Ministry of Higher Education and Scientific Research

وزارة التعليم العالي والبحث العلمي

Badji Mokhtar Annaba University

جامعة باجي مختار عنابـــــــــة

Faculty of Technology
Department of Computer Science

كلية التكنولوجيا
قســـــــم الاعلام الآلي

Thesis

Presented to obtain the diploma of

Doctorate

Complex Systems Engineering Laboratory (LISCO)

By:

Meriem LOUNIS

Title:

Integration of Machine Learning and Bigdata approaches for pronunciation assessment.

Thesis defended on September 24, 2025 in front of the jury:

N°	Last Name and First Name	Grade	Institution	Quality
1	MOHAMED BENALI Yamina	Prof.	UBMA	Chairman
2	BAHI Halima	Prof.	UBMA	Supervisor
3	BOUNOUR Nora	Prof.	UBMA	Examiner
4	DRISSI Samia	Prof.	Cherif Messadia – Souk Ahras University	Examiner
5	TALAI Zoubir	MCA.	UBMA	Examiner
6	BOUCHA Ismail	MCA.	ENTI	Examiner

Acknowledgments

I am very proud and honored by this achievement, which has been one of my most cherished aspirations ever since I completed my Master's degree. My heartfelt thanks go first and foremost to my father, mother, and husband, whose unconditional support over the past four years has enabled me to reach this milestone.

I am also grateful to my family and friends, whose encouragement and reassurance have helped me through the stresses and uncertainties.

I especially dedicate this work to my children and nephews, hoping that they will help shape a better future.

Finally, I would like to sincerely thank the members of the jury for carefully reading, correcting, and dedicating a significant amount of their time, as well as traveling, to attend this thesis defense. Your insights and presence were invaluable.

THANK YOU, PR. BAHI HALIMA, FOR EVERYTHING, I OWE THIS SUCCESS TO YOU.

Table of Contents

Publications List

Bahi, H., Dendani, B., Lounis, M. (2024). Automatic Pronunciation Assessment and Feedback for Arabic Learners: A Review. International Journal of Asian Language Processing. https://doi.org/10.1142/S2717554524300019.

Lounis, M., Dendani, B., Bahi, H. (2024). One-Class Convolutional Neural Network for Arabic Mispronunciation Detection. the 4'th International Conference on Intelligent Systems and Pattern Recognition "ISPR'2024".

Lounis, M., Dendani, B., Bahi, H. (2024). Anomaly detection with a variational autoencoder for Arabic mispronunciation detection. International Journal of Speech Technology, 27(2), 413-424.

Lounis, M., Dendani, B., Bahi, H. (2024). Mispronunciation detection and diagnosis using deep neural networks: A systematic review. Multimedia Tools and Application. https:// doi. org/ 10. 1007/ s11042- 023- 17899-x.

Lounis, M., Dendani, B., Bahi, H. (2023). Arabic Speech Augmentation for Mispronunciation Detection. The 1 ST National Conference on New Educational Technologies and Informatics, NCNETI'23 (Guelma, Algeria, October 3-4, 2023).

الملخص

هناك اهتمام متزايد بتعدد اللغات، والنطق هو الجانب الأكثر تحديًا في إتقان اللغة. يهدف التدريب على النطق بمساعدة الحاسوب (CAPT)، وهو مجال متخصص في تعليم اللغة بمساعدة الحاسوب، إلى تحسين طرق التعلم التقليدية وتعزيزها باستخدام الأجهزة الرقمية في تعليم اللغة وتعلمها من خلال مجموعة من برامج تقييم النطق. تقترح هذه البرامج وسيلة للكشف عن أخطاء النطق وتشخيصها وتزويد المتدربين بتقييمات وملاحظات تعليمية فردية.

يمكن استخدام تقنية التعلم العميق الخاضع للإشراف لمعالجة مشكلة التصنيف الثنائي للكشف عن النطق الخاطئ؛ ومع ذلك، يتطلب هذا النهج تسجيلات صوتية عالية الجودة مصنفة لكلا الفئتين، الألفاظ المنطوقة بشكل خاطئ والأخرى المنطوقة بشكل جيد. إلا أن ندرة البيانات النوعية والكمية في هذا المجال هي إحدى العقبات الرئيسية. كان هذا هو الدافع الأساسي لإجراء مقترحاتنا الثلاثة المقدمة في هذه الأطروحة، وبعبارة أخرى، حل مشكلة ندرة البيانات لتنفيذ مهمة الكشف عن أخطاء النطق على مجموعة بيانات غير موسومة وغير متوازنة.

في الحل الأول، درسنا قوة النماذج التوليدية في تعلم التمثيلات، خاصةً أداة الترميز التلقائي المتغير (VAE) حيث استخدم للقيام بالكشف عن الأخطاء من خلال تعلم التوزيعات في الفضاء الكامن للنطق" الجيد"ومن ثم، الكشف عن النطق" السيئ" كقيم متطرفة. اما مساهمتنا الثانية فتتمثل في استخدام الشبكة العصبية التلافيفية التمييزية (CNN) واستكشاف قوتها في استخراج السمات من بيانات الكلام لأداء نهج تصنيف من فئة واحدة. أخيرًا، تم اقتراح تقنيات تعزيز البيانات كحل ثالث لزيادة الأشكال الموجية لبيانات التدريب الخاصة بنا، مما يتيح لنا إجراء عملية الكشف عن النطق الخاطئ بطريقة خاضعة للإشراف باستخدام نموذج آلة دعم المتجهات (SVM).

الكلمات المفتاحية: تقييم النطق، التعلم العميق، التعلم الآلي، البيانات الضخمة، تصوير البيانات (الإظهار المرئي للمعلومات والمعطيات).

Abstract

There is a growing interest in multilingualism, and pronunciation is the most challenging aspect of language mastery. Computer Assisted Pronunciation Training (CAPT), a specialized area of Computer Assisted Language Learning (CALL), aims to automate and enhance traditional learning methods by using digital devices in language teaching and learning through a range of pronunciation assessment software programs. These programs propose a means of detecting pronunciation errors, diagnosing them, and providing apprentices with educational and individualized feedback.

A supervised deep learning technique might be used to tackle the binary classification issue for mispronunciation detection; still, this approach requires high-quality labeled audio recordings for both classes, mispronounced and well-pronounced utterances. However, the scarcity of qualitative and quantitative data in this field is one of the main obstacles. This was the primary motivation to conduct our three contributions presented in this thesis, in other words, dealing with the data sparsity problem to carry out a pronunciation error detection task on a mislabeled and imbalanced dataset.

In the first solution, we considered the strength of generative models in learning representations, particularly Variational Autoencoder (VAE). VAE was used to perform an anomaly detection task by learning distributions in the latent space of the "good" pronunciations and then, detecting the "bad" ones as outliers. Our second contribution consists of using a discriminative Convolutional Neural Network (CNN) and exploring its power in extracting features from speech data to perform a one-class classification approach. Finally, Data Augmentation (DA) techniques were proposed as a third solution to augment the waveforms of our training data. DA allows us to perform mispronunciation detection in a supervised manner with the Support Vector Machine (SVM) model.

Keywords: Pronunciation assessment, Deep learning, Machine learning, BigData, Data visualization

Résumé

Le multilinguisme suscite un intérêt croissant, et la prononciation est l'aspect le plus difficile de la maîtrise d'une langue. La prononciation assistée par ordinateur (CAPT) qui est un domaine spécialisé de l'apprentissage des langues assisté par ordinateur (CALL), vise à automatiser et à améliorer les méthodes d'apprentissage traditionnelles. Cela est devenu de plus en plus possible grâce aux dispositifs numériques et des logiciels d'évaluation de la prononciation. Ces logiciels proposent un moyen de détecter les erreurs de prononciation, de les diagnostiquer et de fournir aux apprentis un retour éducatif et individualisé.

Une technique d'apprentissage profond supervisé est généralement utilisée pour résoudre le problème de la classification binaire pour la détection des erreurs de prononciation ; cependant, cette approche nécessite des enregistrements audios étiquetés de haute qualité pour les deux classes, les énoncés mal prononcés et les énoncés bien prononcés. Cependant, la rareté des données qualitatives et quantitatives présente un grand challenge dans ce domaine. Ce qui nous a motivé, dans cette dissertation, à proposer nos trois contributions, en d'autres termes, concevoir un Framework qui est habilité à détecter les erreurs de prononciations sur un ensemble de données mal étiquetées et déséquilibrées.

Dans la première solution, nous avons exploité la force des modèles génératifs dans l'apprentissage des représentations des caractéristiques pertinentes des données, en particulier le Variational Autoencoder (VAE). Le VAE a été utilisé afin d'effectuer une tâche de détection d'anomalies en apprenant les distributions des « bonnes » prononciations dans l'espace latent pour détecter, lors des prédictions, les « mauvaises » prononciations comme des valeurs aberrantes. Notre deuxième contribution consiste à utiliser un Réseau Neuronal Convolutionnel (CNN) discriminant et à explorer sa capacité à extraire les caractéristiques des données vocales pour réaliser une approche de classification à une classe. Enfin, les techniques d'augmentation des données (Data Augmentation : DA) ont été proposées comme troisième solution pour ajouter des données synthétiques à notre base de données d'entrainement afin de créer un équilibre entre les deux classes. La technique de DA nous a permis de résoudre notre problème en adoptant une approche supervisée avec le modèle de la machine à vecteur de support (SVM).

Mots-clés: Évaluation de la prononciation, apprentissage profond, apprentissage automatique, BigData, visualisation des données.

List of Figures

List of tables

Glossary

A

AD
: Anomaly Detection

AGPM
: Acoustic-Graphemic Phonemic Model

AI
: Artificial intelligence

ASMDD
: Arabic Speech Mispronunciation Detection Dataset

ASR
: Automatic Speech Recognition

C

CAI
: Computer-Assisted Instruction

CALI
: Computer-Assisted Language Instruction

CALL
: Computer-Assisted Language Learning

CAPT
: Computer-Assisted Pronunciation training

CBE
: Computer-Based Education

CNN
: Convolutional Neural Network

CTC
: Connectionist Temporal Classification

D

DA
: Datat Augmentation

DER
: Diagnostic Error Rate

DFNs
: Deep Feed-Forward Networks

DISCO
: Development and Integration of Speech technology into COurseware

DL
: Deep Learning

DNNs
: Deep Neural Networks

E

E2E
: End-to-End

EC
: English Center

ELBO
: evidence lower bound

ELSA
: English Language Speech Assistant

EPs
: Error Patterns

ERN
: Extended Recognition Network

ESL
: English as a second language

F

FA
: false acceptance

FAR
: False Acceptance Rate

FFL
: Foreign Language Learning

FLLS
: Foreign Language Learning Systems

FR
: false rejection

FRR
: False Rejection Rate

FST
: Finite State Transducer

G

GANs
Generative Adversarial Networks
GLL
Global Log-Likelihood
GMM
Gaussian Mixture Model
GOP
Goodness-of-Pronunciation

H

HAC
Hierarchical Agglomerative Clustering
HMM
Hidden Markov Model

I

iOS
iPhone Operating System
IPA
International Phonetic Alphabet

K

KL
Kullback-Leibler
KNN
K-Nearest Neighbor

L

L1
Mother tongue
L2
Second languange
LDA
Linear Discriminant Analysis
LGN
lateral geniculate nucleus
LSTM
Long Short-Term Memory

M

MALL
Mobile-Assisted Language Learning
MDD
Mispronunciation Detection and Diagnosis
MFCCs
Mel-Frequency Cepstral Coefficients
ML
Machine Learning
MLPs
Multi-Layer Perceptrons
MSA
Modern Standard Arabic

N

NLP
Natural Language Processing
NN
Neural Network

O

OCC
One-Class classification

P

PCA
Principal Component Analysis
PDF
probability density function
PL
Pseudo-Labeling
PLATO
Programmed Logic for Automated Teaching Operations

R

RBF
radial basis function
RF
Random Forest
RNN
Reccurent Neural Network

S

SLA
Second Language Acquisition
SOTA
State-Of-The-Art
SSL

Self-Supervised Learning
SVM
Support Vector Machine

T

TA
true acceptance
TDS
Time Duration Score
TR
true rejection

U

UPP
Universal Phoneme Posteriorgram

V

VAE
Variational Autoencoder
VAEs
Variational Autoencoders

1 Introduction

1.1 Overview

While the precise onset of foreign language acquisition among humans remains uncertain, it is evident that humans are naturally social beings with a fundamental desire for communication. Thus, we can infer that speaking a language other than one's native tongue has been fundamental to human existence for a while. Dating from the third century, The Hermeneumata or Interpretamenta Pseudodositheana, is one of the earliest known tools for language instruction, teaching Latin speakers in the Greek language [1]*, and "is a peculiar chapter in the history of ancient erudition"* [2]. Although there have been numerous advancements in language learning before the 20th century, they have remained limited and dispersed. Thus, cognitive psychologists combined these efforts at the close of the 20th century into a single area of study known as Foreign Language Learning (FLL) or Second Language Acquisition (SLA) [3].

In parallel, introducing computers into everyday life has been a gradual process marked by significant technological advances, both in hardware and software. This has been a key driver in the development and growth of e-learning and Computer-Based Education (CBE). As a result, the first computer-based training program, PLATO (Programmed Logic for Automated Teaching Operations), was launched in 1960 at the University of Illinois [4]. Since then, the use of computers in foreign language learning has grown considerably with multimedia computing and Web 2.0 technologies, making Computer-Assisted Language Learning (CALL) an increasingly important issue for language learners worldwide.

Figure 1.1: Evolution of Language Instruction Software

CALL was formerly known as CALI (Computer-Assisted Language Instruction), which was a subset of the more general term CAI (Computer-Assisted Instruction), where the initial English

pronunciation software appeared [5]. In the early 1980s, CALI lost favor among language teachers because it seemed to prioritize a teacher-centered instructional strategy. CALL replaced CALI because language teachers preferred a student-centered approach emphasizing learning rather than instruction [6].

The introduction of CALL systems marked an immense shift in language education. It revealed significant growth in CALL research, characterized by increased article quantity, quality, consistency, complexity, and diversity of new areas, indicating a confident research field [7]. The modeling analysis of papers and citations from 2007 to 2019 conducted by [8] revealed that CALL has attracted considerable attention in second language acquisition. This is mainly due to the numerous advantages that the CALL offers, such as: "(a) experiential learning, (b) motivation, (c) enhanced student achievement, (d) authentic materials for study, (e) greater interaction, (f) individualization, (g) independence from a single source of information, and (h) global understanding."[9]. The subsequent coronavirus pandemic has accelerated the adoption of CALL applications in language learning such as MALL (Mobile-Assisted Language Learning), underlining their importance in maintaining continuity in learning [10].

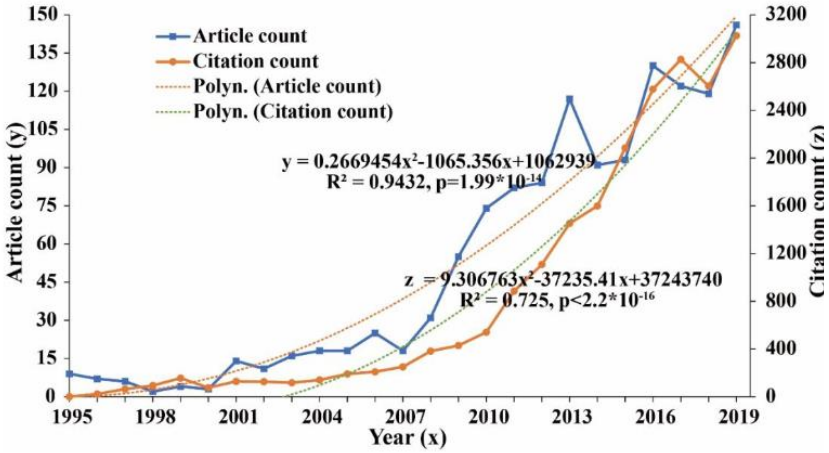


Figure 1.2: Trends of Annual Articles and Citations in CALL Research
After [8]

With innovative instructional strategies and pedagogical design, CALL has broadened its objectives to find the best ways to use available technologies such as interactive whiteboards, educational apps, and online resources in language learning [8]. All this is aimed at achieving the "normalization" evoked by Stephen Bax almost 22 years ago, when technology must become invisible and integrated into everyday practice [11]. As a result, several previously unknown technologies in this field have become widespread. The autonomy that CALL offers

holds great promise in the context of formal and informal learning, inside or outside language classrooms, giving learners access to a wide variety of ubiquitous apps [12]. Thus, Prominent applications in this domain with substantial subscriber bases include the English Center (EC) (https://www.englishcentral.com), ELSA speak (English Language Speech Assistant) (https://elsaspeak.com/en/), Mondly (https://fr.mondly.com/), Busuu (https://www.busuu.com/fr) and Duolingo (https://fr.duolingo.com/). They can be accessed via web browsers or downloaded onto Android or iOS smartphones.

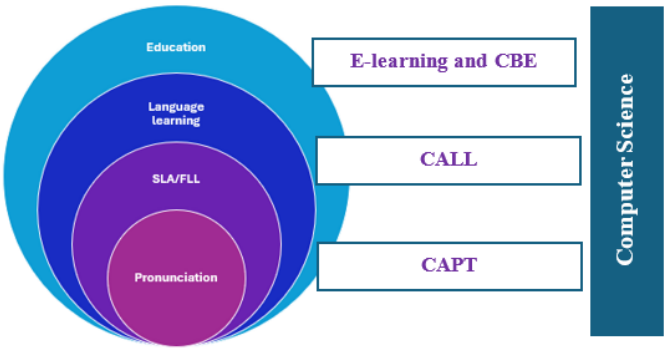


Figure 1.3: Pedagogical and technological advances in foreign language learning

Several factors, including rapid technological advancements, diverse learning contexts, research specializations, and different pedagogical approaches and theories, have led to the division of CALL into several areas and sub-areas to enhance its functionalities. One such domain is Automatic Speech Recognition (ASR), which has prompted research on Computer-Assisted Pronunciation Training (CAPT).

Designed to enhance language learners' pronunciation skills via technology, CAPT aims to provide a personalized and flexible learning environment. It allows users to learn at their own speed, access various multimedia materials, and get instant and specific feedback on their pronunciation accuracy, fluency, and other phonetic aspects [13]. Thus, many CAPT tools have been developed for various languages, including English, Chinese, Dutch, French, and Japanese, such as SpeechRater, DISCO (Development and Integration of Speech technology into COurseware), and Qvoice. With the advances in Artificial intelligence (AI), CAPT applications have significantly enhanced their ability to improve assessment and feedback in educational settings.

Feedback is possible through the Mispronunciation Detection and Diagnosis (MDD) module. The MDD module can correctly pinpoint mispronunciations and state how they are wrong. However, MDD is hard to implement due to the complexity of processing speech data

and the scarcity of large-scale linguistic resources, such as nonnative speech data and human annotations. Nevertheless, the emergence of Deep Learning (DL) boosted the research in pronunciation assessment and pronunciation error diagnosis. In this context, Deep Neural Networks (DNNs) have been integrated into MDD modules as feature extractors, classifiers, or End-to-End (E2E) architectures to discover and model complex relationships among data [14].

In almost all literature studies, MDD is considered as a binary classification problem, treated in a supervised manner, where the classifier decides whether the uttered sequence is well pronounced or not [14]. Meanwhile, the lack of data systematically affects the accuracy of the model, its ability of generalization and the risk of falling into overfitting. Thus, this issue can be circumvented via several unsupervised approaches that have already been tested, approved, and successfully implemented in other fields.

When the data set is imbalanced in favor of the correct data, generative models, such as VAEs, can be implemented in the Anomaly Detection (AD) approach to learn normal behavior and identify deviations. This approach can be adopted in the MDD module to model correct pronunciations, learn relevant features, and compress original data representations in the latent space. The mispronunciations are detected as anomalies during testing.

Similarly, One-Class classification (OCC) can be used in this situation. This technique aims to detect instances of a certain class and differentiate them from any other possible outliers or anomalies. It is frequently applied in datasets that are devoid of representative samples from other classes and only comprise instances of the target class, often known as the "positive" class. Also, this approach can be used in MDD with a system employing the OCC to address the issue of limited ground truth samples.

Data augmentation (DA) is a robust solution that offers various techniques to generate artificial data from available ones. It can be performed with traditional audio data augmentation techniques such as pitch shift, time stretch, and time shift. Thus, the time and money needed to collect data are avoided, enhancing the accuracy of the model and its generalizability. Increasing the size of the dataset enables the use of supervised ML approaches for classification.

1.2 Motivations

The data gathered during the learning processes in Foreign Language Learning Systems (FLLS) is beginning to expand. Although the magnitude of this data is not comparable to that generated by artificial intelligence systems or online consumer services, it should be regarded from a big data perspective due to its volume, variety, and velocity nature [15]. These massive amounts of data, whether in subscription learner databases or linguistic data corpora (native and second-

language learner corpora), are undeniable [3]. However, the data collected is often noisy, inadequately labeled, and frequently imbalanced. To tackle these issues, big data employs machine learning methods focusing on advanced techniques such as representation learning and deep learning. Thus, Machine Learning (ML) can solve data integration problems by extracting features, learning good data representations, and performing classifications in supervised and/or unsupervised manners [16].

Figure 1.4: Big data and speech data

According to "Ethnologue[1], Standard Arabic is fifth among the world's top 10 most widely spoken languages for 2024, after English, Mandarin, Hindi, and Spanish. With 332.5 speakers, 12% are non-native speakers. Arabic fluency provides a competitive advantage in politics, religious education, economy…etc. [17]. Thus, CAPT applications can be essential in improving pronunciation skills among Arabic learners. However, they need large labelled corpora for training, which are hard to come by and frequently not available to the general public. For instance, access to the King Saud University (KSU) Arabic Speech Database (https://catalog.ldc.upenn.edu/LDC2014S02), a useful resource, is limited. The development of CAPT systems suited to the Arabic language is severely hampered by this data restriction.

