once again achieved the lowest accuracy, followed by k=1. However, as expected, due to the more training data, the overall accuracies obtained in this run were slightly higher than in the 70%-30% one.

Table 2. The mean values of accuracy for each k-level at 80%-20% training-to-test split.

1	2	3	4	5
95.822%	95.131%	96.344%	96.260%	96.504%
6	**7**	**8**	**9**	**10**
96.334%	96.350%	96.328%	96.300%	96.255%
11	**12**	**13**		
96.117%	96.111%	96.120%		

In terms of means, the case k=5 has remained the one with the highest accuracy in the 80%-20% training to test percentage split. Although at a higher difference than before, the k=3 case is also one of the top models, having been overtaken slightly only by the k=7 case. All the obtained prediction accuracy means surpass the 95% mark, with k=2 case being the lowest at 95.131%.

4 Discussion

Both levels of training and testing show very high accuracies for k=3 and k=5 methods. Additionally, the same hierarchy pattern holds for both formats applied, where the case of k=2 attains the lowest accuracy. However, it must be noted that all the accuracies presented in the graphs revolve around the 95% mark, which implies very accurate predictions even for the k=2 case. It must be noted that using the kNN method for prediction may have certain subtleties that could affect the accuracies to have been this high: one is that a data label will be picked no matter the Euclidean distance calculated, irrespective of it being just too far away from the other data points. In contrast, it is important to note that the kNN technique is one of the most reliable methods to use for incorporating the multidimensionality of the datasets.

The cases k=3 and k=5 have been selected for further analysis given their presence in the top ranking in both training-test scenarios. As such, additional testing had been performed in order to see which predicted digit had most benefit in terms of accuracy for either of the training – testing variants, 70%-30% and 80%-20%, with the results following in **Figure 5**.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	0.26843	9	0.02983	809.95	1.06268e-287
Error	0.01804	490	0.00004		
Total	0.28647	499			

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	0.23504	9	0.02612	755.37	9.09657e-281
Error	0.01694	490	0.00003		
Total	0.25198	499			

Fig. 5. The ANOVA results table of the simulations at 70%-30% training-to-test split. (Left) k=3. (Right) k=5.

The results of the ANOVA tests show similar results for both the k=3 parameter model and k=5. The low p-values obtained confirm that the obtained results among the spoken digits display a high degree of independence, at a very elevated statistical significance level.

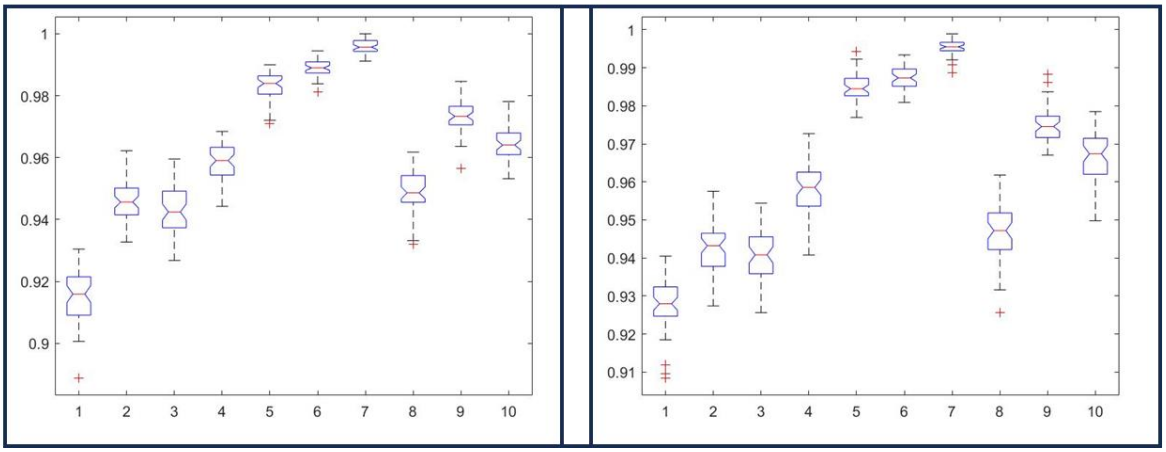


Fig. 6. Box-plot graph displaying the obtained accuracy for each k parameter at 70%-30% training-to-test split. (Left) k=3. (Right) k=5.