To solve this difficulty, it is critical to investigate sophisticated deep learning approaches that can compensate for data scarcity. Unsupervised learning and data augmentation are promising approaches that enable the best use of limited data while also offering more robust and scalable pronunciation evaluation models for under-resourced languages like Arabic.

[1] What are the top 200 most spoken languages? », Ethnologue. https://www.ethnologue.com/insights/ethnologue200/

1.3 Main contributions

In this thesis, we investigated pronunciation assessment in the context of CAPT systems. Thus, as a first contribution, we have attempted to review the literature on mispronunciation detection using deep learning methods in CAPT systems. Our statistical analysis was published on January 9, 2024, in the “Multimedia Tools and Applications journal”. This enabled us to detail the MDD sub-title in the background chapter comprehensively and then guided us in proposing some solutions for this research area.

As already said, this thesis is motivated by the absence of work in CAPT for Arabic language, which is sorely lacking in data. We focus on the automatic evaluation of Arabic pronunciation over MDD. To carry out our task, we used the ASMDD dataset (Arabic Speech Mispronunciation Detection Dataset) to train and test our model. As this dataset is mislabeled and completely imbalanced, we have used a generative model based on representation learning to learn the data distributions in the latent space. After training, the system could identify as "abnormal" any representations that fall outside of the boundary. Experimental results show the effectiveness of our approach compared to the SOTA DL (State-Of-The-Art Deep Learning) classifiers with an imbalanced dataset. The study was published on 25 June 2024 in the “International Journal of Speech Technology”

Consistently, to address the challenges present in our dataset, we published a study in “The 4'th International Conference on Intelligent Systems and Pattern Recognition ISPR'2024” proposing a One-Class Classification method for mispronunciation detection using a Convolutional Neural Network (CNN) as a discriminative model trained in a semi-supervised manner. The model learns exclusively from well-pronounced utterances and must distinguish between well-pronounced and mispronounced ones in the test stage. This enabled us to provide a second solution to the data set imbalance problem.

Furthermore, to address the issues related to the quality and quantity of the dataset in under-resourced languages, we were motivated by the success of Data Augmentation (DA) techniques, which we employed to artificially expand the audio samples in our dataset. Rebalancing our data allowed us to handle the MDD task as a binary classification problem in a supervised approach with a machine learning model. This proposition was published in the first national conference, “National Conference on New Educational Technologies and Informatics NCNETi’23”

The promising results obtained in our various experiments prove that the proposed methods can be practical in real-world applications, as the volume and variety of data

collected by the speakers are growing increasingly. These data are typically unordered and unlabeled; however, they have significant value and must be integrated into CAPT databases. Unsupervised machine learning approaches permit their utilization in their inherent form. Additionally, data augmentation can be implemented in big data when discriminative models are used for the classification task. In this case, DA can rebalance the dataset and thus enhance the model accuracy.

1.4 Thesis outline

The dissertation is organized into seven chapters. The structure of each one is as follows:

Chapter 1 includes an introduction that overviews the history of foreign language learning, highlights its importance, and discusses the technological advances that have helped automate the field. It also presents our main contributions and outlines the structure of the thesis.

In Chapter 2, we explored CAPT systems based on traditional methods, highlighting their limitations and emphasizing the need to modernize them using deep learning architectures. This review of the state of the art guided us in selecting the most appropriate approaches to our problem.

Chapter 3 introduces the fundamentals of deep learning by comparing its two main approaches from a functional perspective. This comparison clarifies how each method operates, allowing us to choose the best way to tackle our specific research issues.

Chapter 4 presents the first major contribution of this dissertation; the use of an anomaly detection approach based on a variational autoencoder (VAE) for detecting pronunciation errors in Arabic. This method introduces a different point of view in two key areas. First, it focuses on the Arabic language, which has received little attention in pronunciation assessment research. Secondly, it uses VAE as a generative model, a technique that has yet to be explored in this field. This combination aims to address the challenges posed by poorly resourced languages and improve pronunciation error detection efficiency.

We have contributed to this field with two other approaches proposed in Chapter 5. The first consists of using Convolutional Neural Networks and their efficacy in extracting features and learning patterns from data. CNN is used in the One-Class Classification (OCC) approach, an approach known for its power to identify instances of a specific class in scenarios where negative examples are scarce or absent. The second explores audio data augmentation techniques used to expand and rebalance the ASMDD dataset to train a Support Vector Machine

(SVM) model as a binary classifier. Applying these techniques led to promising results, demonstrating improved model accuracy and enhanced performance in detecting pronunciation errors in a supervised manner.

Finally, in Chapter 7, we concluded this work by discussing the presented methods and potential future research in pronunciation error detection, intending to recommend diagnostic and feedback mechanisms.

2 Computer-Assisted Pronunciation Training CAPT

2.1 Introduction

In today's globally interconnected world, people live in closely knit communities, and to communicate with others productively, learning various languages is a must. As a result, Computer-Assisted Language Learning (CALL) is a dynamic research area that challenges the most difficult part of language learning: pronunciation. According to the communicative language teaching theory, pronunciation is learned by exposure to the language rather than form-focused instruction. Research into the automation of pronunciation learning has shown that this is feasible via Computer-Assisted Pronunciation Training (CAPT) [18]. In this context, the pronunciation assessment is a crucial component in CAPT systems that aims to detect mispronunciations and provide learners with informative feedback. However, building reliable assessment systems requires the availability of large, dedicated speech corpora, which is not possible, particularly for low-resource languages such as Arabic. Due to the complexity of the topic, research on automatic pronunciation assessment is still in its early stages. It is an open, complicated problem at the intersection of various study areas, including signal processing, natural language processing, pattern recognition, etc. In this chapter, we review previous research on Computer-Assisted Pronunciation Training (CAPT) and detail the crucial elements of the assessment component.

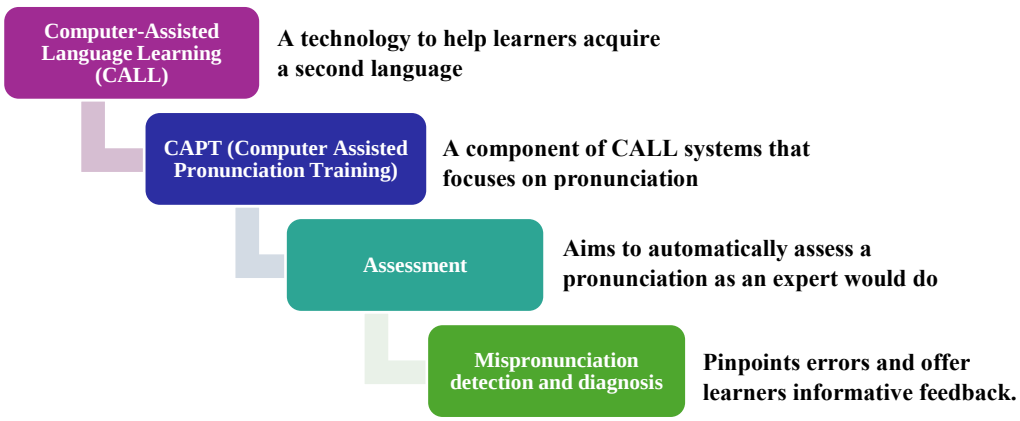


Figure 2.1: Location of the study area

2.2 Front and Back Ends of Computer-Assisted Pronunciation Training

The critical component of a CAPT system is the assessment module that aims to evaluate pronunciation over 1) pronunciation scoring and 2) mispronunciation detection and diagnosis (MDD). The first is a comprehensive evaluation of fluency, typically consisting of one or more sentences, with students receiving feedback as a numerical score. While the second is more specific, it focuses on identifying errors at the word or sub-word level and provides corrected feedback.

MDD refers to the foundational technology and infrastructure supporting the functioning of CAPT software applications. It operates predominantly in the background and remains invisible to end-users. Some researchers emphasized correct voicing of vowels, others tried to identify frequently occurring mispronunciations, and others provided learners with personalized feedback that is more accessible with reduced anxiety and individualized instruction [19].

MDD is the first and most crucial phase in implementing a CAPT system. It has a profound impact on the quality of feedback provided to learners. By offering rapid, consistent, and objective evaluations that correlate well with human assessments, these systems enhance the learning experience and support effective language acquisition over frontend multiple designs, such as reading exercises, question-answering, and task-oriented conversation systems [20].

2.3 Pronunciation Assessment

The pronunciation assessment, which is often related to CAPT systems, refers to a methodical task that aims to assess the correctness and fluency of spoken language, most of the time, this is done throughout a language learning framework. The pronunciation assessment combines both manual and automatic techniques to evaluate the learner's pronunciation proficiency. Manual assessment involves human experts to rate the pronunciation, while automatic assessment refers to technologies to provide objective feedback on pronunciation accuracy.

According to [21], Pronunciation assessment denotes two ways to provide learner feedback: 1) ASR state-of-the-art approaches and 2) mispronunciation detection and diagnosis approaches. In the light of the above, the following diagram was drawn up.

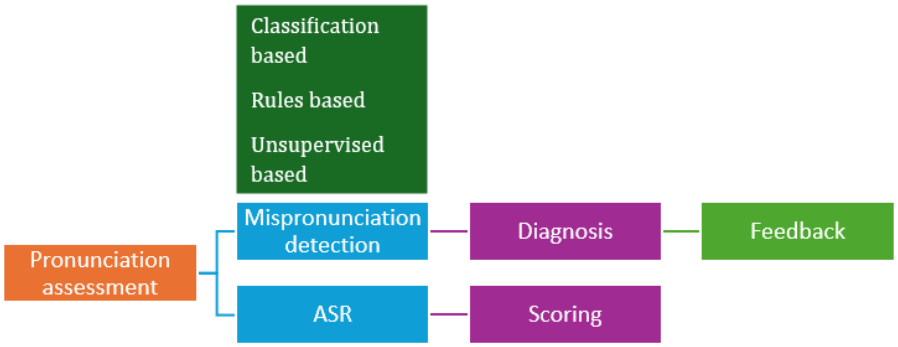


Figure 2.2: Pronunciation assessment

From our point of view, and given the rise of deep learning, which has changed how almost every domain is designed and implemented, the landscape of automatic pronunciation assessment has changed considerably considering the before and after this boom.

In what follows, we would like to offer a comparative overview of pronunciation assessment methods before and after using DL—in other words, traditional approaches versus deep learning approaches.

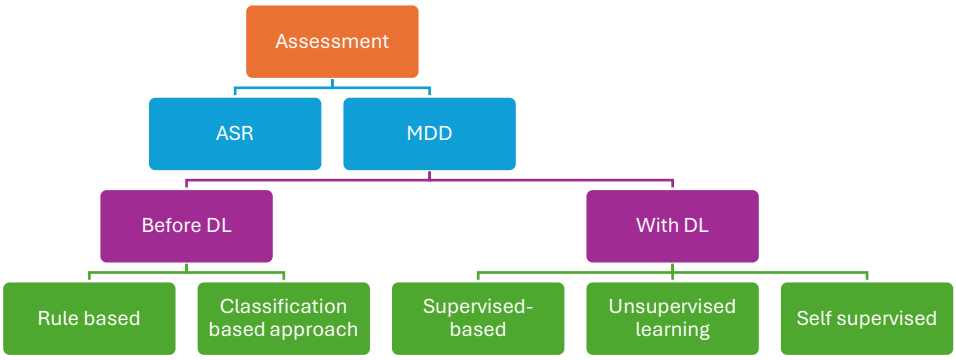


Figure 2.3: Pronunciation assessment traditional Vs. DL methods

Below, we review the work on MDD and feedback in CAPT systems, with a focus on the former, which is the main subject of this thesis.

2.4 Automatic Speech Recognition (ASR)

Automatic Speech Recognition (ASR) is a technology that converts spoken language into written text, allowing users to interact with devices using their voice. The process involves

capturing spoken audio, extracting relevant features, modeling phonemes based on extracted features, and decoding using algorithms to match predicted phonemes to words. The different stages of an ASR system are provided in Figure 2.4:

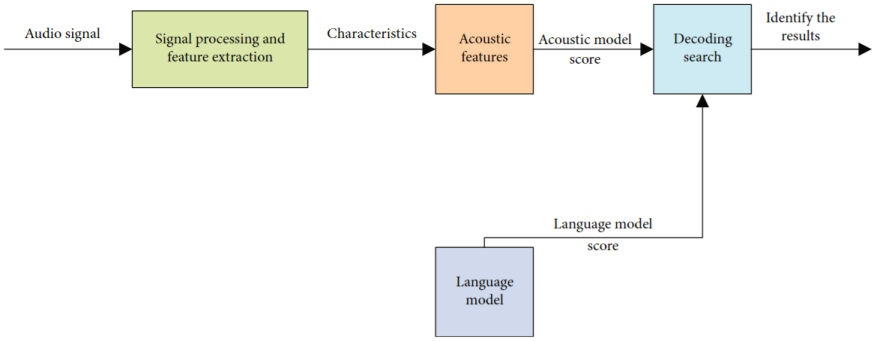


Figure 2.4: Speech recognition process

After [22]

ASR was an essential element in CAPT systems, offering confidence scores for error detection. The most commonly used is Likelihood-Based scoring for phonetic error detection, which generally focuses on the Goodness-of-Pronunciation (GOP) score. Introduced by Witt and Yong in 1999 [23], "The GOP approximates the posterior probability of each phoneme by taking the ratio between the forced alignment likelihood and the maximum likelihood of the free-phone loop decoding using a Hidden Markov Model (HMM) acoustic model." [21]. In other words, phonemes are accepted or rejected by computing threshold rates to measure pronunciation quality, which is not informative to the learner because it cannot diagnose mispronunciations.

Conversely, pronunciation error detection assessment is more challenging. It can be performed at the phoneme level, providing rich information about the kind and position of errors the learner makes, motivating and encouraging him to progress. Traditional and modern approaches have been developed to achieve this objective, and the next sections will be devoted to reviewing them.

2.5 Mispronunciation detection and diagnosis

While robust ASR should perform well with all variations, including dialects and non-native speakers, mispronunciation detection and diagnosis (MDD) should mark phonetic variations from the learner, which may occasionally be subtle differences. In many cases, MDD is more challenging to model than the vanilla automatic speech recognition system, which converts speech into text regardless of pronunciation errors [24].

2.5.1 Before Deep Learning: Traditional Approaches

Early automated systems used basic algorithms for speech recognition, often focusing on phoneme recognition without considering broader contextual factors like prosody or speech fluency.

2.5.1.1 Rule-based approach

The rule-based approach offers the advantage of pinpointing the location(s) and type(s) of phonetic differences between the canonical pronunciation and the foreign language learner transcription. Developed by domain experts, this approach can only succeed if the learner produces errors that are a priori knowledge of expected mispronunciation rules.

The ERN (for Extended Recognition Network) for mispronunciation detection and diagnosis was one of the traditional methods with a rule-based approach. The architecture of such a system is presented below.

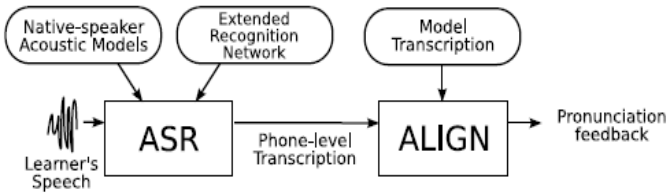
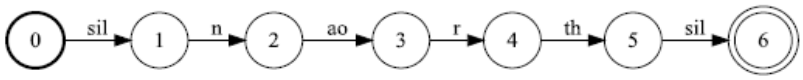


Figure 2.5: Extended Recognition Network Architecture

Functionally, the system elicits the learner to pronounce a given corpus utterance; after recording the learner's speech, recognizing the input signal according to the ERN, and aligning it with the pre-listed transcription of the model native speaker, the differences are diagnosed as pronunciation errors. Regarding implementation, the ERN is a Finite State Transducer (FST) representing phonological rules based on arbitrary canonical pronunciation. The figure presents the ERN architecture after [25]; the Recognition Network includes standard English pronunciation, which is extended to the common mispronunciation of learners.

The rules are given in the following form: $\varphi \rightarrow \psi \,/\, \lambda \rightarrow \rho$

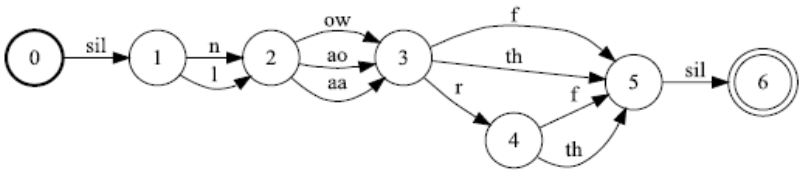
This rule is read as follows: φ in the target language may be pronounced as ψ by the learner when following λ and preceding ρ.



1. Standard recognition network

ao → ow
ao → aa
r → / V
th → f
n → l/ #

2. Phonological rules for the word "north" pronunciation: albeit limited method: "even though" or "although."



3. The Extended Recognition Network of "north"

Figure 2.6: ERN functionnung

Later, [26] achieved the MDD task by comparing the recognized transcriptions with the canonical ones using the Acoustic-Graphemic Phonemic Model (AGPM)

2.5.1.2 Classification-based

In this approach, pronunciation error detection is considered a binary classification problem [25], with "correct" or "incorrect" pronunciations through a set of conventional state-of-the-art classifiers like Linear Discriminant Analysis (LDA), SVM, and decision trees [27] [28] [29].

[29] proposed a CAPT tool for young Algerian pupils using a decision tree algorithm for classification. A decision tree is a graphical classifier and supervised learning tool that splits individuals into homogeneous groups based on discriminating attributes. It can be interpreted as rules, thus facilitating its understanding.

As illustrated in Figure 2.7, the process in [29] involves a learner reading a text, capturing and transmitting speech signals to a speech recognition engine. This generates an acoustic model of pronunciation, which is compared to possible pronunciations in a database. A pronunciation score is determined based on likelihood computation, Time Duration of Speech (TDS), and total phonemes of each pronounced word. The decision tree is then used to determine the pronunciation's quality, accepting or rejecting the pronunciation.

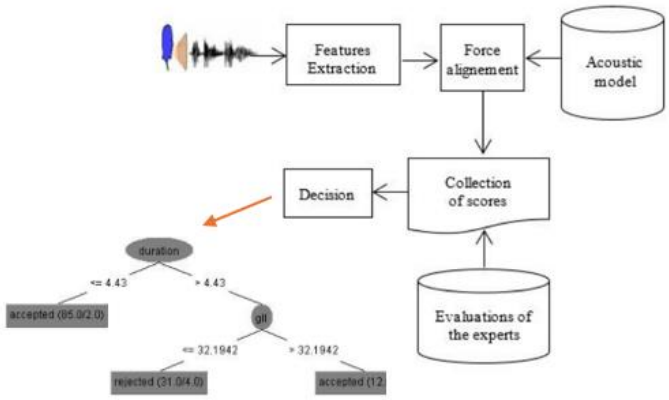


Figure 2.7: Arabic Pronunciation Assessment with Decision Tree

After [29]

2.5.2 Limitations of Traditional approaches

Traditional approaches to automatic pronunciation assessment face many significant limitations that reduce their performances.

- Variability of Human Evaluation: The conventional approaches highly depend on expert humans; in this case, evaluations are more prone to be different due to the subjective interpretations of the pronunciation quality. This variability negatively impacts the issue of the assessment.
- Accents and Dialects: Accents and dialects are inherent challenges in language learning, and conventional methods frequently encounter difficulties with accents and dialects as they are unable to handle this aspect. Indeed, non-native speakers have distinct phonetic backgrounds, and the evaluators could not evaluate them accurately.
- Generalized Feedback: Traditional methods provide broad feedback that fits many situations. Such feedback is not enough helpful for the learners to identify and correct their mistakes effectively.
- Static Assessment Tools: Traditional assessment tools do not adapt to individual learner needs or progress over time, which can limit their effectiveness in promoting improvement.

- Labor-Intensive Evaluation: The process of evaluating pronunciation manually is time-consuming and requires significant human resources, limiting the scalability of traditional assessment methods.
- Specific knowledge of L1 or L2: using some conventional techniques is tightly related to knowledge of L1 or L2 for conducting rules in L1/L2 pairings.

The limitations of traditional approaches for automatic pronunciation assessment highlight the need for more advanced, technology-driven solutions that can provide consistent, objective, and detailed feedback. Integrating automated systems like Automatic Speech Recognition (ASR) and machine learning techniques can address many of these challenges, offering a more effective way to evaluate and enhance pronunciation skills in language learners.

2.5.3 With Deep Learning

The explosive growth of big data, DL technologies, and the use of clouds is speeding the advancement of speech recognition and assessment. Deep learning is a robust machine learning technique founded on artificial neural networks, proficient in processing extensive feature sets and managing unstructured data. It demonstrates efficacy for less complex problems but necessitates access to large amounts of data. Deep learning models comprise many artificial neural networks with several hidden layers capable of executing artificial intelligence tasks such as image recognition, audio recognition, picture retrieval, and natural language comprehension. The architecture of deep learning directly influences its capacity to model and articulate features. In contrast to conventional networks, deep learning models can achieve superior accuracy, occasionally surpassing human performance [22].

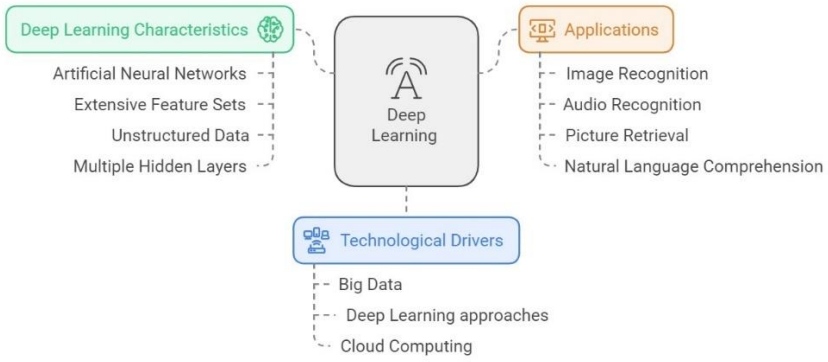


Figure 2.8: DL characteristics, technological drivers, and applications

2.5.3.1 Supervised-based approach

Seventy-nine percent (79%) of the studies in this field, according to [14], fit under the supervised approach, considering that MDD is typically treated as a classification problem. In this new era of Deep Learning, researchers are devoting much attention to exploiting the power of these models and building classification-supervised pronunciation systems.