When viewing **Figure 6**, the boxplots in both cases show that digit "7" had the highest prediction success, with a very high accuracy ratio. Moreover, the mean being higher than 99% would imply that the particularities of pronouncing this specific digit make it much easier to recognize through the MFCC-parameterized kNN method used. Although all digits obtained an accuracy mean higher than 90% at the 70%-30% training level, the digits "6" and "5" had higher than 98% accuracy in both k=3 and k=5 cases, ranked in this order. The overall ranking according to the accuracy means is kept in both the left boxplot and the right one, which means that the changing of the k parameter does not affect the correlation between digits, however, the k=5 parameterized-model displays slightly better accuracies overall.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	0.23798	9	0.02644	507.28	6.79726e-242
Error	0.02554	490	0.00005		
Total	0.26352	499			

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	0.19932	9	0.02215	438.35	5.47926e-228
Error	0.02476	490	0.00005		
Total	0.22408	499			

Fig. 7. The ANOVA results table of the simulations at 80%-20% training-to-test split. (Left) k=3. (Right) k=5.

For the 80%-20% training-to-test percentage split ratio, the ANOVA tests confirm the validity of the independence of the obtained results in both cases of k=3 and k=5.

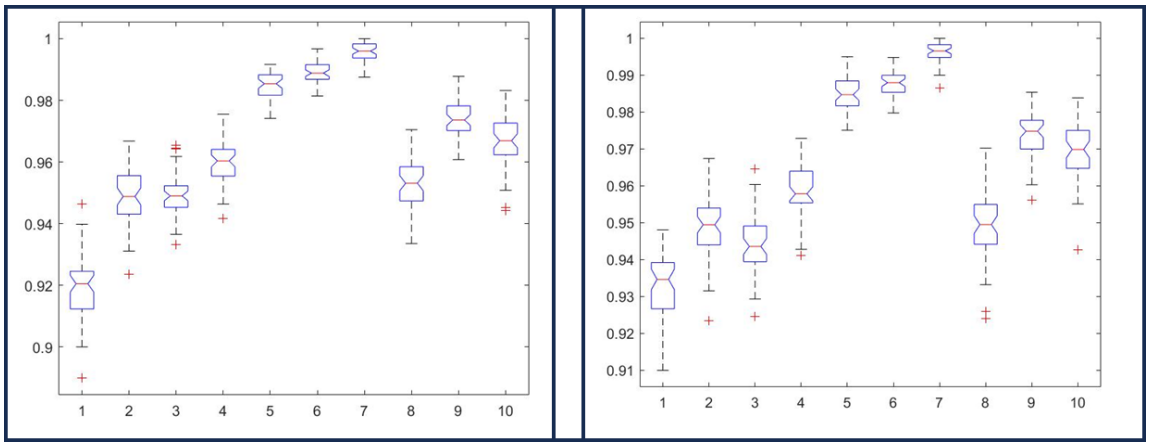
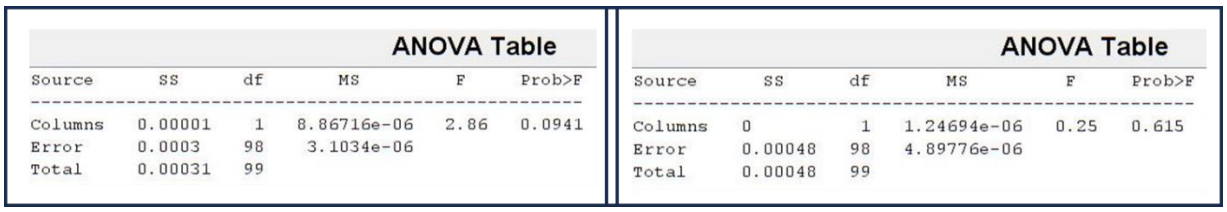


Fig. 8. Box-plot graph displaying the obtained accuracy for each k parameter at 80%-20%

training-to-test split. (Left) k=3. (Right) k=5.

The boxplots for the prediction model using 80%-20% data split maintain the same pattern in terms of ranking as those for the 70%-30% cases. Digits "7", "6" and "5" top the ranking, in this order, with similar accuracy means as before. Likewise, no digit falls below the 90% prediction success mean, while the k=5 parameter once again seems to obtain better results than the k=3 parameter, more evidently so than before. As expected, the overall obtained accuracies are only slightly better for the 80%-20% vs 70%-30% dataset splits. Given that the k=5 parameter seems to have the highest accuracy in all tests and is closely followed by k=3, a closer comparison between k=3 and k=5 is shown in **Figure 10** to give a final recommendation in between the two.



ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	0.00001	1	8.86716e-06	2.86	0.0941
Error	0.0003	98	3.1034e-06		
Total	0.00031	99			

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	0	1	1.24694e-06	0.25	0.615
Error	0.00048	98	4.89776e-06		
Total	0.00048	99			

Fig. 9. The ANOVA results table of the simulations. (Left) at 70%-30% training-to-test split (Right) at 80%-20% training-to-test split.

Given that only two means were analyzed, the ANOVA test results do reflect less independence or statistically significant differences between the two means of the k=3 and k=5 groups. In the 70%-30% training-to-test split scenario, the F-statistic shows a variance ratio of 2.86 and a p-value of 0.0941. This would imply statistically independent means at 90% statistical significance, which can be deemed satisfactory given the reduced analyzed dataset. However, at the 80%-20% training-to-test split scenario, the F-statistic shows a very small variance ratio of 0.25 with a very high p-value of 0.615. This comes as expected, as there is even less data available for testing in the 20% category.

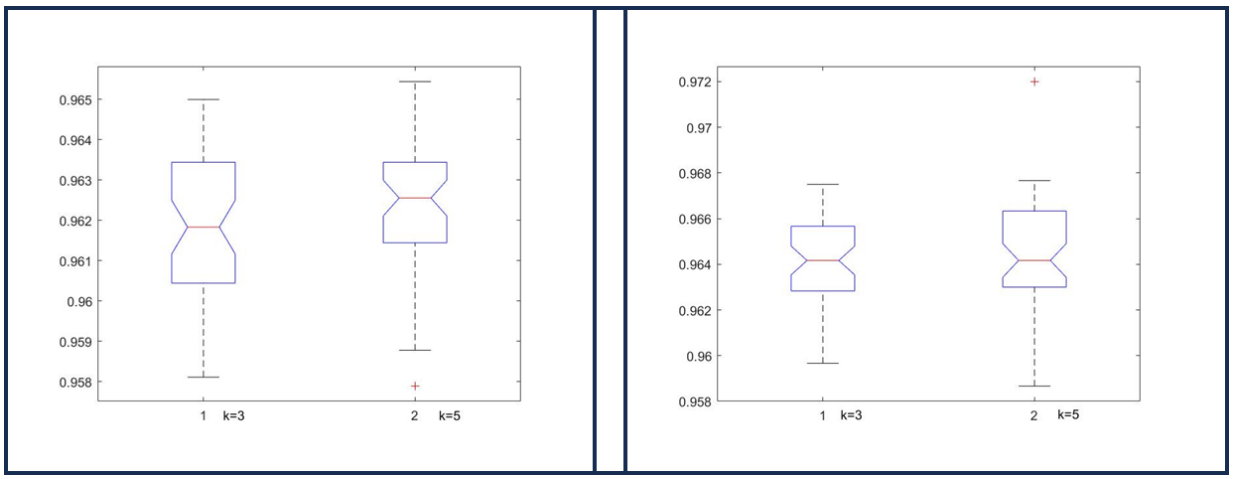


Fig. 10. Box-plot graph displaying the obtained accuracy for each k parameter. (Left) at 70%-30% training-to-test split. (Right) at 80%-20% training-to-test split.

Certain characteristics of the dataset may be of particular interest when analyzing the data through machine learning or deep learning techniques. The most obvious is that different speakers take different times to pronounce the same digit, and even more so, may themselves vary their speech time for different recordings of the same digit. Additionally, the starting points in time of consistent or adequate signal data may also vary, as do the ending point in

time, as the recording may not have a sudden cut.