The first DNNs-based MDDs were used primarily to replace the old classification methods, feature extraction, and acoustic modeling [26] [30] [31]. Later, the use of DNNs was extended to solve various problems that had been an obstacle to achieving binary classification in MDD. This necessitated the incorporation of other models, such as CNN, which became the new SOTA in MDD, whether stand-alone or combined with other models. Thus, different DL architectures were used in articulatory modeling [32] [33], speech representation [34], intonation classification [35], phonetic embeddings [36], denoising [37], and end-to-end architectures [38].

The paper "Improving Mispronunciation Detection of Arabic Words for Non-Native Learners Using Deep Convolutional Neural Network Features" highlights significant contributions to mispronunciation detection, particularly for Arabic words [39]. The study demonstrates that features extracted from the AlexNet model outperform transfer learning-based and traditional handcrafted features in detecting mispronunciations of Arabic words. The authors propose a CNN features-based model that utilizes automated feature selection methods, improving mispronunciation detection accuracy.

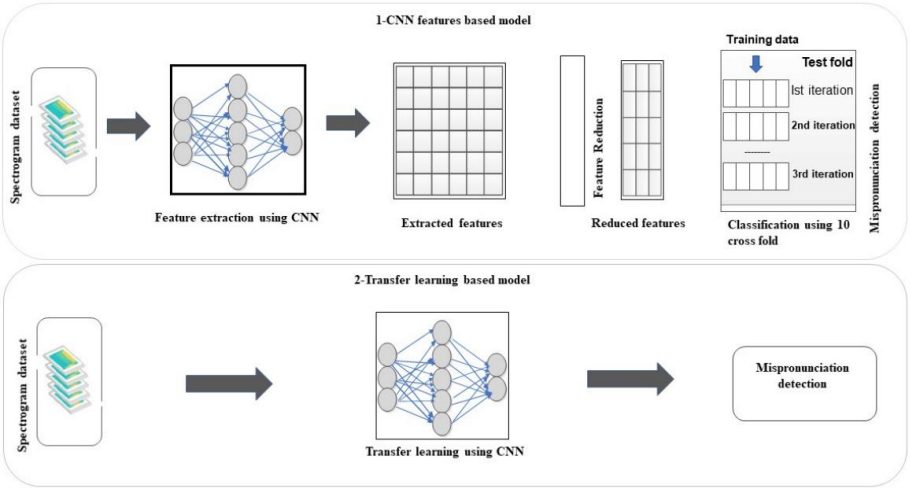


Figure 2.9: The use of CNN in feature extraction and transfer learning

After [39]

This paper compares the proposed methods with the traditional approaches, and it concludes that the CNN model achieved the best accuracy of about 93.20%. In particular, the conducted study research addresses the challenges of mispronunciation detection in Arabic; Arabic being a specific language with many difficulties carried by its phonetic and linguistic structure as well. To conduct the comparison, the study employs three different machine learning classifiers: K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Random Forest (RF). This study contributes to the understanding and application of deep learning techniques in language learning and mispronunciation detection processes, particularly for Arabic language.

After that, the supervised MDD approach has used more complex DL architectures to provide complete solutions for CAPT systems like Connectionist Temporal Classification (CTC) and Recurrent Neural Networks (RNN), improving accuracy and providing detailed feedback [24] [40] [41].

Although these classification methods are promising, difficulties such as the variability of pronunciation errors and the lack of large training datasets to represent them limit the generalization of the models. This forces researchers to investigate models that do not depend on labeled datasets, in this case, moving towards unsupervised learning.

2.5.3.2 Unsupervised-based approach

[42] proposed an approach to overcome the scarcity of non-native speech corpora and develop CAPT systems. The paper presents a new framework for the supervised detection of pronunciation Error Patterns (EPs) using hierarchical Multi-Layer Perceptrons (MLPs). This approach enhances the accuracy of EP diagnosis, making it valuable for language learners and educators. The unsupervised EP discovery framework uses the Hierarchical Agglomerative Clustering (HAC) algorithm to analyze sub-segmental variations within phoneme segments. The research also introduces the Universal Phoneme Posteriorgram (UPP) as frame-level features for supervised detection and unsupervised discovery of EPs. The paper addresses challenges in distinguishing EPs from canonical pronunciations and among different EPs of the same phoneme. Preliminary results show promising results, advancing understanding and methodologies for detecting and discovering pronunciation errors.

Later, the unsupervised approach was adopted for use in end-to-end systems. The authors in [43] present a novel mispronunciation detection and diagnosis model using advanced deep learning techniques. It introduces a multi-feature and multi-modal model that uses the Squeezeformer encoder as the audio encoder and a transformer as the decoder. The model incorporates phoneme length information into speech features, allowing a more comprehensive

understanding of the speaker's pronunciation patterns. A secondary decoding mechanism is used to reprocess the initially decoded sequence, addressing the lack of a priori knowledge during the first decoding stage. The performance metrics show substantial improvements, with the F1 index increasing from 0.4060 to 0.7943 and diagnostic accuracy improving from 83.93% to 88.45% compared to the baseline model. The authors suggest incorporating a multi-task learning model to enhance the training process. The experiments were conducted on the PSC-Reading Mandarin mispronunciation detection and diagnosis dataset, ensuring consistency with real test scenarios. These contributions advance the state of the art in mispronunciation detection and diagnosis, providing a robust framework for future research and applications.

2.5.3.3 Self-supervised based approach

Self-supervised approaches for mispronunciation detection leverage unlabeled data to enhance model performance, particularly for second-language learners. Techniques such as Wav2vec 2.0 are utilized for pretraining on unlabeled L2 speech, allowing effective fine-tuning with minimal labeled data [44]. This approach achieved F1 scores exceeding 0.610, indicating effective mispronunciation detection.

The authors in [45] leverage unlabeled L2 speech via a pseudo-labeling (PL) procedure and extend the fine-tuning approach based on pre-trained Self-Supervised Learning (SSL) models. The method improved robustness and reduced phoneme error rates by 5.35%, leading to a 2.48% improvement in mispronunciation detection F1 scores.

2.6 Feedback

One of the main challenges of technology-based language learning is providing adequate interactivity and intelligent, personalized feedback; these areas still need to be improved for CAPT [46]. Precise and immediate feedback is crucial for assisting learners in recognizing inconsistencies between their output and the target L2 model [13]. More exact and explicit feedback facilitates learners' mastery of specific skills. Constructive feedback should concentrate on error patterns and may utilize examples to demonstrate effective behavior [47].

A visual simulation-based CAPT system captures a learner's speech, analyses it, and delivers feedback via explanatory visuals, animated characters, and compared videos. The method is straightforward and efficient, particularly for young learners and those with hearing difficulties [48]. Based on this approach, the author in [49] has developed a tool to give auditory and visual feedback to the learner. The system provides text scripts for students to read, follow

prompts, and evaluate their speech based on these prompts. It generates informative/corrective feedback in two ways. The first one is visual feedback: "animating the head", drawing a series of snapshots of the vocal tract pronouncing the phoneme for the correct and incorrect spelling, which helps the learner to adjust their pronunciation. The second one is auditory feedback, utilizing a text-to-speech system to get the right phone length and the stress assignment to enhance pitch variation by listening to the correct pronunciation. An illustration of the approach is shown in the diagram below.

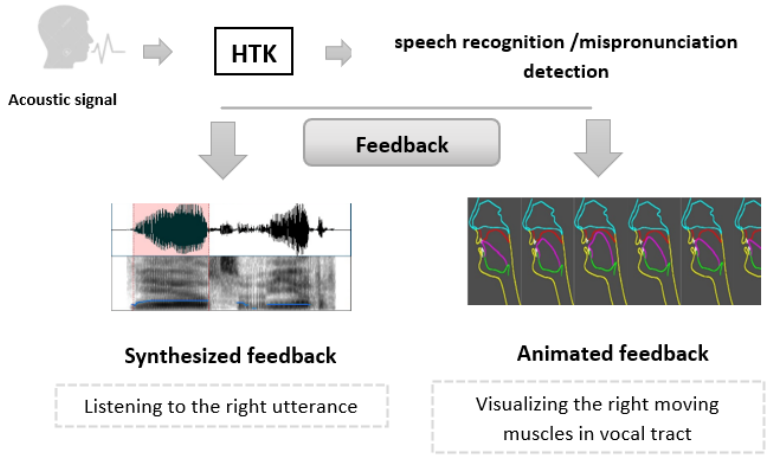


Figure 2.10: An example of visualized feedback

After [49]

The previously described reading practice method is the predominant technique for providing feedback to L2 learners. Nonetheless, alternative methodologies, such as Question Answering in game-based systems, present many advantages, notably through improved engagement, motivation, and the acquisition of practical skills [20] [48]. The authors in [50] examined the utilization of educational games intending to improve English pronunciation among Spanish speakers by introducing an innovative mobile app for pronunciation training over different competitive events. They have proven that these games have the capacity to enhance social engagement and enrich educational experiences. Finally, gamification programs such as Duolingo improve vocabulary and pronunciation abilities, rendering the learning process entertaining and accessible [51].

Traditional approaches to pronunciation training often rely on explicit corrective feedback and visual aids, while novel methods leverage advanced technologies for more interactive and personalized learning experiences. While traditional methods have proven

effective, they may lack the adaptability and engagement that novel approaches provide, potentially limiting their effectiveness in diverse learning environments.

2.7 Pronunciation constructs and pronunciation error patterns

There is no clear distinction between correct and incorrect pronunciation but a spectrum ranging from unintelligibility to native speech. Since pronunciation errors are challenging to measure, they can be classified into two categories: (a) objective and (b) subjective evaluations [24].

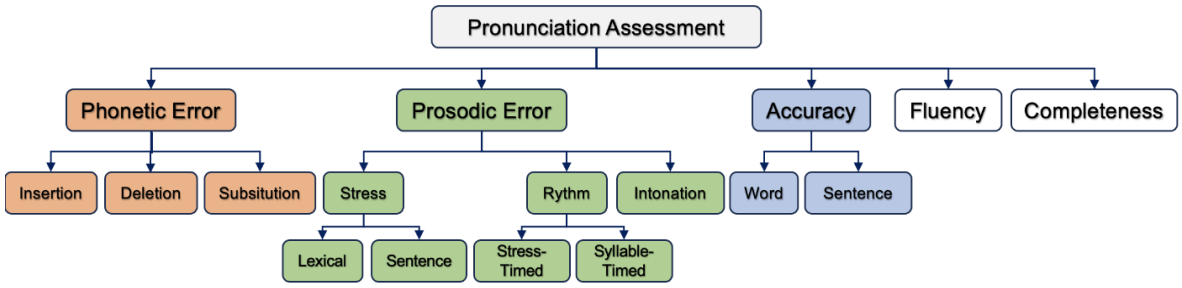


Figure 2.11: Pronunciation errors for mispronunciation detection and diagnosis After [24].

2.7.1 Objective evaluation

The goal of developing L2 pronunciation is to communicate effectively in the target language. This can be achieved using three pronunciation constructs: intelligibility, comprehensibility, and accentedness.

The accuracy of sound, word, and phrase can describe intelligibility; it pertains to the learner's correct pronunciation of each phoneme or word. On the other hand, comprehensibility represents the difficulty or the ease that listeners encounter while interpreting L2 discourse. Comprehensibility is generally determined by fluency, characterized by the fluidity of articulation and appropriate use of pauses. Finally, accentedness (linguistic native-likeness) is defined as listeners' perception of how L2 speech is influenced by their mother tongue and/or characterized by other non-native elements [24].

2.7.2 Subjective evaluation

Understanding the patterns of pronunciation errors among language learners is crucial for developing practical assessment tools by identifying areas where students struggle. Various

studies have identified specific patterns and types of errors made by students. They can be categorized into two main types: segmental and suprasegmental errors [24] [52].

Phonemic (segmental) errors involve incorrect articulation of phonemes. Phonemic errors are caused by adding, deleting, and substituting one phone with another. On the phonemic side, there are the 'extreme' mistakes where phonemes may be replaced with another phoneme, removed, or, on the other hand, embedded. An example of phonemic errors is misunderstanding the phoneme distinction between /p/ and /b/ in Arab ESL learners (English as a second language), such as pird and brison instead of bird and prison [53].

Stress, intonation, and rhythm are the main components of suprasegmental pronunciation problems beyond individual sounds. Common types include intonation patterns that can result in misunderstandings, syllable stress errors that may reduce speech intelligibility, and rhythm issues that cause unnatural speech sounds to non-native speakers. This is particularly true for tonal languages, where pitch variation can lead to words with different meanings, such as Mandarin [24].

Segmental and suprasegmental pronunciation errors are pivotal in language education. Although phonemic errors are frequently more apparent, prosodic errors can also be detrimental as they impact the natural rhythm and significance of speech, underscoring the necessity of a balanced methodology in teaching pronunciation. Tackling these challenges through focused practice and feedback can considerably enhance learners' pronunciation, improving their overall communicative competence.

2.8 CAPT Datasets

Pronunciation assessment datasets are challenging and expensive; most research focuses on private data. The graphic (Figure 2.12) below illustrates a quantitative analysis of the number of data sets in Arabic, contrasted with the data sets for the two most studied languages at CAPT, English and Mandarin.

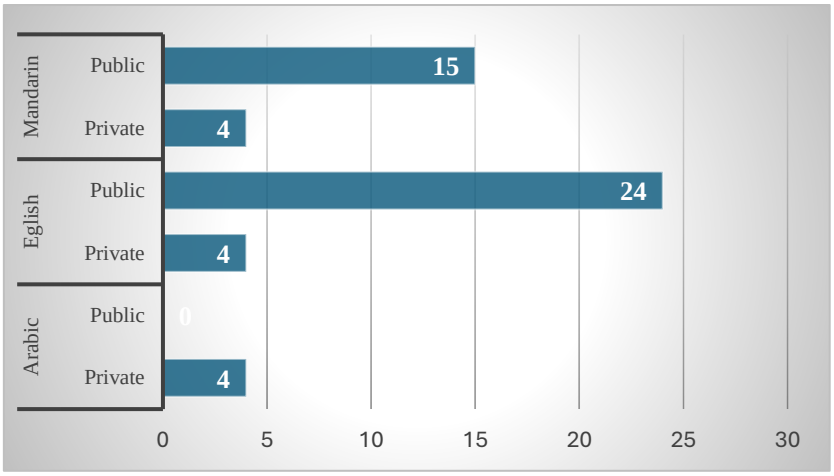


Figure 2.12: Number of datasets used for MDD for English, Arabic, and Mandarin According [14]

The analysis highlights that English is the dominant selection for the target language L2 in CAPT due to its wide use in interacting with citizens in most countries. China is an emerging nation with millions of projects worldwide, which explains why Mandarin occupies second place in CAPT studies in terms of the number of dedicated corpora. In contrast to Arabic, which has no public corpora, the absence severely limits the language's development.

Table 2.1: Arabic, English, and Mandarin Corpora

Arabic		English		Mandarin	
Private	**Public**	**Private**	**Public**	**Private**	**Public**
KSU speech corpus		"RA" corpus	CU-CHLOE corpus	L2 Mandarin	BLCU Corpus
L2 Arabic		L2 English	EMA-MAE Corpus	PSC-Reading dataset	ICALL Corpus
		Speech Accent Archive pronunciation corpus + Noise-92 corpus	ETRI Corpus (Electronics and Telecommunications Research Institute)		Mandarin annotated spoken (MAS) native and non-native Corpus
			Interactive Spoken Language Education (ISLE) Dataset		
			L2-Arctic Corpus		
			Supra-CHLOE corpus		

The CU-CHLOE [54] and L2-ARCTIC [55] corpora are extensively utilized in L2 English research. The CU-CHLOE corpus comprises 110 Mandarin and 100 Cantonese speakers, organized into five categories. Expert linguists have systematically annotated all segments, offering dependable and accurate linguistic data. The L2-ARCTIC corpus is a specialized speech corpus for speech conversion, accent conversion, and mispronunciation detection

among non-native English speakers. The dataset comprises 26,867 utterances from 24 non-native speakers whose first languages include Hindi, Korean, Mandarin, Spanish, Arabic, and Vietnamese. Only 150 utterances are manually designated to detect segmental mispronunciation issues. The predominant dataset for Mandarin is iCALL [56], consisting of 90,841 spoken utterances from 305 speakers. The speaker composition guarantees gender equality and represents a diverse spectrum of adult Mandarin learners.

Computer-Assisted Pronunciation Training (CAPT) datasets encounter many challenges, such as class imbalance, limited diversity, poor audio data quality, insufficient training data, absence of contextual information, and missing annotations. Class imbalance may lead to models performing in majority classes while failing in minority classes, altering the training process. Limited diversity may constrain the model's capacity to comprehend and assess various pronunciation variations. Poor audio quality can lead to inaccuracies in pronunciation assessment. Insufficient training data might affect model performance, while lacking contextual information may lead to inaccurate evaluations. Inconsistent labels affect the models' overall reliability. Some researchers have opted to generate their own datasets to circumvent the issues related to the datasets mentioned above. For example, the Arabic-CAPT dataset constructed by [57] is a corpus for detecting mispronunciations in Arabic. It comprises 62 non-native speakers from 20 distinct nationalities, with 2.36 hours of speech data.

2.9 Performance measures

Diverse metrics have been suggested to assess the quality of mispronunciation detection. Figure 2.13 illustrates the hierarchical structure of these indicators. These scores, mainly at the phoneme level, are designated to evaluate the efficacy of a pronunciation assessment system as follows:

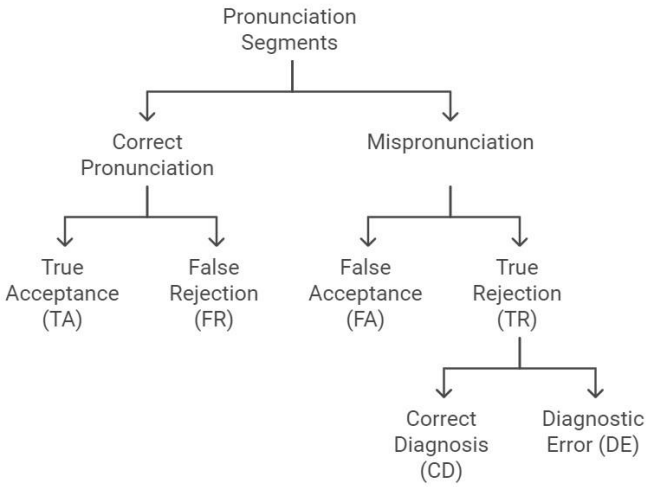


Figure 2.13: Evaluation metrics for MDD

After [42]

Figure 2.13 indicates that, for correct pronunciation, true acceptance (TA) and false rejection (FR) serve as the system's judgment parameters. In contrast, false acceptance (FA) and true rejection (TR) represent the potential evaluations for mispronunciation. From these four scores, the subsequent values may be calculated:

a. Accuracy: Evaluates pronunciation accuracy by comparing uttered words with a reference standard. This comprises assessments at the phoneme, syllable, and word levels, providing comprehensive feedback on particular problems such as deletions, insertions, and mispronunciations.

$$Accuracy = \frac{(TA\ +\ TR)}{TA + TR + FA + FR}$$

b. False Rejection Rate (FRR): the percentage of accurately pronounced instances that the system erroneously classifies as mispronounced. FRR is calculated as the ratio of correctly identified mispronounced phonemes (FR) to the total number of correct phonemes (TA + FR).

$$FRR = \frac{FR}{TA + FR}$$

c. False Acceptance Rate (FAR): the proportion of incorrectly uttered segments that the system accurately accepts.

$$FAR = \frac{FA}{TR + FA}$$

d. Diagnostic Error Rate (DER): the percentage of wrong diagnoses (DE) relative to the total number of correctly rejected pronunciations (TR).

$$DER = \frac{DE}{TR}$$

The choice between these metrics depends on the specific context of the classification problem, particularly regarding class balance and the implications of false positives and negatives. However, the F1 score measure is especially pertinent in imbalanced datasets when one class predominates over the other. It ensures that precision and recall are considered, providing more information than accuracy alone in imbalanced class distributions.

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Where:

- *Precision:* The ratio of true positive predictions to the total predicted positives:

$$Precision = \frac{TA}{TA + FA}$$

- *Recall:* The ratio of true positive predictions to the total actual positives:

$$Recall = \frac{TA}{TA + FR}$$

2.10 Arabic Assessment

2.10.1 Language particularities

A comprehensive study of the Arabic language's distinctive features facilitates error diagnosis and the development of appropriate feedback mechanisms. This is why we want to detail the main properties of the language in this section.

The Arabic language comprises three forms: classical Arabic, modern standard Arabic (MSA), and colloquial Arabic. Classical Arabic is the language of the Quran, Islamic religious teachings, and famous writers and poets. Modern Standard Arabic (MSA), also known as “Al-fus?ha”, is a dialect of Arabic taught in educational institutions and utilized in most media broadcasts, official speeches, and the majority of written materials in the Arab world, including

literature. Colloquial Arabic (or al-ammiyya) is the variant of Arabic employed in daily conversations [58] [59].

Arabic is a member of the Semitic language family. It is characterized by a restricted vocalic system and a complex consonantal system, with the exception of several difficulties in representing short and long vowels [47]. The Arabic alphabet comprises 28 letters, all of which denote consonants. Modern Standard Arabic comprises six vowels and two diphthongs. The six vowels are /a/, /i/, /u/, /aa/, /ii/, and /uu/, with the first three being short vowels and the later three representing their longer versions. Conversely, the two diphthongs are /ae/ and /ao/. The duration of vowel sounds is phonemic in Arabic, distinguishing it significantly from languages like English or Japanese. Each short Arabic vowel is phonetically identical to its long equivalent, with the sole exception being its duration [58].

Table 2.2: List of Arabic letters and their IPA symbol (International Phonetic Alphabet) counterpart

Arabic letter	ء	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ	ف	ق	ك	ل	م	ن	ه	و	ي
IPA symbol	ʔ	b	t	θ	dʒ	ħ	x	d	ð	r	z	s	ʃ	sˤ	tˤ	dˤ	ðˤ	ʕ	ɣ	f	q	k	L	m	n	H	w	j

Table 2.3:Arabic consonants

			Bilabial	Labio-dental	Inter-dental	Alveco-dental	Alveolar	Palatal	Velar	Uvular	Pharyngeal	Glottal
Stop	Voiced	Emphatic				ض /dˤ/						
		Non-Emphatic	ب /b/			د /d/		ج /ʒ/				
	Unvoiced	Emphatic				ط /tˤ/						
		Non-Emphatic				ت /t/			ك /k/	ق /q/		ء /ʔ/
Fricative	Voiced	Emphatic			ظ /ðˤ/							
		Non-Emphatic			ذ /ð/	ز /z/				غ /ɣ/	ع /ʕ/	
	Unvoiced	Emphatic				ص /sˤ/						
		Non-Emphatic		ف /f/	ث /ɵ/	س /s/		ش /ʃ/		خ /x/	ح /ħ/	هـ /h/
Nasal	Voiced	Non-Emphatic	م /m/				ن /n/					
Liquid	Voiced	Non-Emphatic					ر ل /l/r/					
		Emphatic					ل /l/					
Semivowels	Voiced	Non-Emphatic	و /w/					ي /j/				

After [60] [58]

Arabic pronunciation presents challenges due to the presence of consonants that are absent in other languages. This creates confusion for the beginner learners of Arabic, such as the confusion between [ð and ðˤ], [t and dˤ], [d and tˤ], [s and sˤ], and [h and ħ] [47] [59].

2.10.2 Traditional methods for Arabic assessment

Traditional methods for Arabic assessment relate to scoring and confidence measures. Pronunciation scoring gives learners insights regarding their pronunciation precision and fluency through scores. The score is generated by either the regression or classification methodology. In the regression approach, a collection of characteristics is aggregated to get the final score. In contrast, in the classification approach, a classifier assigns the input speech to categories representing human grades.

Similarly to Khan et al. (2013) [61], Bahi and Necibi (2020) [62] employed a Hidden Markov Model-based speech recognizer trained on Mel-frequency cepstral coefficients (MFCCs). The article proposes the Fuzzy logic-based System for Pronunciation Assessment (FuSPA) to improve existing thresholding methods, address rating disparities, make teachers more comfortable with automatic scores, and motivate learners. Two scores were combined using fuzzy rules: Time Duration Score (TDS) and Global Log-Likelihood (GLL). Global average Log-Likelihood was introduced to normalize the effect of the word length.

HAFSS is the first mispronunciation diagnosis system for Arabic (Abdou et al., 2006) [63]; it is a language learning program designed to instruct proper Quran recitation and employs a voice recognizer to identify and diagnose mispronunciations. HAFSS employs a GMM-HMM (Gaussian Mixture Models-Hidden Markov Model) model to identify faults in learners' recitations; a confidence score is calculated using the likelihood ratio. The writers developed a database with 663 rules about pronunciation problems in the recitation of the Holy Quran. They employed the Correct Judgment (CJ) to assess the performance of the HAFSS system. Later, they attempt to improve the confidence score by utilizing articulation features. In 2014 [64], the researchers addressed speech segmentation in the Hafss system at the phonemic level.

Al Hindi et al. (2014) [65] used the Goodness of Pronunciation (GOP) score to detect pronunciation errors in non-native Arabic speakers at the phoneme level, focusing on five difficult phonemes: Tha'a (/θ/ث), a'a (/ح ħ/), Sad (/ş/ص), Dad (/ď/ض), and Dha'a (/đ/ظ). Maqsood et al. (2016) [66] trained an SVM classifier for each phoneme using acoustic phonetic features (APF). The proposed system outperformed the GOP-based classifier for Arabic mispronunciation.

2.10.3 Arabic dispronunciation detection with DL Models

Numerous studies have shown the effectiveness of DL models to deal with the specific problems presented by Arabic phonetics. DNNs were first used to replace older methods, as in [59]. After that, CNNs were investigated for their power in feature extraction and classification in several studies using different approaches.

Al-Marri et al. (2018) [59] introduced a DNN-HMM acoustic model to enhance HAFSS, substituting the GMM-HMM to improve recognition efficacy. It exceeded the GMM-HMM model for the ten assessed errors, reducing the insertion issue by 2.59% and improving the confusion of phonemes /dˤ/, /ðˤ/, and /t/ by 15.09%, 17.28%, and 3.16% for substitutions, respectively.

Considering the 28 Arabic phonemes, the study of Nazir et al. (2019) [52] used 11,164 samples from a private dataset to propose two methods for mispronunciation detection that surpassed SOTA techniques. In the first technique, a deep convolutional neural network (CNN) extracted the features of 400 Pakistani speakers from different layers. Subsequently, the classification algorithms (KNN, SVM, and NN (Neural Network)) identified mispronunciations in the extracted features—the second method utilized transfer learning as an end-to-end method MDD task.

In the same way, Akhtar et al. (2020) [40] introduced a CNN feature-based model for detecting pronunciation errors in Arabic words by extracting features from layers 6, 7, and 8 of the pre-trained AlexNet model. They utilized them to train three machine learning classifiers: KNN, SVM, and Random Forest (RF). The experimental results were derived from a private dataset; utterances were gathered from non-native Arabic learners reciting Quranic verses. The findings indicated that the proposed method attained average accuracies of 73.67%, 85%, and 93.20% with handcrafted features and transfer learning.

In [67], The authors developed a CNN model to classify Arabic short vowels using a new audio dataset. The model was built from scratch, optimized, and fine-tuned over multiple iterations to achieve high classification accuracy.

Algabri et al. (2022) [57] developed an Arabic mispronunciation detection method using multi-label object detection, overcoming the scarcity of Arabic CAPT-dedicated datasets by building a private dataset and adding artificial errors.

Research indicates that Long Short-Term Memory (LSTM) networks effectively detect mispronunciations in Arabic. The capacity of LSTM to capture temporal dependencies in

speech data makes it highly effective in differentiating between similar phonemes and identifying articulation issues. The study in [68] identifies the class of each Arabic alphabet letter using an audio dataset containing 4872 audio files. A binary classification method is proposed to detect the correct pronunciation of the alphabet. Deep convolutional neural networks, AlexNet with transfer learning, and Bi-LSTM are used for classification. However, the study focuses on isolated Arabic alphabet letters. Ahmed et al. (2023) [69] proposed a classification approach using long short-term memory (LSTM) architecture, focusing on Mel frequency cepstral coefficients (MFCC) as discriminative features. The method significantly improved gender recognition and pronunciation error detection with an accuracy of about 81.52%. The paper [70] uses innovative methods to detect mispronunciations in Quranic recitation. Key methods include Mel-Frequency Cepstral Coefficients (MFCC) for audio signal processing, Long Short-Term Memory (LSTM) neural networks for time series data, and the QDAT dataset for comparing the performance of the proposed LSTM model against traditional machine learning algorithms. The results show high accuracy rates of 96%, 95%, and 96% for the three Tajweed rules, demonstrating the effectiveness of deep learning techniques in detecting mispronunciations in Quranic recitation.

2.11 Chapter Summary

This chapter demonstrates that deep learning models have revolutionized mispronunciation diagnosis by improving accuracy, adaptability, and automation. Before the introduction of DL models to this field, CAPT systems mostly used rule-based procedures and statistical methods like Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMM). Despite good phonetic pattern matching, these systems required manual feature engineering. They also struggled to detect pronunciation errors, especially when learners had distinctive speech differences.

Conversely, deep learning algorithms like CNNs, RNNs, and LSTM models may learn complicated pronunciation patterns from vast data sets. Transformer-based systems like wav2vec 2.0 use self-attention to capture long-range speech dependencies, boosting mispronunciation detection. End-to-end deep learning models learn directly from raw audio to diagnose mispronunciations without user intervention, improving scalability and flexibility to different languages and accents.

We believe that generative models can improve mispronunciation detection by modeling complex distributions, improving feature extraction, and providing realistic data. They can also mimic language learners' common speech errors to balance datasets. Thus, they may deliver real-time feedback tailored to each learner and improve pronunciation training to increase engagement.

3 Deep Learning and Generative Modeling

3.1 Introduction

Generative models have attracted considerable interest in recent years. They represent a robust category of machine learning algorithms designed to generate novel samples that emulate a designated dataset. This capability enables them to serve as versatile tools across multiple domains, such as content generation, data augmentation, and anomaly detection. They can produce realistic visuals, music, and text, enhance model efficacy under challenging situations, and detect anomalies or fraudulent actions by comprehending "normal" data. In focus of our review of recent developments in deep learning for pronunciation assessment and the potential advantages of generative systems in this field, we propose this chapter to analyze these models by contrasting them with discriminative ones, with a focus on the variational autoencoder (VAE), which we intend to utilize in the subsequent chapter for our proposal.

3.2 Discriminative Vs. Generative models

Machine learning can be divided into two schools of thought: discriminative and generative (and/or informative) learning [71]. To understand the difference between the two, we first need to define how they work, their objectives and their learning processes in the following sections.

3.2.1 Discriminative models

Conditional models, or discriminant models, often linked to supervised learning, are a type of machine learning applied to static classification. They are called conditional models because they discern the boundaries between classes or labels within a data set, thus differentiating one class from another [72]. Discriminative methods attempt to determine correspondences between inputs and outputs for classification and regression tasks, eschewing any modeling of data distributions.

Mathematically, training a discriminative classifier requires calculating a function $\boldsymbol{f}: \boldsymbol{X} \rightarrow \boldsymbol{Y}$ or a probability $\boldsymbol{P(Y|X)}$ assuming a specific functional form for the probability, such that $\boldsymbol{P(Y|X)}$, then estimating its parameters by using the training data. Thus, the goal is just optimizing the learning of a function for mapping the inputs to the target outputs without

creating a generator capable of modeling all variables within the system, prioritizing the enhancement performance [71].

Machine learning techniques such as gaussian process models, support vector machines, boosting algorithms, and traditional neural networks have enabled robust discriminative classification and regression to be carried out successfully in many domains. Despite this, deep neural networks have become fundamental in classification and regression tasks due to their capacity to approximate complex functions and learn from large datasets.

3.2.2 Generative models

In contrast to discriminative models that identify class boundaries, generative or informative approaches are primarily used in unsupervised learning, where the model learns from unlabeled data without explicit guidance. They comprehend the underlying data distribution, thus enabling the generation of new instances statistically similar to those in the training set. Therefore, a generative model primarily aims to understand and estimate the probability distribution P(x) of observed data points x. It can identify patterns such as colors, textures, shapes, and other visual attributes, categorize a dataset, and, as a result, improve various applications [72].

Training generative classifiers requires determining a function $\boldsymbol{f}: \boldsymbol{X} \rightarrow \boldsymbol{Y}$ **,** or the probability $\boldsymbol{P(Y|X)}$:

- Assume a certain functional form for the probabilities, such as $\boldsymbol{P(Y)}$ and $\boldsymbol{P(X|Y)}$.
- Utilizing training data, we estimate the parameters of $\boldsymbol{P(X|Y)}$ and $\boldsymbol{P(Y)}$.
- Utilize Bayes' theorem to compute the posterior probability $\boldsymbol{P(Y|X)}$**.**

Generative models are used extensively in diverse sectors, such as picture and video synthesis, text generation, speech and audio synthesis, healthcare, medication development, data augmentation, and synthetic data generation. The early ones, such as Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs), were widely used in speech recognition and clustering but were limited in terms of scalability and expressiveness for high-dimensional data. These limitations were overcome with the advent of autoregressive models such as PixelCNN and WaveNet, which enabled sequential generation. Hence, it proves effective in image synthesis and audio generation.

Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) have recently received considerable scholarly attention. The first was introduced in the early 2010s, providing a significant advance in latent variable modeling by approximating complex

probability distributions through variational inference. The second was proposed by Ian Goodfellow et al. in 2014 and represents one of the most influential advances in generative modeling. They employ two competing networks—the generator and the discriminator—to generate remarkable results in realistic images and videos.

Other models have emerged, such as Denoising Diffusion Probabilistic Models (DDPM) and Transformers. Language models like GPT and BERT have become powerful frameworks for sequence-based generative tasks. DDPM generates highly realistic images by simulating diffusion processes, producing high-resolution images that compete with GANs in quality. These models have been extended to multimodal domains and demonstrate coherent text generation.

3.2.3 Comparison between discriminative and generative models

Discriminative methods frequently employ parametric models, which lack the sophisticated probabilistic proposals of priors, structure, and uncertainty that are advantageous in generative contexts. Instead, alternative concepts of penalty functions, regularization, and kernels are employed [71] [72].

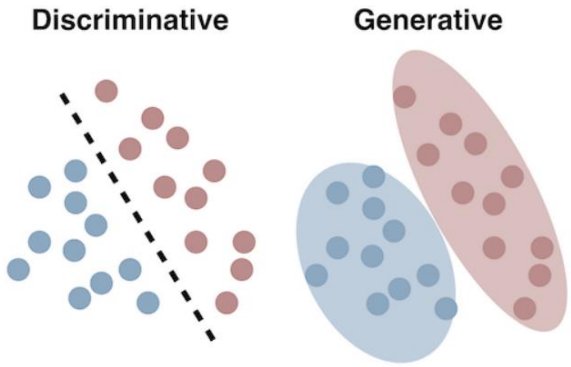


Figure 3.1: Discriminative Vs. Generative modeling

Moreover, discriminative techniques' primary focus is on learning classifiers and mappings, which complicates the integration of flexible modeling tools and generative prior knowledge regarding variable space. Consequently, discriminative approaches can seem like black boxes, wherein the relationships between variables may not be as explicit or visualizable as generative models.

Table 3.1: Comparison between Discriminative and Generative models

Aspect	Generative models	Discriminative models
Purpose	Data distribution modeling	Conditional probability modeling of labels given data
Use case	Data generation, denoising, unsupervised learning	Classification, Supervised learning
Example of architectures	Variational autoencoders (VAEs), Generative Adversarial Networks (GANs)	Logistic Regression, Support Vector Machines, Deep Neural, Networks
Training goals	Optimize likelihood of observed data, Capture the structure and distribution of the data	Learn decision boundaries, differentiate between classes.
Applications	Image generation, Sound generation, Inpainting…etc.	Text classification, object detection, etc.

Generally, generative models are suitable for generating new data or handling unlabeled or mislabeled datasets. Discriminative models are effective when the primary focus is on classification tasks, utilizing a well-annotated dataset, and where predicted accuracy is the highest priority.

3.3 Deep Learning

3.3.1 Big data and transition from shallow ML techniques to Deep Learning

In general, large data sets are essential for understanding the constitution of knowledge, the absorption of information, the organization of reality, and research processes. They offer a new way of affirming quantitative science and objective methods in humanistic disciplines. However, they have significant shortcomings, such as the fact that they do not always indicate quality data, lose their meaning when removed from the content, and are not ethically valid due to their accessibility. These data represent w²hat is known as Big Data, mainly due to their volume, variety, and velocity. Therefore, conventional machine learning algorithms have struggled to extract advanced features and model the data representations to tackle intuitive problems that can only be solved using deep learning techniques.

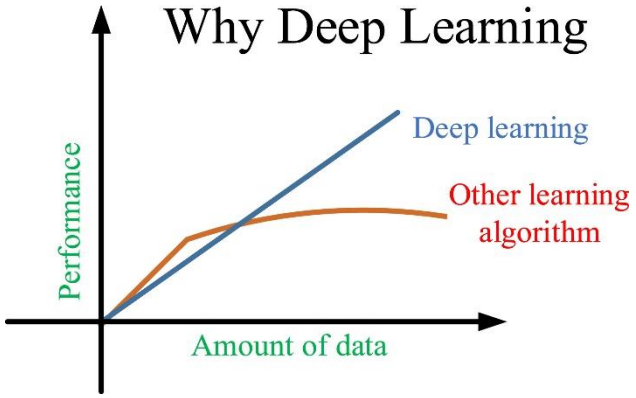


Figure 3.2: Deep learning Vs. Traditional learning techniques after [73]

3.3.2 Definition of Deep Learning

Thanks to big data, deep learning can solve problems that cannot be described formally by enabling computers to learn from experience and understand the world through a hierarchy of concepts. Each of these is defined by its relationship to more straightforward concepts, avoiding the need for human operators to specify all the necessary knowledge [71]. The hierarchy allows computers to learn complex concepts by building them from simpler concepts, creating a deep, multi-layered graph known as AI deep learning [74].

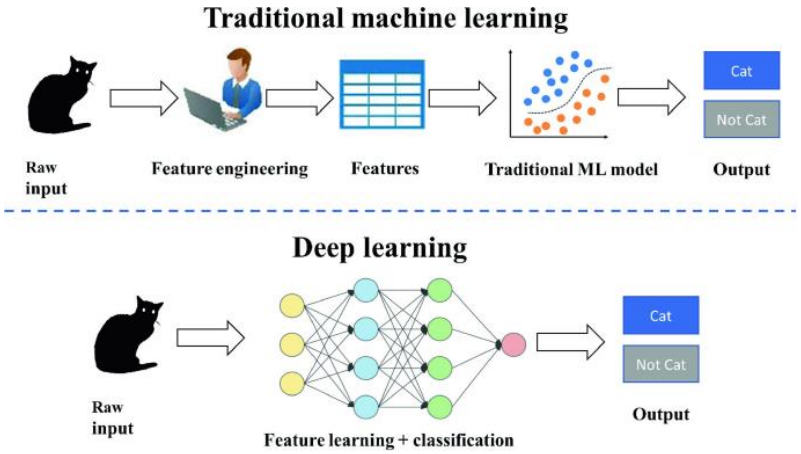


Figure 3.3: The difference between shallow ML techniques and Deep Learning

Deep learning is a pillar in machine learning and a major asset for artificial intelligence. It uses artificial neural network models to extract new knowledge from large datasets. Deep learning leverages cognitive science to mimic how the human brain works and thus enables machines to identify patterns and make decisions based on complex data. The structure of the neural

networks is based on layers, including an input layer, one or more hidden layers, and an output layer, each layer converts the data it receives to an abstract representation. The number of layers in these networks specifies the depth, varying from a few to hundreds or even thousands.

3.3.3 The functioning of a perceptron

To understand how a deep learning model works and the relationships between its layers, it is essential to comprehend the functioning of an artificial neuron, which is the primary unit of a DL graph.

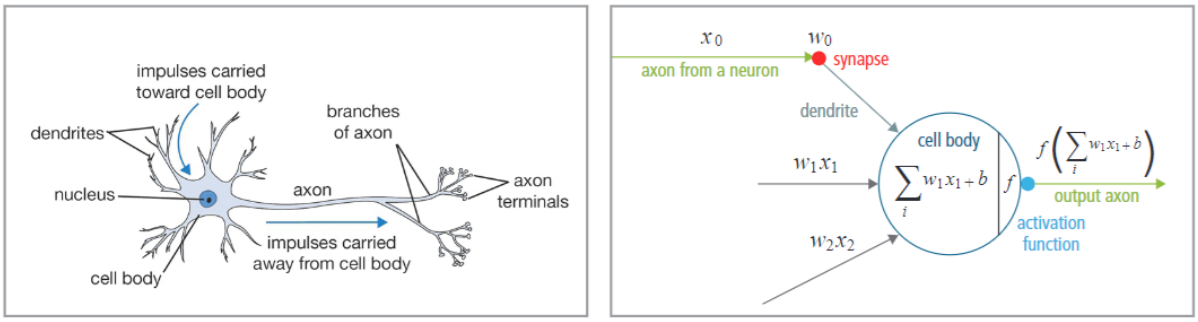


Figure 3.4: Comparison between biological neuron and artificial neuron (a perceptron) after [75]

Artificial neuron functioning resembles a biological neuron, which is the fundamental computational unit of the human brain. A neuron is seen as receiving input signals from its dendrites and producing output signals along its axon. The axon branches and connects via synapses to the dendrites of other neurons. The neuron is activated only when the combination of input signals reaches a specific threshold condition among its input dendrites, and activation is communicated to subsequent neurons at this point [75].

In the computational model, artificial neurons receive input signals from raw data or previous layers, each associated with a weight that determines its significance in the computation. They compute a weighted sum of these inputs, multiplying each input by its corresponding weight and adding these values together. The formula can be expressed as:

$$z = \sum(w_i x_i) + b$$

After calculating the weighted sum, the neuron applies an activation function to determine whether it should "fire" (produce an output), introducing non-linearity into the model. Common activation functions include Sigmoid, ReLU, and Tanh. The output from the activation function is sent to neurons in subsequent layers through synapses. Neurons learn by

adjusting their weights based on network performance feedback. Backpropagation is used during training to minimize prediction errors by updating weights [75].

3.3.4 Activation functions, learning process, and backpropagation

Neural networks are mainly used for the prediction and classification tasks using data fed into the network. This data can be complex and highly dimensional, such as images, video, audio, speech, text, etc. For this purpose, neural networks need non-linear functions to establish non-linear correspondences between inputs and outputs. This inherently renders the network dynamic, allowing it to model complex and complicated patterns from the data and represent random, non-linear functional correspondences between input and output. Without non-linearity, a neural network would behave essentially like a linear regression model, regardless of its number of layers [76].