5 Conclusions

In this study, the application of the k-nearest-neighbor technique was explored in the domain of speech recognition, utilizing the AudioMNIST dataset as a benchmark. Through extensive evaluations with varying training-test splits and k parameter values, valuable insights were gained into the effectiveness of kNN in recognizing spoken digits. The results revealed that kNN, when appropriately tuned with an optimal value of k=5, achieved competitive accuracy rates across different training-to-test split configurations. Interestingly, digit "7" emerged as the easiest to predict among the ten spoken digits, highlighting potential variations in recognition difficulty among classes. Moreover, the study underscored the significance of feature extraction techniques in enhancing model performance. By leveraging MFCCs, informative representations of audio samples were obtained, facilitating robust classification of spoken digits. The averaging of MFCCs over 25 ms frames with 10 ms overlap proved effective in capturing essential spectral features with the 13 filters, contributing to the discriminative power of the kNN model.

The study's findings contribute to the growing body of literature on ML-based speech recognition systems, providing valuable insights into the practical considerations and optimization strategies when employing kNN algorithms in this domain. Furthermore, this work demonstrates the importance of dataset selection, parameter tuning, and feature engineering in maximizing recognition accuracy and generalization capabilities. Moving forward, future research endeavors could explore the applicability of kNN in more complex speech recognition tasks, such as continuous speech recognition or speaker identification. Additionally, investigations into alternative distance metrics, neighborhood selection strategies, and ensemble methods could further enhance the performance and robustness of kNN-based speech recognition systems. Overall, the results of this study highlight the promising potential of kNN algorithms in speech recognition applications and underline the importance of methodical experimentation and refinement in obtaining better prediction accuracy.

References

[1] A. Ross, S. Banerjee and A. Chowdhury, "Security in smart cities: A brief review of digital forensic schemes for biometric data", *Pattern Recognition Letters*, vol. 138, pp. 346-354, 2020, DOI: 10.1016/j.patrec.2020.07.009.

[2] Z. Can and E. Atilgan, "A Review of Recent Machine Learning Approaches for Voice Authentication Systems", *Journal of Information and Communication Technologies*, vol. 5, no. 1, pp. 96-114, June 2023, DOI: 10.53694/bited.1296035.

[3] Z. Meng, M.U. Bin Altaf and B.-H. Juang, "Active voice authentication", *Digital Signal Processing*, vol. 101, 102672, 2020, DOI: 10.1016/j.dsp.2020.102672.

[4] K.L. Tan, C.P. Lee, K.S.M. Anbananthen and K.M. Lim, "RoBERTa-LSTM: A hybrid model for sentiment analysis with transformers and recurrent neural network", *IEEE Access*, vol. 10, pp. 21517-21525, 2022, DOI: 10.1109/ACCESS.2022.3152828.

[5] S. Yu, S.R. Indurthi, S. Back and H. Lee, "A Multi-Stage Memory Augmented Neural Network for Machine Reading Comprehension", *Proceedings of the Workshop on Machine Reading for Question Answering*, pp. 21-30, 2018, DOI: 10.18653/v1/W18-2603.

[6] M.M. Lopez and J. Kalita, "Deep learning applied to NLP", *arXiv*, 2017, DOI: 10.48550/arXiv.1703.03091.

[7] S. Ghosh, O. Vinyals, B. Strope, S. Roy, T. Dean and L. Heck, "Contextual LSTM (CLSTM) models for Large scale NLP tasks", *arXiv*, 2016, DOI: 0.48550/arXiv.1602.06291.

[8] M.H. Rafizah, I. Khalid and M. Shamsul, "A review on speaker recognition: Technology and challenges", *Computers and Electrical Engineering*, vol. 90, 107005, 2021, DOI:

10.1016/j.compeleceng.2021.107005.

[9] V. Tiwari, "MFCC and its applications in speaker recognition", *International Journal on Emerging Technologies*, vol. 1, no. 1, pp. 19-22, 2010.

[10] S. Becker, J. Vielhaben, M. Ackermann, K.-R. Muller, S. Lapuschkin and W. Samek, "AudioMNIST: Exploring Explainable Artificial Intelligence for audio analysis on a simple benchmark", *Journal of Franklin Institute*, vol. 361, no. 1, pp. 418-428, 2024, DOI: 10.1016/j.jfranklin.2023.11.038.

[11] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *arXiv*, 2019, DOI: 10.48550/arXiv.1810.04805.

[12] G. Lample and A. Conneau, "Cross-lingual Language Model Pretraining", *arXiv*, 2019, DOI: 10.48550/arXiv.1901.07291.