The neural network has neurons that work in correspondence with weight, bias, and their respective activation function. In a neural network, we would update the weights and biases of the neurons on the basis of the error at the output. This process is known as back-propagation. Activation functions make the back-propagation possible since the gradients are supplied along with the error to update the weights and biases. On the other hand, an activation function must be differentiable to allow the implementation of the backpropagation so that errors or losses can be calculated during learning, which optimizes the calculation of weights using gradient descent or other optimization techniques [76].

3.3.5 Variants of Activation Functions

Many factors, including the number of hidden layers in a network, learning methods, and the definition of hyperparameters, closely influence the choice of activation function to achieve optimal performance. Selecting the appropriate activation function for a neural network task can be time-consuming and necessitate extensive research and experimentation, as there is no set rule for this process. However, the context, or the task at hand, determines the choice of activation function. Various activation functions offer advantages and disadvantages, necessitating experimentation during implementation to identify the most effective solution for the problem at hand. For example, the ReLU function is widely used in hidden layers and generally gives better results; however, it is never used for the outer layer, unlike the sigmoid and tanh functions, which are unsuitable for hidden layers because of their low slopes [76] [76].

Several activation functions can be used depending on the learning objective, namely 1. binary staircase function 2. linear function 3. sigmoid 4. Tanh 5. ReLU 6. ReLU with leakage

7. Parametric ReLU 8. Linear exponential unit 9. Swish 10. SoftMax [76]. However, in the following chapters, we present the two activation functions used in our proposals: ReLU and Sigmoid.

3.3.5.1 ReLU function in hidden layers

The rectified linear unit is the most commonly used activation function. Generally implemented in the hidden layers of neural networks, it is a non-linear activation and less costly calculation than the tanh and sigmoid functions. It involves simpler mathematical operations while simultaneously, only a few neurons are activated at any given moment, making the network sparse. Consequently, the ReLU learns faster than the sigmoid and tanh functions.

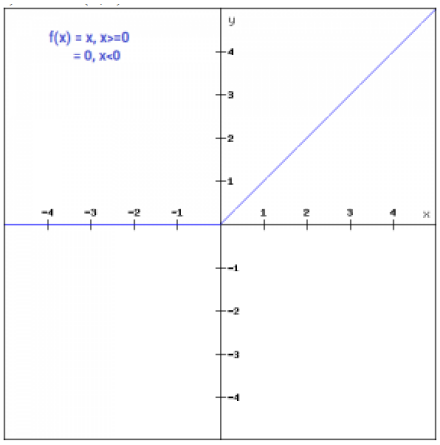


Figure 3.5: Relu Activation Function plot

Equation:

$$f(x) = \max(0, x)$$

f(x) = max (0, x) It gives an output x if x is positive and 0 otherwise.

Value Range: [0, inf)

3.3.5.2 Sigmoid function in output layer

In neural networks, the sigmoid function is characterized by its non-linearity. It is often used in the output layer for binary classification tasks, where it converts raw output scores into probabilities whose sum equals one. It is also fundamental in logistic regression, transforming any real-valued input into a value between 0 and 1. Thus, he sigmoid is very useful for models in which the output can be interpreted as a probability.

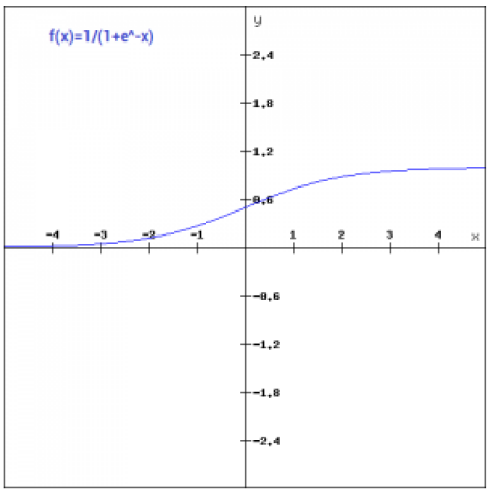


Figure 3.6: Sigmoid Activation Function plot

Equation:

$$f(x) = \frac{1}{1+e^{-x}}$$

Value Range: 0 to 1

3.4 Deep Neural Networks as Discriminative Models

3.4.1 Deep Feedforward

Deep Feed-Forward Networks (DFNs), often called Multi-Layer Perceptrons (MLPs), are one of the fundamental designs of deep learning. They are mostly utilized for supervised learning tasks. In accordance with their name, they are organized in the form of several layers completely connected to one another. In such networks, information always propagates forward, starting from the input toward the output, and no feedback is allowed. Sometimes, the feedbacks are allowed to handle the temporal progression, in this case, the resulting networks are referred to as recurrent neural networks. Deep forward networks are the most basic artificial neural network type, and they are used for a wide range of applications, including image and speech recognition, language processing, and predictive analysis [74] [77].

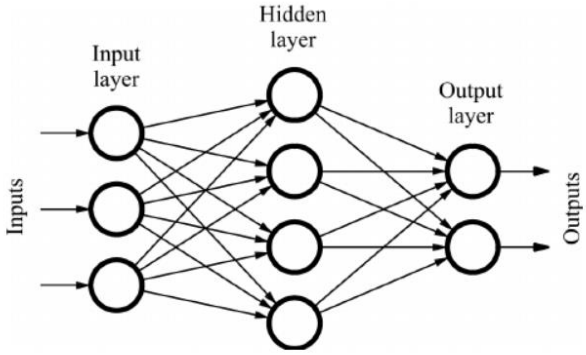


Figure 3.7: Feedforward neural network architecture

For an input vector x and an output prediction y, each layer executes a sequence of linear transformations succeeded by a non-linear activation function. Mathematically, with a network including L layers, the output of each layer l can be defined as:

$$h^l = \sigma\left(W^{(l)}h^{(l-1)} + b^{(l)}\right)$$

Where:

- $h^{(0)} = x$ (the input),
- $W^{(l)}$ is the weight matrix for layer l,
- $b^{(l)}$ is the bias vector for layer l,
- σ is an activation function (e.g., ReLU, sigmoid).

DFCs are simple and versatile to implement and can be adapted to many tasks if the activation functions are correctly defined. However, they are unable to capture the mechanisms of sequential or spatial models. Another disadvantage of feedforward architecture is that, if they are too deep, they can become computationally expensive and require substantial resources for learning.

3.4.2 Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNNs) are artificial neural networks engineered for analyzing sequential data, including time series, language, and speech. Recurrent Neural Networks (RNNs) are distinguished by their capacity to retain memory through feedback connections, enabling information from prior time steps to affect the current output. RNNs are particularly advantageous for jobs where context or sequence is significant, including language translation, speech recognition, and video analysis [73] [74].

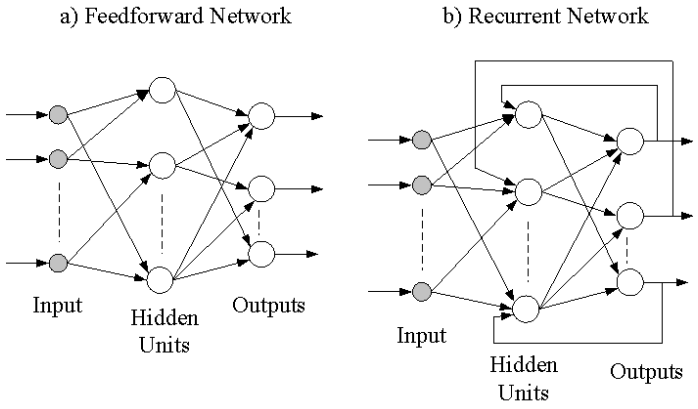


Figure 3.8: Feedforward Neural Network Vs. Recurrent Neural Network

Recurrent Neural Networks (RNNs) utilize feedback connections to handle inputs of varying lengths by preserving hidden states that develop over time. They comprise an input layer, a recurrent hidden layer, and an output layer. The input layer processes each sequence element sequentially, whereas the hidden layer retains information from prior inputs. This enables RNNs to preserve information over several time steps, rendering them essential for context-dependent tasks [78].

Recurrent Update: The hidden state is computed as:

$$h_t = f(W_x x_t + W_h W_{t-1} + b)$$

Where:

- x_t is the input at time t,
- W_x and W_h are weight matrices for the input and hidden state, respectively,
- b is a bias term,
- f is an activation function, typically the hyperbolic tangent (tanh) or ReLU.

Output Calculation: Depending on the application, the output y_t can be computed at each time step or only after the final time step. If calculated at each step, it is typically computed as:

$$y_t = g(W_y h_t + c)$$

Where:

- W_y is the weight matrix for the output,
- c is a bias term,

- g is the activation function, that may be Softmax for classification tasks.

Recurrent neural networks are fundamental architectures for modeling sequences. Indeed, with their memory and temporal awareness, deep learning applications for time-series are increasingly easy to deploy. Despite their limitations, such as the vanishing of gradients, architectures such as LSTMs have demonstrated their effectiveness in managing long-term dependencies. They remain essential in areas such as NLP, speech recognition, and time series analysis, although more recent models, such as transformers, are gradually being favored for tasks involving extended sequences.

3.4.3 Attention mechanism

The attention mechanism is a revolutionary component in deep learning history; it deals particularly with transformers, it was presented by Vaswani et al. in 2017, in the seminal paper entitled "Attention is all you need". Initially developed for sequence-to-sequence translation, the attention mechanism has since been adapted for diverse tasks in natural language processing, computer vision, and speech processing. It allows the model to dynamically weight input data according to its relevance to the task, enabling it to 'pay attention' to important features of each input for each output step [79] [80].

Three components support the implementation of the attention mechanism which are: **query** (Q), **key** (K), and **value** (V) vectors. In the context of language processing, these components work as follows:

1. **Query** (Q): Represents what the model is currently looking for.
2. **Key** (K): Represents different parts of the input to evaluate relevance.
3. **Value** (V): Represents the actual information to be used for the prediction.

The mechanism implements the attention as described by the following formula:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Where:

- QK^T: Measures the similarity between queries and keys. Higher similarity implies higher attention weight.
- $\sqrt{d_k}$: Stabilizes gradients when d_k, the dimensionality of keys, is large.

- Softmax: Converts similarity scores to probabilities, summing to 1 across each sequence.

The attention mechanism allows the handling of long-range dependencies and task parallelization, it also permits dealing with various data types and tasks. However, it involves high computational costs and memory usage. To address such limitations, researchers are developing techniques like sparse and linearized attention to reduce complexity while preserving performance.

3.4.4 Convolutional Neural Networks (CNN)

3.4.4.1 Definition of Convolutional Neural Networks

Convolutional Neural Networks (CNNs), conceived by LeCun et al. (1989) [81], have a hierarchical architecture similar to that of the visual cortex, as depicted in the adjacent figure, where fundamental features are extracted in the initial layers, and progressively complex features are formed in the subsequent layers. The local connection of neurons in the visual cortex and convolutional neural networks facilitates an efficient representation of visual stimuli. Translation invariance is achieved through pooling layers in convolutional neural networks, which condense local features. Convolutional Neural Networks (CNNs) replicate the human visual system by utilizing several filter maps in each convolutional layer and incorporating non-linearity through activation functions like ReLU, which are applied after each convolution [74] [82].

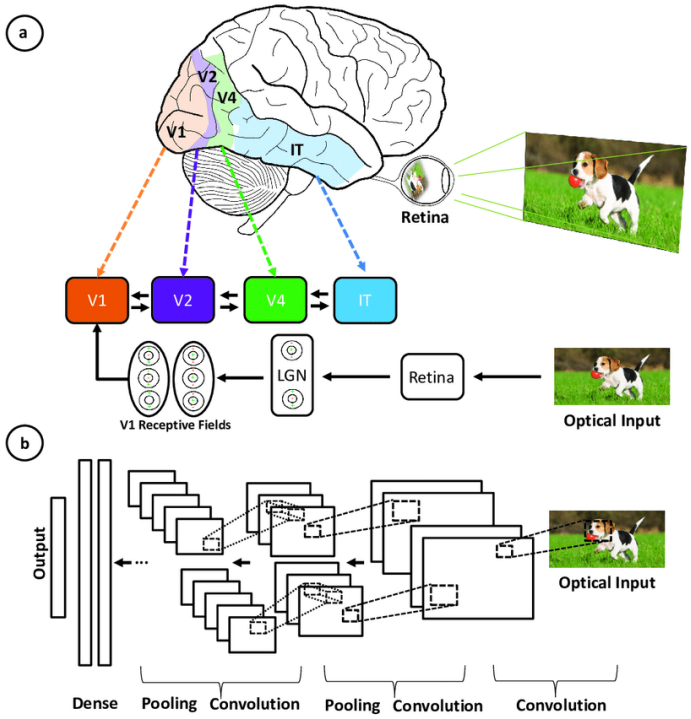


Figure 3.9: The inspiration that contributed to CNN after [82].

3.4.4.2 Pooling layers

Besides the aforementioned convolutional layers, convolutional neural networks also incorporate pooling layers, which minimize the output data from the convolutional layer. Max-pooling simulates the function of cells in the Lateral Geniculate Nucleus (LGN) by dividing the input image into non-overlapping rectangles and providing the maximum value for each sub-region. After the gathering of convolved features and the specification of the area size, pooled convolved features are derived by partitioning them into non-overlapping n × n regions. Only the maximum is chosen as a feature activation across these regions to derive pooled features [82].

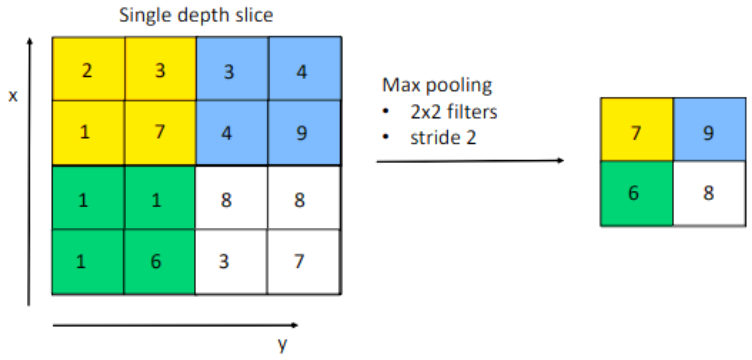


Figure 3.10: Max-pooling on a 4 × 4 depth slice

Max-pooling is advantageous for two primary reasons: it diminishes computational requirements for subsequent layers by a factor of n^2 by discarding non-maximal values, thus reducing the number of parameters to be learned in later layers, and it offers a degree of spatial invariance, ensuring that the same (pooled) feature remains active despite minor translations of the image.

3.4.4.3 Forward Pass of CNNs

In a CNN, the forward pass progresses from the first convolutional and pooling layers to the fully connected layers, terminating in the production of an output, such as a probability distribution for classification tasks [74]. The output of each layer can be mathematically represented as follows:

$$Output = Activation\left(Pooling\left(Convolution(Input)\right)\right)$$

For a specific layer l, the output is obtained as follows:

$$O_l = \sigma\left(P_l\left(C_l(O_{l-1})\right)\right)$$

Where O_{l-1} is the output from the previous layer,C_l is the convolution operation, P_l is the pooling operation, and σ is the activation function.

3.4.4.4 Backpropagation in CNNs

Backpropagation is a method used in CNN training. This method involves calculating the gradients of the loss function with respect to each weight, which updates the weights in each layer. Therefore, backpropagation must also take into account the sharing of weights by adding gradients between points in the feature map where the same filter is applied. This is necessary because convolutional layers share weights across spatial dimensions [74].

Table 3.2: Summary Table: Key CNN Architectures and Contributions

Architecture	Year	Key Contributions	Reference
LeNet	1989	Introduced convolutional layers and pooling for digit recognition	LeCun et al. (1989) [83]
AlexNet	2012	Deep CNN with ReLU, dropout, and image augmentation	Krizhevsky et al. (2012) [84]
VGGNet	2014	Standardized small convolution filters (3x3)	Simonyan & Zisserman (2014) [85]
ResNet	2015	Introduced residual connections for a very deep network	

3.5 Deep Neural Networks as Generative Models

3.5.1 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) are machine learning models that, starting from a given set of samples, produce new samples that are close to the initial ones. GANs were first introduced by Ian Goodfellow in 2014 [86], consisting of two main components: the generator (G) and the discriminator (D). Starting from a random noise, the generator creates new data samples; the new samples are expected to be similar to real samples. Meanwhile, the discriminator seeks to untangle true from fake samples; given that the discriminator was previously trained on real samples. The two networks are trained in the logic of a minimax game, with the generator looking to maximize the probability that the discriminator classifies its output as real and the discriminator seeking to minimize its classification error. This competitive vision encourages the two networks to improve their outcomes: the generator will improve the production of realistic samples while the discriminator will improve its classification capabilities [87]. GANs are gaining much popularity for tasks such as image synthesis, video generation, and style transfer because they can create realistic, high-quality data samples.

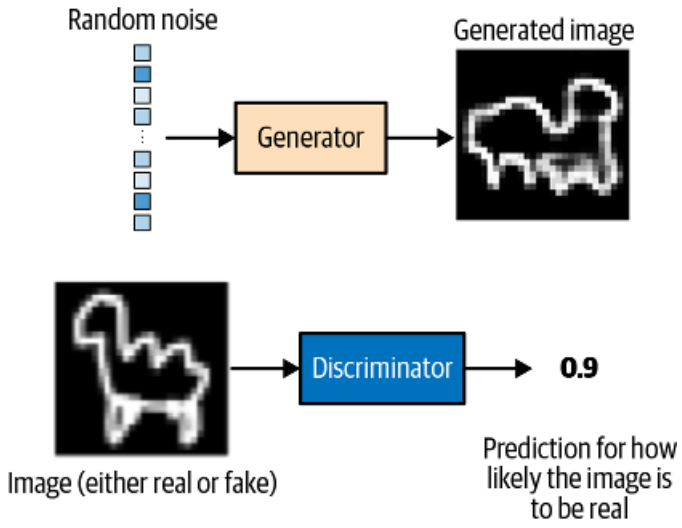


Figure 3.11: The role of the generator and the discriminator in GAN after [72]

$$G\min D\max V(D,G) = \mathbb{E}_{x\sim p_{data(x)}}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \qquad (3.10)$$

Where:

- $D(x)$ represents the discriminator's probability estimate that xxx is real.
- $G(z)$represents the generator's output when given input noise z, typically sampled from a normal or uniform distribution.
- p_{data} is the true data distribution, and p_z is the noise distribution used by the generator.

Generative adversarial networks are highly effective and flexible in deep learning. They generate high-quality data for visual tasks and provide photorealistic outputs. However, they have several drawbacks, such as training instability, mode collapse, and substantial tuning, all of which can be time-consuming and computationally costly. These trade-offs demonstrate the potential and challenges associated with employing GANs in complicated generative modeling tasks, thus helping to balance the complexity of training and the quality of the model.

3.5.2 Variational Autoencoders (VAEs)

Variational Autoencoders (VAEs) are powerful generative models that learn and capture a dataset's basic distribution, allowing for the production of new samples that closely resemble the original. VAEs, which were first described by Kingma and Welling in 2013 [88], use the principles of deep learning and probabilistic modelling to generate structured data, such as

images, audio, or text, by learning a compressed representation in a continuous latent space. This latent space is built so that each data point corresponds to a latent variable with a Gaussian probability distribution, which guides the model's production of new samples that follow the learnt distribution [89].

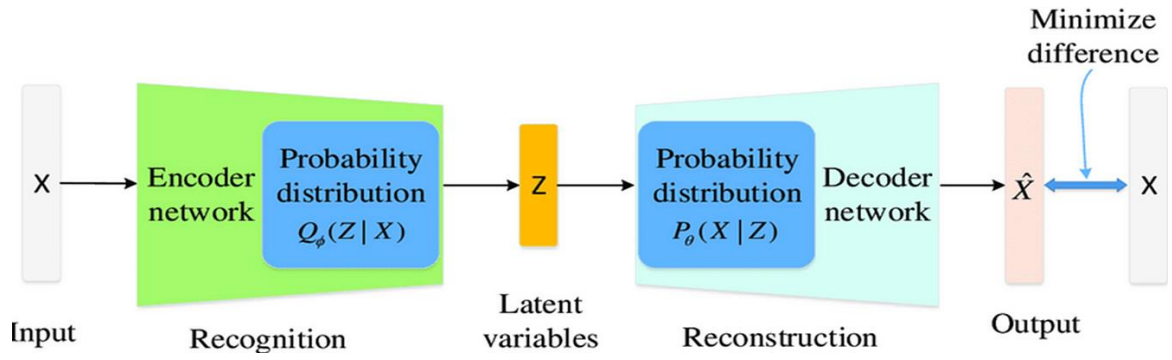


Figure 3.12: The structure of a VAE

Variational autoencoders (VAEs) fundamentally aim to encode data into a latent space that retains the most significant properties, facilitating precise reconstruction and the production of novel data points. Through training on a dataset, the VAE can transform intricate, high-dimensional data into a more straightforward latent representation, encapsulating essential patterns and relationships. This approach is essential for applications such as data augmentation and anomaly detection, where the capacity of VAEs to capture complex data structures enables them to discern deviations or produce realistic data variations.

Variational autoencoders (VAEs) employ variational inference to estimate complicated posterior distributions. They try to maximize the evidence lower bound (ELBO) of data log-likelihood while minimizing reconstruction error and guaranteeing that latent variables conform to a Gaussian prior distribution.

$$\mathcal{L}_{VAE} = \mathbb{E}_{q_\phi(z|x)}[log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z))$$

Where:

- $q_\phi(z|x)$ is the approximate posterior (the encoder).
- $p_\theta(x|z)$ is the likelihood of reconstructing x given z (the decoder).
- D_{KL} is the Kullback-Leibler divergence, which regularizes $p_\theta(x|z)$ to be close to the prior distribution $p(z)$
- ϕ and θ are the parameters of the decoder and encoder networks, respectively.

The probabilistic nature of VAEs, the fluidity of their latent space, and the stability of their training are the main advantages of these generative models. They acquire the ability to learn a probabilistic mapping of data to the latent space, which enables them to perform tasks that involve uncertainty estimation properly. In addition, VAEs also have a well-defined objective function (ELBO), and they do not require adversarial loss throughout the training process.

3.6 Comparison between generative models

In what follows, we present a comparative table of the different generative models identified in the previous section, in this case, VAE and GAN. The attention mechanism is also considered because it is not intrinsically linked to generative or discriminative modelling; rather, it is a polyvalent tool that can enhance both.

Table 3.3: Comparison between the different generative models

Feature	Attention Mechanism	GANs	VAEs
Architecture Type	Encoder-Decoder or Transformer-based	Adversarial Network (Generator + Discriminator)	Autoencoder with Latent Variable Sampling
Generative Approach	Sequential Generation (Token-by-Token)	Generates data by adversarially learning distribution	Probabilistic Model (learns latent variable distribution)
Focus Mechanism	Attention weights allow selective focusing	No inherent focus mechanism; generates based on noise input	No inherent focus mechanism; samples from learned latent space
Training Objective	Cross-entropy or likelihood maximization	Minimax game: discriminator and generator loss	ELBO (Evidence Lower Bound) to approximate likelihood
Control over Output	High, through learned contextual attention weights	Medium, depends on generator loss convergence	Low-to-medium, but disentanglement can be enforced
Primary Applications	Language generation, image captioning, translation	Image generation, data augmentation, unsupervised learning	Image generation, anomaly detection, structured data sampling
Interpretability	Attention weights offer insight into model focus	Typically low interpretability	Latent variables can be interpretable if disentangled

3.7 Which model for the Anomaly detection task

Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) are proficient approaches for anomaly detection. VAEs are designed to comprehend the latent space of typical data through reconstruction, providing advantages such as resilient reconstruction, probabilistic interpretation, and training stability. Nevertheless, they may provide less distinct outputs, constraining their capacity to discern complex data in anomalies. GANs may identify anomalies by creating samples close to regular data and recognizing inconsistencies when confronted with

abnormal samples. They generate more precise and realistic images and can be utilized for anomaly detection by evaluating the discriminator's error rate. Still, they present training challenges due to mode collapse and the necessity for balance between the generator and discriminator. GANs do not provide an explicit probability output, complicating the interpretation of anomaly scores in terms of likelihood.

Table 3.4: VAE Vs GANS

Method	Best For	Key Advantages	Limitations
VAE	Scenarios needing explicit probability scores, stable training, and low-resolution tasks	Probability interpretation, stable training	Less sharp reconstruction may miss fine-grained details
GAN	High-resolution data and applications needing realistic reconstructions	Sharp, realistic reconstructions, discriminator learning patterns	Training instability lacks explicit probability interpretation

Both methods are widely used, but VAEs are often preferred when the goal is reliable, interpretable anomaly detection. At the same time, GANs are chosen when detail-rich outputs are required and the data and resources support stable training.

3.8 Chapter summary

In this chapter, we have covered the fundamental concepts of deep learning through the discriminative and generative modeling approaches.

Here, we began with discriminative models. These models are designed to classify data by learning the boundaries between classes. In recent years, they have formed the fundamental feed-forward recurrent neural networks (RNNs) and convolutional neural networks (CNNs) architectures of deep learning. They have also been successful in many applications, including image classification, speech recognition, and natural language processing.

We then moved on to generative models, i.e., models that learn the underlying distribution of the data in order to generate new samples from it. The most well-known generative models in the literature, which we have discussed in this chapter, are GANs and VAEs.

The comparative analysis with which we ended the section highlighted the strengths and limitations of the discriminative and generative models we examined. If we had to choose direct applicability and a relative balance between strengths and weaknesses, we would certainly favor the generative models. Especially for our proposal, which consists of pronunciation error detection with anomaly detection, the best model in this case is the variational autoencoder. VAEs resorted to probabilities to model data distributions, and their training is relatively stable, which is essential for effectively identifying anomalies within complex data.

As in our proposal, VAE has a central place, in the following subsections, particular attention is given to the VAE model to explain why it is the most suitable model for our proposal in anomaly detection. Thus, mathematical aspects and the training mechanisms are detailed to highlight its application in anomaly detection.

4 Anomaly Detection for Arabic MDD

4.1 Introduction

After doing an exhaustive analysis of the field of Computer-Assisted Pronunciation Training (CAPT), we were motivated by the difficulties faced by the mispronunciation detection module. Some very interesting suggestions have been brought out in this area. Despite this, scientists have constantly fallen into significant obstacles of mislabeled and imbalanced datasets, making the representation of the broad spectrum of pronunciation errors of non-native speakers very challenging. In fact, CAPT systems based on traditional methods, such as forced alignment and statistical models, are restricted in terms of their flexibility and adaptability to this variability. The literature review presented in the second chapter has shown that these problems have been overcome by introducing deep learning in this field.

In this chapter, we propose a new solution that conceives of pronunciation error detection as an anomaly detection task. This allows the model to identify atypical pronunciations based on a learned distribution of correct pronunciations. Our approach aims to enhance CAPT systems by effectively detecting pronunciation variances without requiring exhaustive labels for each probable inaccuracy.

4.2 Mislabeled and imbalanced datasets issue in MDD

Mislabeled or generally imbalanced datasets can affect the accuracy with which Computer-Assisted Pronunciation Training (CAPT) systems can detect learner errors and provide feedback. Such datasets can limit the model's performance, leading to lower pedagogical effectiveness, and can reduce robustness and generalizability. This problem is challenging in CAPT, as large-scale, high-quality labeled datasets are rare, and manual annotation is extremely time-consuming. Because of these limitations, the degree to which pronunciation accuracy contributes to CAPT is not well defined. In fact, manual annotation of longer sentences or recordings is extremely demanding, introducing potential bias or noise into annotations due to human fatigue and inconsistency. Unfortunately, when training data contains incorrect labels,

the model learns to associate incorrect pronunciations with correct phonetic output, resulting in high rates of false negatives and inefficient feedback for learners [90] [91].

Representation learning may address problems in mislabeled datasets by allowing models to focus on significant, high-level features in the data instead of depending exclusively on erroneous or noisy labels. It aims to acquire resilient data representations that exhibit reduced sensitivity to label noise, hence enhancing model performance even when the dataset quality is compromised.

4.3 Representation and feature learning

"An AI must fundamentally understand the world around us" [92], i.e., AI must acquire the ability to recognize and differentiate between the deeper explanatory elements hidden inside low-level data, which allows for a fundamental comprehension of everything that surrounds us. Unfortunately, the efficacy of machine learning algorithms significantly depends on selecting data representation or features, which is a laborious process that underscores the limitations of existing learning algorithms in extracting and structuring discriminative information. To enhance the capacity and applicability of machine learning, algorithms should exhibit less dependence on feature engineering. This will facilitate the development of innovative applications and advancement in Artificial Intelligence (AI). This is made possible by Representation Learning, which, with regular workshops and conferences, is becoming a field in its own right.

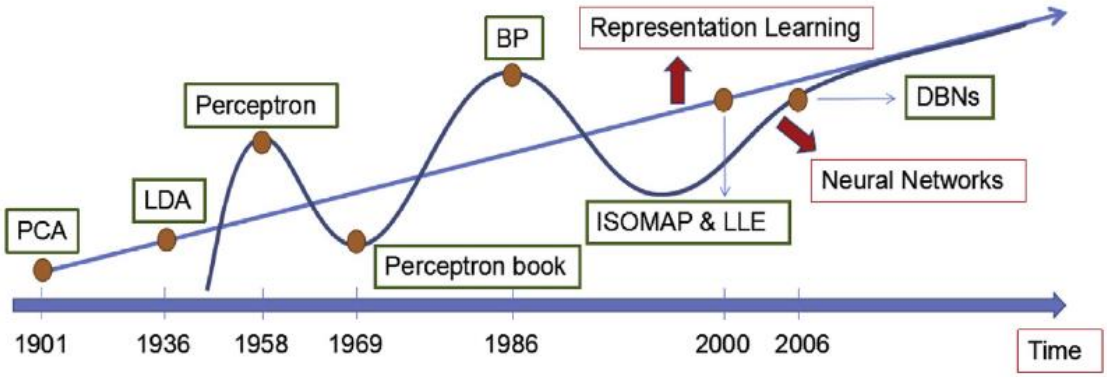


Figure 4.1: The development of data representation learning in DL

after [93]

The main goal of representation learning is to learn data transformations to get relevant data for predictors or classifiers. An efficient representation captures the posterior distribution of

explanatory variables for observed inputs and subsequently integrates various non-linear data transformations to provide more abstract and insightful representations [92] [93] [94] [95].

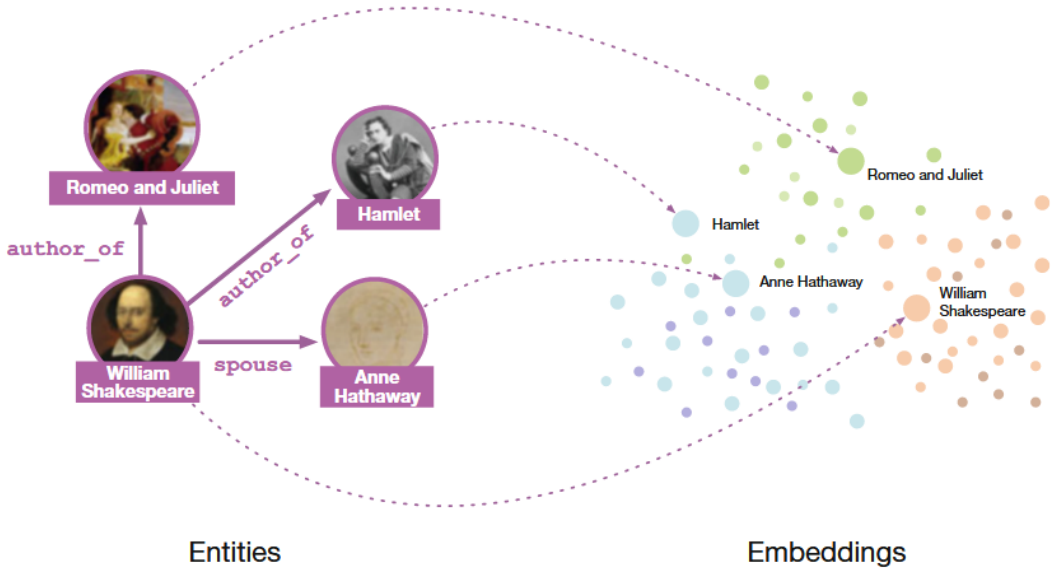


Figure 4.2: Illustration of discovering the underlying structure of data through Representation Learning after [95]

Representation learning has achieved considerable breakthroughs in speech recognition, signal processing, object recognition, statistical language modeling, and domain adaptation, evidenced by numerous achievements in both academia and industry. It diminishes the cutting-edge error rate across diverse benchmarks, illustrating the capabilities of representation learning in numerous domains. Representation learning has also been applied to statistical machine translation, word sense disambiguation, and sentiment analysis. It was also combined with transfer learning, which has the ability to develop algorithms that leverage similarity across diverse tasks in order to identify fundamental features pertinent to each task. Successful applications encompass domain adaptation and multi-task learning [92].

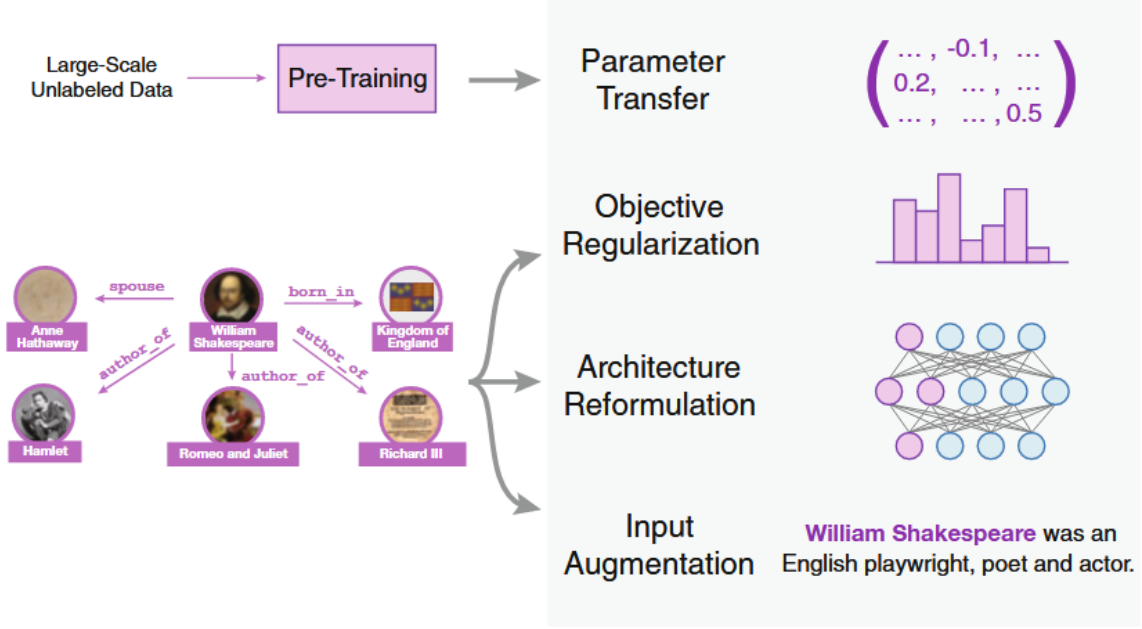


Figure 4.3: An example of using representation learning for different goals in NLP after [95]

We use representation learning to extract abstract features from poorly labeled datasets, thereby reducing reliance on incorrect labels. Contrastive learning enables models to discern similarities and contrasts among data pieces without necessitating precise labels. In the same way, with Self-Supervised Learning (SSL) techniques, representation learning is effective in acquiring representations independently of labels, facilitating fine-tuning on noisy or scarcely labeled datasets. On the other hand, representation learning is essential for anomaly identification, as it identifies outliers within the dataset. Variational Autoencoders and other models can find instances that were wrongly labeled by looking for changes in the learned data distribution.

4.4 Representation learning for an unsupervised MDD (Background)

Recently, representation learning has emerged as a pivotal technique in enhancing the effectiveness of Computer-Assisted Pronunciation Training (CAPT) systems, particularly in the domain of mispronunciation detection. This approach leverages advanced machine learning models to interpret and analyze speech patterns, enabling more accurate feedback for language learners.

The authors of [96] provide significant contributions to mispronunciation detection and diagnosis (MDD) with their proposal that suggests employing transfer learning to overcome data scarcity in MDD task and uses the pre-trained model wav2vec2.0 to acquire

resilient general acoustic representations. The authors present textual modulation gates emphasizing pertinent text information and reducing irrelevant text, hence improving the model's capacity. Also, they present a supplementary contrastive loss to align the learning objectives of phoneme recognition, thus enhancing the model's learning process and augmenting its efficacy in identifying mispronunciations. Experimental validation on the L2-Arctic dataset demonstrated that the suggested models surpassed traditional techniques, attaining an F1-score of 61.75%. However, Although the F1 score is a crucial performance statistic, it would have been preferable to evaluate the model with other measures, such as precision and recall, which may prove more beneficial.

The article "Non-Autoregressive End-to-End Neural Modeling for Automatic Pronunciation Error Detection" [97] emphasizes the significance of representation learning in its approach. The model employs a self-attention mechanism to identify relationships and implicit language semantics inside input sequences, improving spoken language representation and effectively detecting mispronunciations. The decoder module computes the interaction between focus tokens and designated tokens in the input, enhancing the representations and augmenting the context of the utterance. The outputs serve as inputs for the Pronunciation Detection System (PDS), improving the quality of the representations utilized for mispronunciation detection. The model utilizes a position-wise cross-entropy loss function for optimum parameters, reducing the differences between predicted and observed outcomes and enhancing the quality of learnt representations over time. The work underscores the necessity for comprehensive diagnosis in MDD, necessitating comprehensive representations of pronunciation quality at the phoneme level. Employing representation learning approaches enables the model to attain this degree of precision, enhancing its efficacy in detecting particular pronunciation problems. The research addresses shortcomings in its method, such as data sparsity, dependence on the quality of training data and the complexity of modern MDD pipelines. These variables highlight opportunities for enhancement and further study in automated pronunciation error detection.

Another method for improving pronunciation error detection in mislabelled and imbalanced data using representation learning has been proposed by [98]. The authors in this paper introduce an innovative framework for mispronunciation detection and diagnosis (MDD). The authors suggest a novel architecture that employs various perspectives on identical input data, useful in low-resource environments when annotated L2 data is limited. This method enables the model to acquire more distinctive phonetic representations, which is essential for

efficient mispronunciation detection. The system integrates mono- and multilingual encoders, enabling the model to capture acoustic characteristics across many languages and accents. The model has been designed to acquire articulatory features within a multi-task learning framework, enhancing the encoded representations by incorporating additional tasks that facilitate a deeper comprehension of phonetic complexities. The findings indicate that the suggested model surpasses state-of-the-art models, attaining a phoneme error rate decrease of 11.13% and 8.60%, with the enhancement of 5.89% and 2.49% for the F1 score. This notable improvement in the efficacy of mispronunciation detection systems, particularly when utilizing limited L2 datasets, is especially beneficial for learners without access to large L2 data.

4.5 Anomaly detection (AD)

According to [99], "An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism."(Aggarwal, C. 2017. p.1). Anomaly detection seeks to find odd patterns or outliers in data that differ from expected standards [100] [101], with applications that range from fraud detection to cybersecurity, manufacturing, and healthcare. In this context, the statistical approaches are fundamental as they assume that data follows specified distributions and highlight deviations, such as in the z-score analysis. These approaches have the advantage of being interpretable, however, their main drawback is they struggle to deal with high-dimensional and non-Gaussian data.

Machine learning techniques offer more robust alternatives. Supervised learning utilizes labeled datasets for anomaly categorization, but it is hindered by the absence of labeled outliers. Unsupervised learning with clustering and density-based approaches performs better when anomalies are few and distinct. Meanwhile, semi-supervised learning trains models exclusively on normal data to detect deviations during the test stage [102] [99] [103].

Deep learning algorithms provide revolutionary potential, particularly for high-dimensional or complicated datasets. Variational autoencoders (VAEs) create small, hidden versions of normal data and then look for errors or deviations in the reconstruction to find outliers. Generative Adversarial Networks (GANs) use a generator-discriminator setup to show normal distributions and use samples of the discriminator flags to find outliers. For sequential data, recurrent neural networks (RNNs) capture temporal dependencies and detect deviations in time-series data. Hybrid models combine clustering with VAEs, and ensemble techniques such as isolation forests improve performance by utilizing complementary strengths [104].

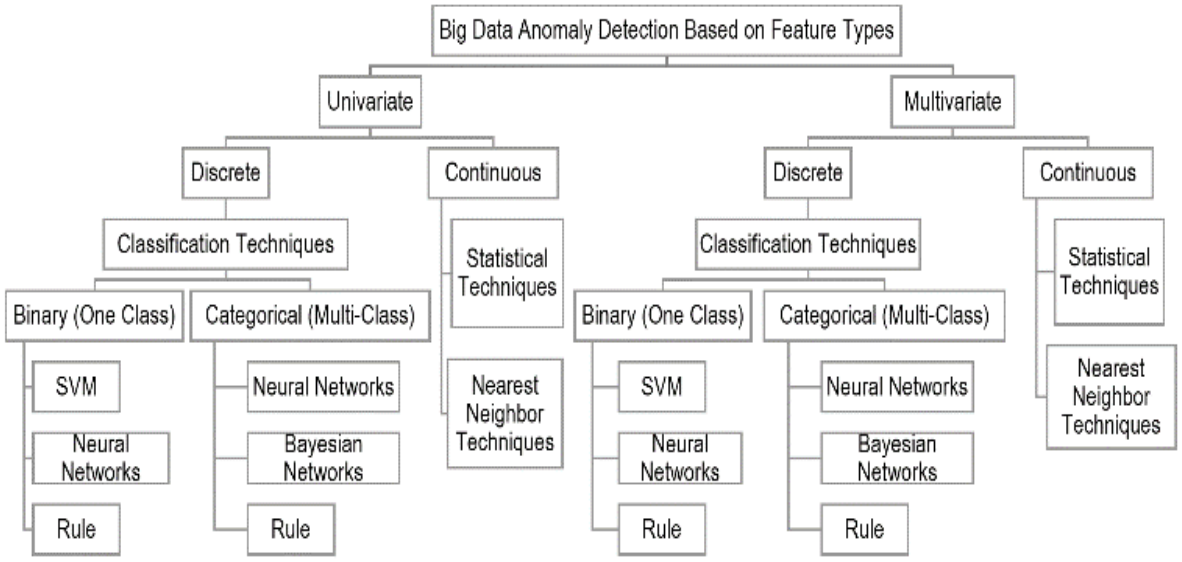


Figure 4.4: Classification Anomaly Detection Techniques Categories

after [105]

Despite these advancements, difficulties remain, especially when High-dimensional datasets hide anomalies, yet changing data patterns necessitate flexible models. The scarcity of labeled anomalies challenges supervised procedures; hence, unsupervised or self-supervised methods are essential. Representation and transfer learning have developed as critical techniques for minimizing these challenges and producing solid data embeddings applicable across activities and domains [104].

4.6 Anomaly detection for MDD (Related work)

The Anomaly Detection (AD) approach has been advantageous across multiple disciplines, especially in fields facing substantial difficulties due to extensive imbalanced data [106]. Nevertheless, in the field of speech processing, specifically in mispronunciation detection, this approach has not been adequately utilized to date despite the critical data-collection challenges present in the CAPT domain.