[13] M. Malik, M.K. Malik, K. Mehmood and I. Makhdoom, "Automatic speech recognition: a survey", *Multimedia Tools and* Applications, vol. 80, pp. 9411-9457, 2021, doi: 10.1007/s11042-020-10073-7.

[14] B.S. Soares, J.S. Luz, V.F. de Macêdo, R.R.V. e Silva, F.H.D. de Araújo, D.M.V. Magalhães, "MFCC-based descriptor for bee queen presence detection", *Expert Systems with Applications*, vol. 201, 117104, 2022, DOI: 10.1016/j.eswa.2022.117104.

[15] H. M. Siyad and A. George, "Spoken Indian Language Identification using MFCC and Vowel Onset Points", *2023 9th International Conference on Smart Computing and Communications (ICSCC)*, pp. 150-155, 2023, DOI: 10.1109/ICSCC59169.2023.10335007.

[16] A. Jayadi, B. H. Prasetio, S. R. Akbar, E. R. Widasari and D. Syauqy, "Embedded Flu Detection System based Cough Sound using MFCC and kNN Algorithm", *2022 International Conference of Science and Information Technology in Smart Administration (ICSINTESA)*, pp. 1-5, 2022, DOI: 10.1109/ICSINTESA56431.2022.10041610.

[17] Y. Arpitha, G.L. Madhumathi and N. Balaji, "Spectrogram analysis of ECG signal and classification efficiency using MFCC feature extraction technique", *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, pp. 757–767, 2022, DOI: 10.1007/s12652-021-02926-2.

[18] S. Lahmiri, C. Tadj, C. Gargour and S. Bekiros, "Optimal tuning of support vector machines and k-NN algorithm by using Bayesian optimization for newborn cry signal diagnosis based on audio signal processing features", *Chaos, Solitons & Fractals*, vol. 167, 112972, 2023, DOI: 10.1016/j.chaos.2022.112972.

[19] D.R. Yerramreddy, J. Marasani, P. S.V. Gowtham, S. Yashwanth, S.S. Poorna and K. Anuraj, "Speaker Identification Using MFCC Feature Extraction: A Comparative Study Using GMM, CNN, RNN, KNN and Random Forest Classifier", *2023 Second International Conference on Trends in Electrical, Electronics, and Computer Engineering (TEECCON)*, pp. 287-292, 2023, DOI: 10.1109/TEECCON59234.2023.10335892.

[20] S. Becker, M. Ackermann, S. Lapuschkin, K.-R. Muller and W. Samek, "Interpreting and Explaining Deep Neural Networks for Classification of Audio Signals", *arXiv*, 2019, DOI: 10.48550/arXiv.1807.03418.

[21] S.A.A. Yusuf and R. Hidayat, "MFCC Feature Extraction and KNN Classification in ECG Signals", *2019 6th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE*), pp. 1-5, 2019, DOI: 10.1109/ICITACEE.2019.8904285.

[22] V.B.S. Prasath, H.A.A. Alfeilat, A.B.A. Hassanat, O. Lasassmeh, A.S. Tarawneh, M.B. Alhasanat and H.S.E. Salman, "Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbour Classifier – A Review", *arXiv*, 2019, DOI: 10.48550/arXiv.1708.04321v3.

[23] K. Chomboon, P. Chujai, P. Teerarassamee, K. Kerdprasop and N.

Kerdprasop, "An Empirical Study of Distance Metrics for k-Nearest Neighbor Algorithm", *3rd International Conference on Industrial Application Engineering*, 2015, DOI: 10.12792/iciae2015.051.

Sorin MURARU is a PhD Candidate at the Faculty of Cybernetics, Statistics and Economic Informatics within the Bucharest University of Economic Studies. His research focuses on integrating machine learning and deep learning techniques with quantum computing, particularly within the field of economics. At the moment, he has co-authored 8 papers in the larger field of Materials Science. Sorin's work aims to revolutionize economic modeling and analysis through advanced computational methods, contributing to both theoretical and applied advancements in these areas.

Cătălina-Lucia COCIANU, Professor, PhD, currently working with Bucharest University of Economic Studies, Faculty of Cybernetics, Statistics and Informatics, Department of Informatics in Economy. Competence areas: machine learning, statistical pattern recognition, digital image processing. Research in the fields of pattern recognition, data mining, signal processing. Author of 20 books and more than 100 papers published in national and international journals and conference proceedings.