Shahin and Ahmed (2019) proposed the unique work, we found, that dealt with AD to address pronunciation verification [107]. To address the absence of dedicated mispronounced training data, their study proposes an anomaly detection approach to pronunciation verification, particularly in the context of disordered speech. The authors use a One-Class Support Vector Machine (OCSVM) to model each phoneme from well-pronounced data. Their solution allows the system to learn how common phoneme properties are distributed without the need to samples representing incorrect pronunciations. Speech attribute detectors are used to learn

OCSVM the place and manner of phoneme articulation. During the evaluation step, the OCSVM compares new, unseen speech segments to the learned phoneme models. The OCSVM labels a segment as an anomaly, indicating a mispronunciation if it appears different. The anomaly detection approach was evaluated on speech corpora from children with typical development and those with Childhood Apraxia of Speech (CAS). The results showed that the OCSVM method significantly decreased the number of false rejections compared to conventional approaches like the DNN Goodness of Pronunciation (GOP) algorithm. This suggests that it could be used for more accurate pronunciation evaluation, especially when This study was reworked in light of computer-aided speech therapy [108].

4.7 Representation Learning, Anomaly Detection, and Variational Autoencoders

Representation learning, anomaly detection, and Variational Autoencoders (VAEs) are interrelated topics in machine learning that are essential for comprehending and detecting abnormal patterns in data. Representation learning comprises strategies that autonomously identify the representations necessary for feature detection or classification from unprocessed data. On the other hand, Anomaly detection involves identifying unusual events, instances, or observations that raise suspicion due to their considerable deviation from the predominant data set. Self-supervised techniques have gained popularity in this domain as they provide robust representation learning from unlabeled data. Variational Autoencoders (VAEs) are generative models that encode input data into a latent space and then decode it to recover the original input. They can be utilized to detect anomalies by using their capacity to learn a compact representation of normal data distributions. The relationship between these notions underscores the significance of effective feature extraction techniques in detecting abnormal patterns in data.

4.8 The theory behind VAEs

There is a major difference between variational autoencoder (VAE) and vanilla autoencoder (AE), which lies in the way they operate. Indeed, AE is simply a data reconstruction model based on two phases: compression/decompression, with no concern for data distribution [109]; it learns and extracts features from the data by compressing them in its latent space, then attempts to reconstruct them by decompressing their representations using the decoder. AE is similar to the Principal Component Analysis (PCA) if trained with a linear function [110].

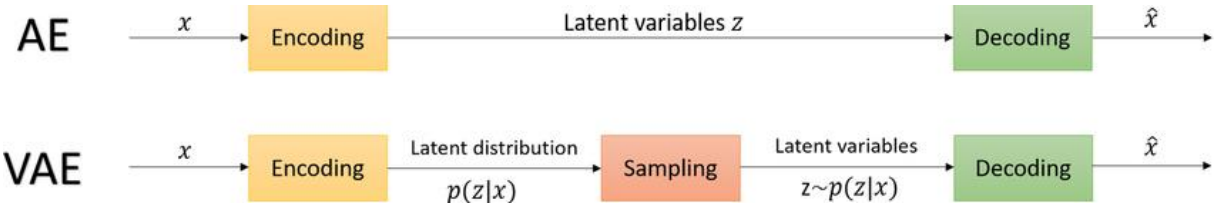


Figure 4.5: The difference between VAE and AE

after [111]

VAE, however, acts differently; it does not rely exclusively on a neural network architecture (encoder and decoder) but follows a Bayesian model embedded in its latent space through a probabilistic distribution function. This function aims to build the P_{model} by observing the correlation between the data and by learning the P_{data}. That's why it's considered one of the last generative models, along with GANs [109].

As VAEs are built on the foundation of AE, we start the section below by explaining in depth the main parts of vanilla autoencoders in terms of functioning, structures, and mathematical foundations. Then, we move to Variational Autoencoders.

4.8.1 Autoencoder

In 1986, (Rumelhart, D. E., et al.) Williams contributed with an effective approach in the field of learning internal representations [112]. This was behind the development of autoencoders by researchers over several decades [113]. Based on the representation learning approach, autoencoders are designed to learn a compressed representation of the input data that captures the most important features of the data, allowing the system to discover unlabeled patterns and data structures [114]. Autoencoders are applied in different domains as data dimensionality reduction, object detection, and image denoising [110].

To create an autoencoder, different types of neural networks can be implemented at the encoder and the decoder; their design might change depending on the particular issue at hand, convolutional autoencoders are one of the most famous architectures of autoencoders [115]. An autoencoder consists of three several parts: (i) an encoder that learns weights and compresses features into a lower dimension representation in the bottleneck/ latent space, (ii) a bottleneck that captures the representations of the inputs, and (iii) a convolutional decoder that decompresses the representations and reconstructs data similar to the original ones. Figure 4.6 illustrates the general principle of an autoencoder; the encoder *h* encodes the initial input *X* into

a latent space *Z*. To approximate the original data *X'*, the decoder *f* decodes the latent space *Z*, making *X' = f (Z) = f (h (X))*. The AE tries to recreate the input as the output after repeated training [109].

To optimize the learning process of an autoencoder, backpropagation is used to minimize the reconstructed error that presents the difference between the original data and the reconstructed data *E (x, ^x)*. The reconstructed loss is generally implemented with Mean Square Error MSE [72] given as

$$\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2$$

Where *N* was the number of samples in a batch.

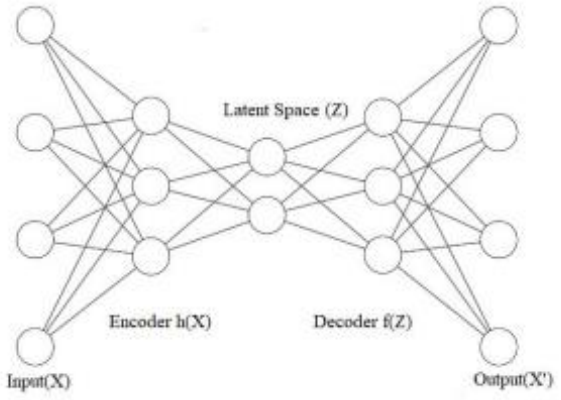


Figure 4.6: Autoencoder architecture

4.8.2 Variational autoencoders (VAEs)

Diederik P. Kingma and Max Welling introduced variational autoencoders in 2013 [88]. VAEs are often compared to autoencoders in terms of architecture, although there are many differences in their objectives and mathematical basis.

Variational autoencoders belong to probabilistic generative models that aim to generate data using distribution estimation and sampling techniques. To clarify, assuming that we have a dataset *X* following an unknown distribution $P_{data}(X)$ in a continuous or a discrete high-dimensional space, the VAE must be able to estimate a distribution $P_{model}(X)$ by observing part of a dataset and learning the unknown distribution $P_{data}(X)$, such that $P_{model}(X)$ is as similar as possible to $P_{data}(X)$. So, with this learning process, the VAE can generate new data [109]. The neural network of the encoder and the decoder in VAEs represent part of their overall structure. The latent space, conversely, contains the parameters of the variational distribution.

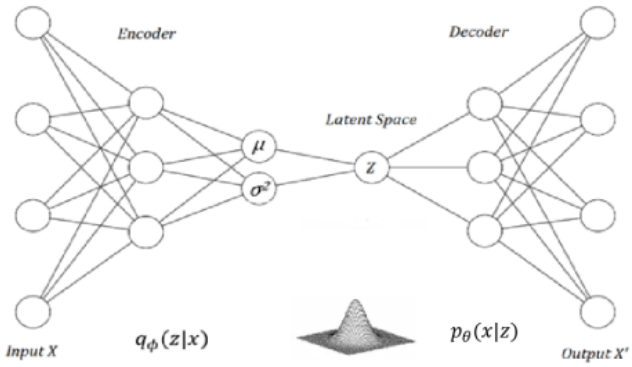


Figure 4.7: Variational autoencoder architecture

4.8.2.1 The Encoder

The encoder maps the sample to a multivariate normal distribution (***Multivariate Gaussian Distribution***) in the latent space (Figure 4.8).

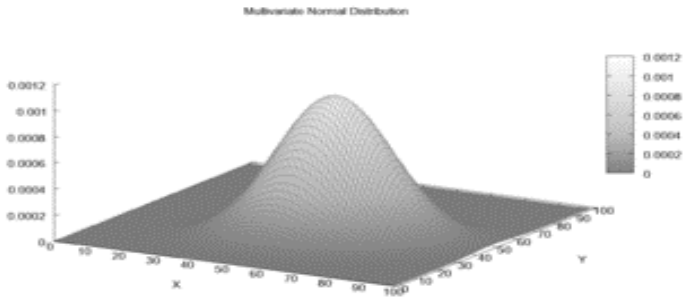


Figure 4.8: Multivariate normal distribution

First, an explanation of a univariate normal distribution (in 1d) must be tackled to understand the concept of a multivariate normal distribution. The univariate normal distribution is a probability distribution that describes a continuous random variable with a single dimension. It is also known as the bell curve and the Gaussian distribution (Figure 4.9.a). The variance (σ^2) and the mean (μ) are the two parameters that define the distribution. The variance indicates the extent of the data distribution, while the mean describes the position of the distribution. The probability density function (PDF) of a univariate normal distribution is given by:

$$f(x) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) * \ exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

here:

x: is the random variable, μ: is the mean, σ^2: is the variance and exp: is the exponential function.

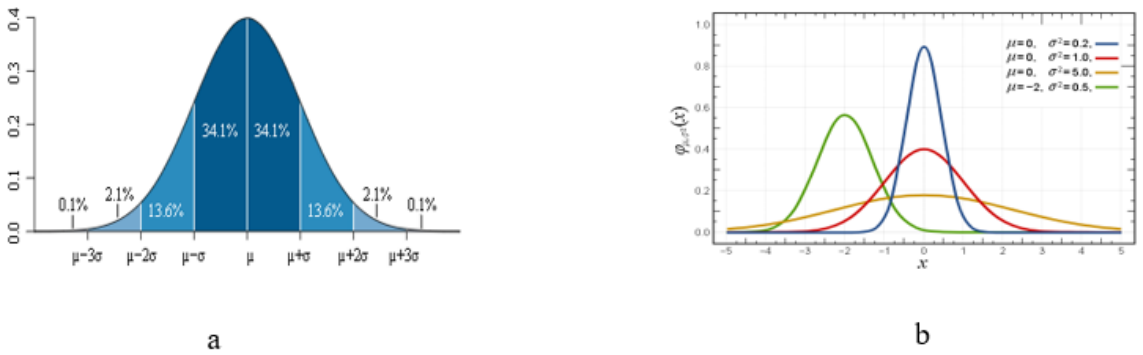


Figure 4.9: Univariate normal distribution

The red curve in (Figure 4.9.b) where μ=0 and σ=1 represents the standard normal distribution. It is a continuous probability distribution that is widely used in statistics and machine learning

To sample a point z from a univariate normal distribution

$$z = \mu + \sigma\varepsilon$$

Where ε is a sampled point in a standard normal distribution

So, the multivariate normal distribution is a generalization of the one-dimensional (univariate) normal distribution to n dimensions.

$$f(x_1, \ldots, x_k) = \frac{e^{-\frac{1}{2}(\vec{x}-\vec{u})^T \Sigma^{-1}(\vec{x}-\vec{u})}}{\sqrt{2(\pi)^k |\Sigma|}}$$

- The formula is a representation of the probability density function (PDF) of a multivariate normal distribution. This PDF describes the probability distribution of a k-dimensional random variable x (x is a real k-dimensional column vector), where each x component is normally distributed with a mean of u_i and a variance of Σ_{ij}. With consideration for the correlations and variances of the variables.
- The Mahalanobis distance in the formula $\sqrt{(\mathrm{x}-\mathrm{u})^{\mathrm{T}}\Sigma^{-1}(\mathrm{x}-\mu)}$ represents a multivariate distance metric that calculates the distance between a point and a distribution.

- The term $\sqrt{2(\pi)^k|\Sigma|}$ is a normalization constant that ensures that the PDF integrates to one over the entire k-dimensional space.
- **VAE in 2D (**sampling a point in the latent space):

to simplify understanding of the theory behind N-dimensional VAE, a quick illustration of the 2D process is given below.

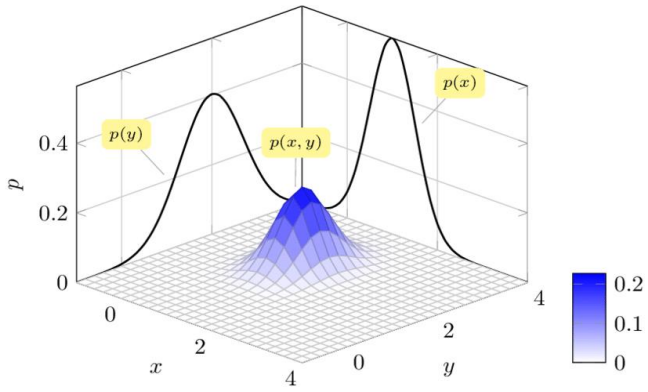


Figure 4.10: bivariate normal distribution

With $k = 2$, the probability density function becomes:

$$f(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2 - 2\rho\left(\frac{x-\mu_X}{\sigma_X}\right)\left(\frac{y-\mu_Y}{\sigma_Y}\right) + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]\right)$$

Where ρ is the correlation between X and Y and $\sigma x > 0$ $\sigma y > 0$, in this case u is presented with the vector $u = \begin{pmatrix} u_x \\ uy \end{pmatrix}$ and the covariance is presented with the matrix:

$$\Sigma = \begin{pmatrix} \sigma_x^2 & \rho_{\sigma_x}\rho_{\sigma_y} \\ \rho\sigma_x\rho\sigma_y & \sigma y^2 \end{pmatrix}$$

Variational Autoencoders assumes that all dimensions are independent [88]. Thus, the complexity of the problem is reduced to the diagonal matrix corresponding to the variance as there is no dependency between the correlations on the different axes. i.e, $\rho = 0$

$$\Sigma = \begin{pmatrix} \sigma_x^2 & \rho_{\sigma_x}\rho_{\sigma_y} \\ \rho\sigma_x\rho\sigma_y & \sigma y^2 \end{pmatrix} \Rightarrow \Sigma = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma y^2 \end{pmatrix}$$

In this case the encoder of the VAE maps each input data to a mean vector $\vec{u} = \begin{pmatrix} u_x \\ u_y \end{pmatrix}$ and a variance vector $\vec{\sigma}^2 = \begin{pmatrix} \sigma_x^2 \\ \sigma_y^2 \end{pmatrix}$ in the latent space Z

Therefore, sampling from a multivariate normal distribution is as follow:

$$z = \vec{u} + \Sigma\varepsilon$$

Where Σ is as follow

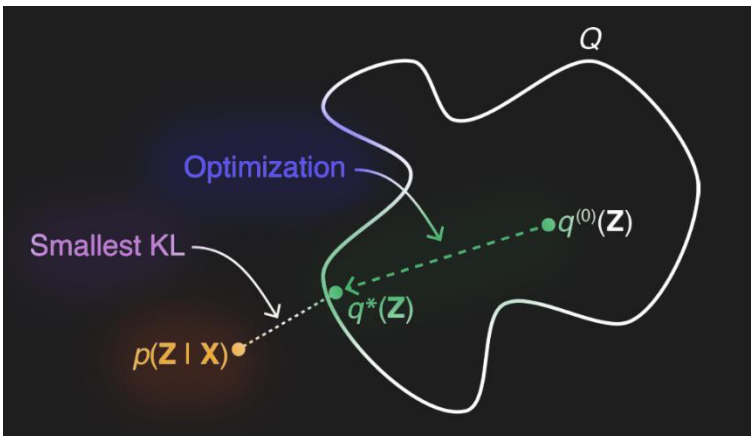
$$\Sigma = e^{\frac{log(\vec{\sigma}^2)}{2}}$$

4.8.2.2 The loss function

The main function of generative models is density estimation, which frequently implies comparing two distributions. This is realized by determining how similar the two distributions are. A popular similarity metric used in VAE is the Kullback-Leibler (KL) divergence [116]. The loss function of a VAE can be presented as follows:

$$\text{DKL} = \text{Loss function (MSE)} + \text{Regularization term (KL Divergence)}$$

The first term of the formula bellow refers to the loss function, and the second one is the KL function, which refers to the regularization term in the objective function; it measures the difference between the approximate posterior distribution (learned by the encoder network) and the prior distribution (the standard normal distribution) of the latent variables. The sum of the KL divergence and the loss function aims to maximize the objective function ELBO (evidence lower bound) [89], which is the fundamental idea of variational inference [116].



In a VAE, the normal distribution with parameters μ and σ is compared to a standard normal distribution. The KL-divergence is calculated by mapping σ^2 to the logarithm of the variance.

Thus, the loss function can be presented as follows:

$$DkL\big(N(\mu,\sigma)\big)\big(N(0,1)\big) = \frac{1}{2}\Sigma(1 + log(\sigma^2) - u^2 - \sigma^2)$$

4.9 Proposed method

Our main contribution, presented in this chapter of our dissertation, is in the field of Computer-Assisted Language Learning (CALL), with particular attention to Computer-Aided Pronunciation Training (CAPT). Thus, we aim to address a significant challenge in the mispronunciation detection and diagnosis (MDD) module by developing a system that may reuse learner-generated pronunciation data for autonomous training, enhancing the efficacy and adaptability of CAPT tools, see Figure 4.11.

Main Points of our Proposal:

1. *Reusing Learner Data:*

In standard CAPT setups, user recordings remain underutilized after being analyzed for feedback. Our approach aims to use this valuable data again, making it a resource for enhancing and enlarging the system's training database. As these recordings do not have clear labels, i.e., correct or incorrect pronunciations, the system must operate autonomously, allowing scalability and efficiency.

2. *Self-supervised Learning Method:*

Our system can extract meaningful hidden features from raw speech data using representation learning with variational autoencoders (VAEs). Hence, it allows the discovery of patterns and irregularities, boosting model performance without needing labeled data.

3. *Enhancing CAPT Performance:*

By continuously incorporating user data, the system becomes better at handling different accents, speech patterns, and pronunciation differences, offering learners more accurate and personalized feedback.

This ongoing improvement helps ensure that CAPT tools remain relevant and effective for users with diverse language backgrounds.

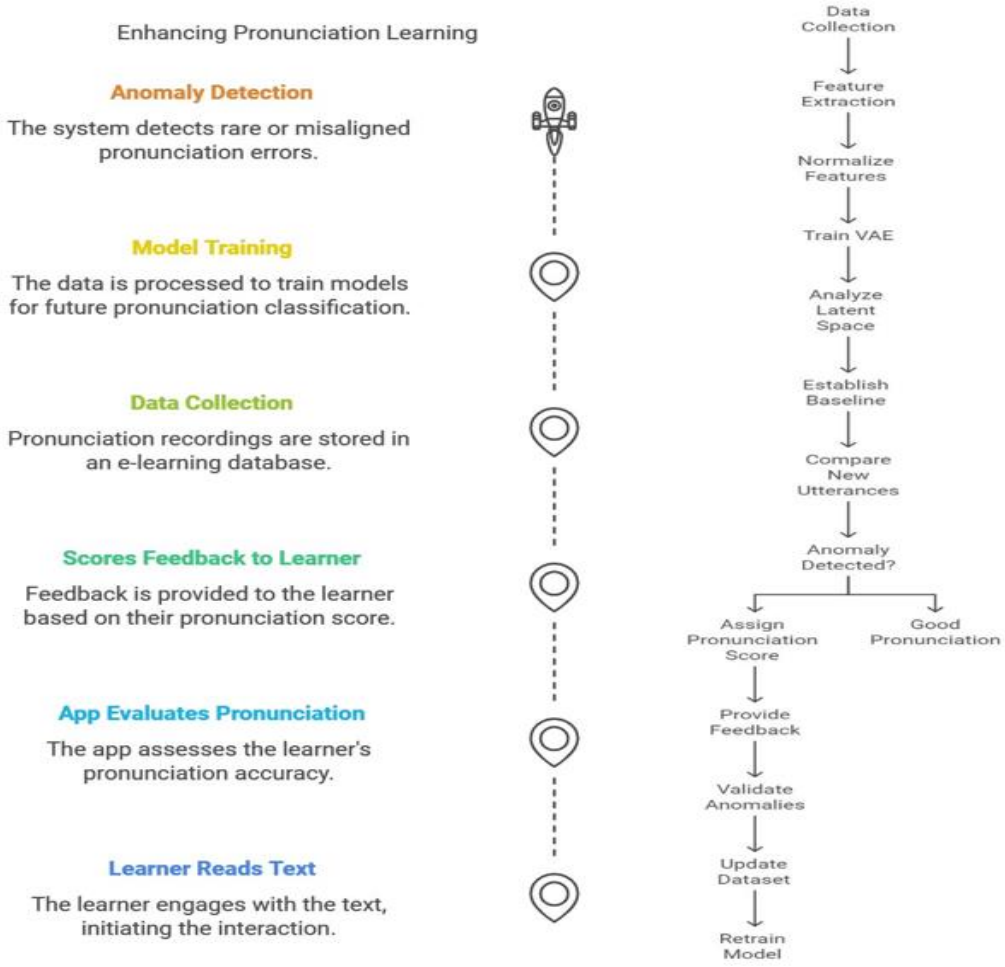


Figure 4.11: Reusing learner-generated pronunciation in real CAPT app

Figure 4.12 gives an overview of our proposed approach to pronunciation error detection using a variational autoencoder (VAE). The VAE is trained on correct pronunciations to learn how to represent, understand, and structure the data in its latent space. By attempting to reconstruct new samples during training, if a sample deviates significantly from the patterns learned during

training, the system identifies it as an 'outlier', i.e., a potential mispronunciation. To do this, it uses a density-based anomaly detection method, which flags irregularities in the latent space.

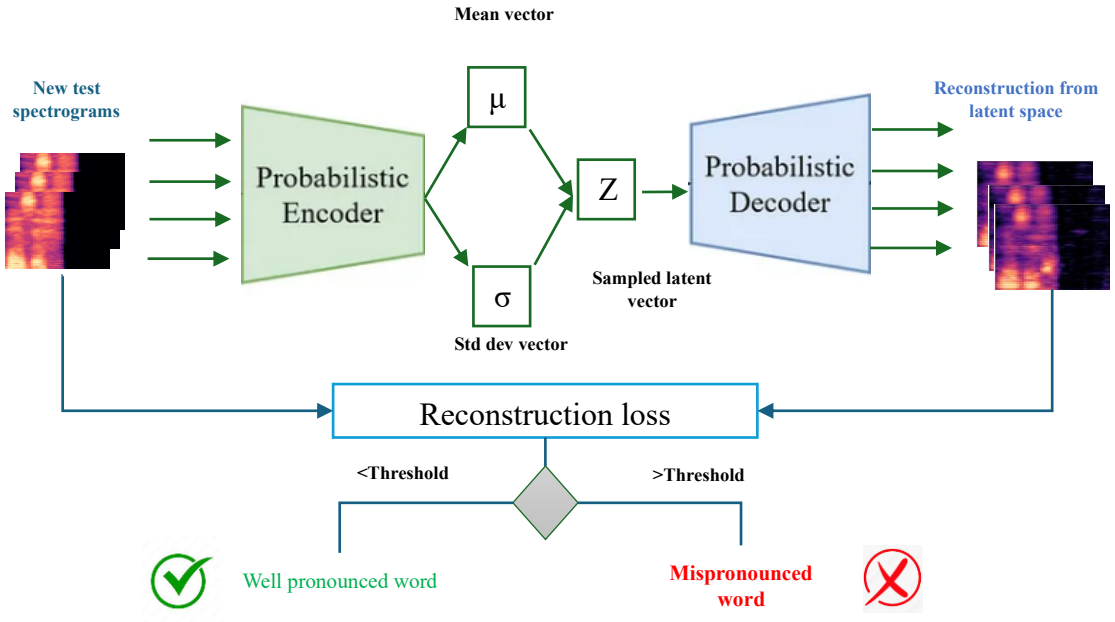


Figure 4.12: System outline

In summary, the Workflow where we contribute with our solution is as follows:

a. User Interaction: input collection

Learner reads text ➡ app evaluates pronunciation ➡ Scores feedback to learner

b. Data collection: disordered and unlabeled records

Learner's pronunciation recordings ➡ Storage in e-learning database

c. Model training for future pronunciation classification

- Processing with representation learning: Unstructured data ➡ Feature extraction ➡ Latent space representation
    - Tools: Autoencoders, Variational Autoencoders (VAEs), Deep Neural Networks (DNNs).
- Anomaly detection: latent representations ➡ Detection of rare or misaligned errors

4.9.1 Preprocessing pipeline

In the first step of the procedure, the dataset is prepared by classifying audio recordings into correct and erroneous pronunciations. These recordings are then stored in separate folders so that the dataset's processing may be optimized. The training dataset consists entirely of correctly pronounced utterances to establish a reliable baseline. On the other hand, the test dataset combines correct and wrong pronunciations to evaluate the model's performance.

a. Padding and alignment:

Padding was used to establish a constant duration of 1.11 seconds for each audio recording, guaranteeing consistency in the data entry. This step is necessary because machine learning models, including deep learning-based approaches, require consistent input in shape and size to function successfully. Adding padding to shorter recordings or truncating longer ones allows for preserving the temporal structure of the audio while adding quiet to shorter recordings.

b. Extraction of spectrogram:

Spectrograms were computed using the audio recording to represent the time-frequency features of speech. A spectrogram is a two-dimensional visual representation that maps sound frequencies against time. It offers a comprehensive feature set that can be utilized for analyzing speech. Spectrograms can capture pitch, tone, and energy fluctuations, all of which are essential for differentiating between correct and erroneous pronunciations.

c. Normalization:

After the spectrograms were generated, amplitude normalization was carried out to bring all data points into a consistent range. In this stage, differences produced by different recording volumes and environmental noise are eliminated. This step ensures that the model focuses on relevant pronunciation features rather than external artifacts.

d. Storage using a standardized format:

The normalized spectrograms were saved in a standardized binary file format, such as. py or.h5, which enables efficient storage and retrieval of the data during model training and testing.

These stages enable the model to acquire reliable representations of accurate pronunciations, guaranteeing that the input features are consistent and of high quality. By concentrating on spectrograms as input features, the system can use comprehensive audio

characteristics, which enables it to facilitate the exact detection of abnormalities or deviations that indicate improper pronunciations.

4.9.2 VAE architecture

In the following table an illustration of the proposed variational autoencoder (VAE) architecture, which consists of two mirrored components: the encoder and the decoder. The encoder compresses the input data into a compact latent space representation, effectively capturing its essential features and serving as the bridge between the encoder and decoder. Thus, the decoder reconstructs the original input data from this latent representation, aiming to preserve its key characteristics while minimizing reconstruction errors. This design enables the VAE to learn meaningful data distributions for our proposed anomaly detection for MDD.

Table 4.1: VAE architecture

Layer (type)	Output shape	#Param
encoder_input (InputLayer)	[(None, 256, 96, 1)]	0
encoder (Functional)	(None, 128)	1,964,896
decoder (Functional)	(None, 256, 96, 1)	896,225

According to the VAE architecture above, we have:

1. An Input layer:

This layer has a shape of (None, 256, 96, 1), representing a set of spectrogram inputs with a height of 256, width of 96, and one channel. It transfers the input spectrograms to the encoder.

2. An encoder:

With a count of around 1.96M of its parameters of multiple layers. The input is compressed into a latent space representation of shape (128).

3. A latent Space (The representation layer / the bottleneck):

Represents a compressed, probabilistic version of the input that includes the most important information for reconstruction.

4. A decoder:

Symmetrical with the encoder. Its role is to Expand the latent space representation back to its

original shape (256, 96, 1). It Contains approximately 896k parameters, indicating layers for high-quality reconstructions.

5. An Output Layer:

Creates the reconstructed spectrogram that matches the original input shape.

4.9.2.1 The encoder architecture

Table 4.2: Encoder architecture

	Layer (type)	Conv. kernels	Conv. strides	Output Shape	#Param	Connected to
Input	encoder_input(InputLayer)			(None,256,96,1)	0	
1st conv	encoder_conv_layer_1 (Conv2D)	3	2	(None,128,48,256)	2560	['encoder_input[0][0]']
	encoder_relu_1 (ReLU)			(None,128,48,256)	0	['encoder_conv_layer_1[0][0]']
	encoder_bn_1 (BatchNormalization)			(None,128,48,256)	1024	['encoder_relu_1[0][0]']
2nd conv	encoder_conv_layer_2 (Conv2D)	3	2	(None,64,24,128)	295,040	['encoder_bn_1[0][0]']
	encoder_relu_2 (ReLU)			(None,64,24,128)	0	['encoder_conv_layer_2[0][0]']
	encoder_bn_2 (BatchNormalization)			(None,64,24,128)	512	['encoder_relu_2[0][0]']
3rd conv	encoder_conv_layer_3 (Conv2D)	3	2	(None,32,12,64)	73,792	['encoder_bn_2[0][0]']
	encoder_relu_3 (ReLU)			(None,32,12,64)	0	['encoder_conv_layer_3[0][0]']
	encoder_bn_3 (BatchNormalization)			(None,32,12,64)	256	['encoder_relu_3[0][0]']
4th conv	encoder_conv_layer_4 (Conv2D)	3	(2,1)	(None,16,12,32)	18,464	['encoder_bn_3[0][0]']
	encoder_relu_4 (ReLU)			(None,16,12,32)	0	['encoder_conv_layer_4[0][0]']
	encoder_bn_4 (BatchNormalization)			(None,16,12,32)	128	['encoder_relu_4[0][0]']
Dense	flatten (Flatten)			(None,6144)	0	['encoder_bn_4[0][0]']
Mu layer	mu (Dense)			(None,128)	786,560	['flatten[0][0]']
Log_var layer	log_variance (Dense)			(None,128)	786,560	['flatten[0][0]']
Output	encoder_output (Lambda)			(None,128)	0	['mu[0][0]', 'log_variance[0][0]']

Encoder Workflow Overview

1. An Input Layer:

Accepts input with dimensions (256, 96, 1), representing a spectrogram of constant size. This layer guarantees the correct data formatting for the following convolutional layers.

2. Convolutional Layers (First to Fourth):

Each convolutional layer systematically reduces the spatial dimensions of the input while augmenting the depth (number of filters).

- Activation functions: ReLU is utilized after to each convolution to introduce non-linearity.
- Batch normalization is applied after each activation to maintain stability and improve training by normalizing the outputs of layers.

3. Flatten Layer:

Transforms the feature mappings from the final convolutional layer into a one-dimensional vector for input into dense layers with an output shape of (6144).

4. Latent Space Representation:

Two dense layers calculate the latent variable distribution's mean (mu) and log-variance (log_variance). Each layer generates a latent space of dimension (128).

5. Reparameterization (Lambda Layer):

Utilizes mu and log_variance to produce the latent variable through the reparameterization method. This stage facilitates backpropagation using stochastic sampling.

6. Encoder Output:

A compressed latent representation of the input data is provided to the decoder for reconstruction.

4.9.2.2 The decoder architecture

Table 4.3: Decoder architecture

	Layer (type)	Conv kernels	Conv strides	Output Shape	#Param
Input layer	decoder_input (InputLayer)	3	(2,1)	(None, 128)	0
Dense layer	decoder_dense (Dense)			(None, 6144)	792,576
Reshape layer	reshape (Reshape)			(None, 16, 12, 32)	0
1st conv. bloc	decoder_conv_transpose_layer_1 (Conv2DTranspose)	3	2	(None, 32, 12, 32)	9248
	decoder_relu_1 (ReLU)			(None, 32, 12, 32)	0
	decoder_bn_1 (BatchNormalization)			(None, 32, 12, 32)	128
2nd conv. bloc	decoder_conv_transpose_layer_2 (Conv2DTranspose)	3	2	(None, 64, 24, 64)	18,496
	decoder_relu_2 (ReLU)			(None, 64, 24, 64)	0
	decoder_bn_2 (BatchNormalization)			(None, 64, 24, 64)	256
3rd conv. bloc	decoder_conv_transpose_layer_3 (Conv2DTranspose)	3	2	(None, 128, 48, 128)	73,856
	decoder_relu_3 (ReLU)			(None, 128, 48, 128)	0
	decoder_bn_3 (BatchNormalization)			(None, 128, 48, 128)	512
4th conv. layer	decoder_conv_transpose_layer_4 (Conv2DTranspose)	3	2	(None, 256, 96, 1)	1153
Activation function	sigmoid_layer (Activation)			(None, 256, 96, 1)	0

Overview of the decoder:

1. Input Layer:

With a dimension of (None, 128), the decoder receives input from the latent space representation (size 128), which encompasses the encoded features of the input data. This input initiates the reconstruction process in the decoder.

2. Dense Layer

Its shape is (None, 6144); it transforms the input from the latent space into a comprehensive, completely connected vector of size 6144. This layer converts latent space information into a format suitable for spatial reconstruction in subsequent stages.

3. Reshape Layer

It converts the output from the dense layer into a 3D tensor and reformulating it to (None, 16, 12, 32). This configuration pertains to a 16x12 spatial grid with 32 feature channels designated for following convolutional processes in the next layers.

4. Convolutional Blocks

Each convolutional block of the decoder uses Conv2DTranspose to progressively increase the sampling of the feature maps, by increasing the spatial dimensions. This process reconstructs the data from the compact representation of latent space into a larger, more detailed output approximating the original input. As the spatial dimensions increase, the number of feature maps (channels) also increases in each successive block, allowing more complex patterns and details to be captured when reconstructing the data.

Non-linearity and Normalization: After each transposed convolution, a ReLU activation function is applied to introduce non-linearity, allowing the network to learn complex relationships. Batch Normalization follows to stabilize learning by normalizing the activations and speeding up training.

5. Final Output Layer:

The final transposed convolutional block produces an output of the desired size and shape, typically a reconstruction of the original input (The spectrogram). Thus, the last deconvolution layer increases the feature map to the specified resolution (256x96) and diminishes the channel count to 1, aligning with the output image or signal's singular channel.

6. Sigmoid Activation Function

Its output dimensions are (None, 256, 96, 1). It utilizes the sigmoid function on the output of the final convolutional block. This activation function constrains the output to the interval [0, 1], guaranteeing that the rebuilt data remains within an appropriate scale.

4.9.3 Anomaly Detection Algorithm

The variational autoencoder (VAE) is an effective generative model applicable to anomaly detection via its reconstruction method, wherein abnormalities are identified by elevated reconstruction errors [101].

Our work to deploy the AD method has adhered to four primary steps:

A. In the training phase:

1. Calculate the difference between the original samples and those produced by the model.

2. Generate an error vector derived from the error term of each sample

3. Establish the threshold by referencing the extreme value of this vector.

B. During the testing phase:

4. Calculate the reconstruction error and categorize samples as abnormal if the error exceeds the threshold.

4.10 Results and discussion

4.10.1 ASMDD dataset

The largest dataset of Arabic speech mispronunciation errors in Egyptian dialogues is used [117]. A dataset of speech records of nursery school constructed with Audacity software tool, producing mono-channel audio files with a 44.1 kHz sampling rate and 32-bit resolution. The vocabulary items consist of 100 isolated Arabic words. The wave files are named using an ID representing words ranging from 0 to 99. Moreover, the name of files includes information on whether the word is well-pronounced or not. The dataset is organized by gender and number of pronounced words, with folders indicating child gender and pronounced words. The dataset aims to improve speech representation models and develop Arabic language pronunciation mistake identifiers. The following table presents the pronounced words and their respective indexes:

Table 4.4: ASMDD dataset

Index	Word	Index	Word	Index	Word	Index	Word	Index	Word
1	نعم	21	الطريق	41	للغاية	61	المدرسة	81	ولد
2	رجل	22	عمل	42	فتاة	62	الصباح	82	رسالة
3	بخير	23	الجميع	43	كبيرة	63	الماء	83	عائلة

4	شخص	24	جيدة	44	آسفة	64	التحدث	84	القائد
5	الوقت	25	المال	45	الأرض	65	الساعة	85	المرأة
6	اليوم	26	الذهاب	46	البيت	66	الليل	86	الطبيب
7	صحيح	27	أرجوك	47	صباح	67	نهاية	87	اسم
8	أستطيع	28	المنزل	48	ألم	68	حياة	88	النقود
9	شكرا	29	الحياة	49	لحظة	69	الواقع	89	الكلام
10	الناس	30	انتظر	50	بالضبط	70	الطفل	90	مدينة
11	أعلم	31	الرجال	51	رقم	71	دكتور	91	مساء
12	رائع	32	الله	52	طريق	72	الهاتف	92	الشمس
13	مرحبا	33	الباب	53	المدينة	73	الطعام	93	ارجوك
14	آسف	34	جميل	54	الرئيس	74	فريق	94	السماء
15	تعال	35	الشرطة	55	صديقي	75	الفتى	95	الزواج
16	بالطبع	36	السيارة	56	ساعة	76	اللقاء	96	أصدقاء
17	العالم	37	النار	57	غرفة	77	نظرة	97	مكتب
18	الحقيقة	38	عظيم	58	عام	78	النساء	98	البحر
19	الليلة	39	الخير	59	الأطفال	79	العشاء	99	الكتاب
20	أمي	40	حالك	60	سنة	80	الأسبوع	100	الشارع

According to the article, there are 100 records + 50 records for speakers 00 to 29 and speakers 30 to 99, respectively, giving us a count of (100X30) +(50X70) = 6500 records. However, when downloading the dataset from the URL mentioned in the article, we found only 5297 WAV files divided between correct and incorrect pronunciations. As shown in Figure 4.13 and Figure 4.14, the ASMDD dataset reflects an unequal distribution of classes at the word level in favor of the "well-pronounced " class. This motivated us to use the anomaly detection approach, as mentioned in the previous sections, to perform the classification.

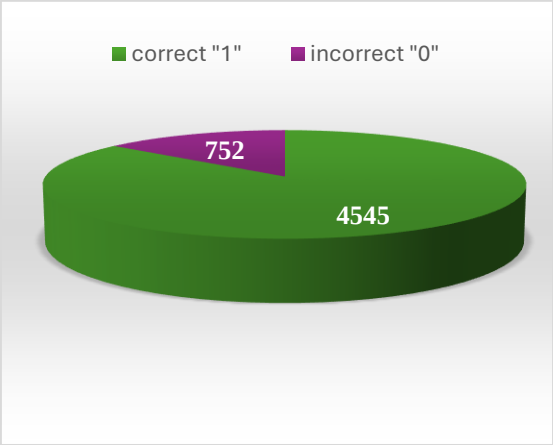
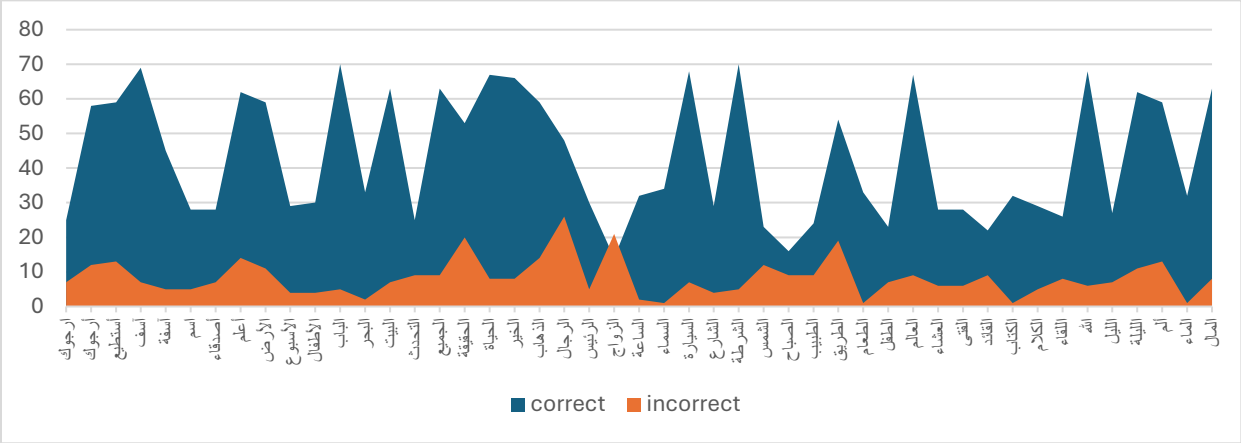


Figure 4.13: Correct and incorrect classes in the ASMDD dataset



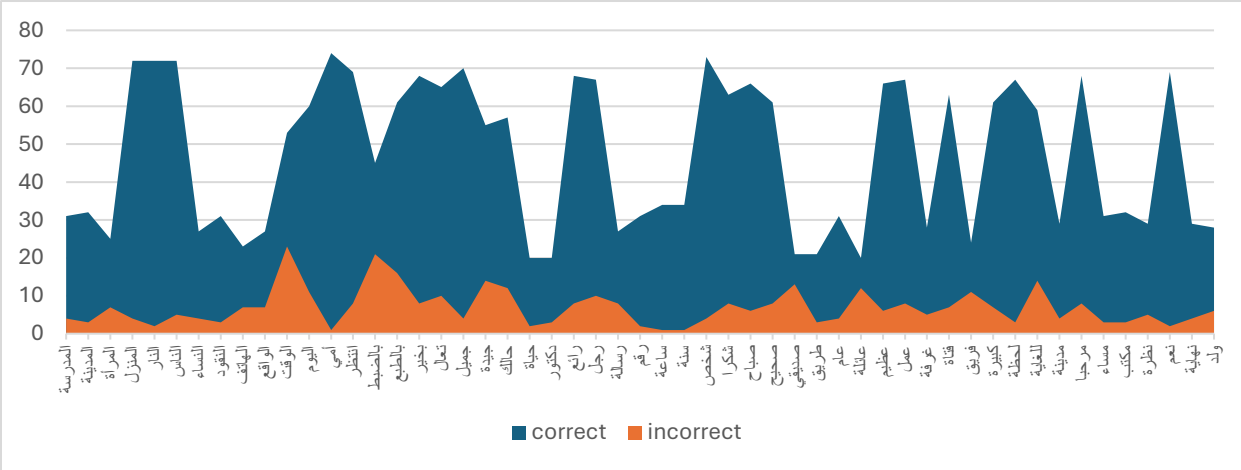


Figure 4.14: Imbalanced classes per word

4.10.2 Performance measures

4.10.2.1 Qualitative evaluation

4.10.2.1.1 Generation performance with VAE

Testing the VAE's generation capabilities is essential to ensure that it can effectively capture normal patterns and accurately identify anomalies in the data. Therefore, before studying the distribution in the latent space, we tested some generations and obtained synthetic spectrograms (Figure 4.15). The resulting spectrograms were passed to the Griffin-Lim reconstruction algorithm to synthesize speech [118]. We used the audacity tool to listen to the generated samples and obtained very satisfactory generations that were clear enough to understand the spoken word.

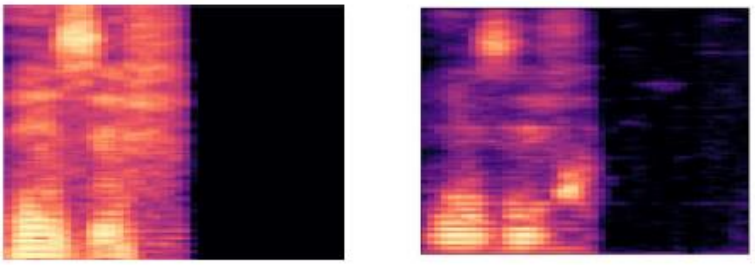


Figure 4.15: An illustration of the original and generated spectrograms of the word

We visualized the scaled amplitude of the two signals to estimate the difference between the original and generated signals. We took as an example a generation of the word "?asif"“آسف”:

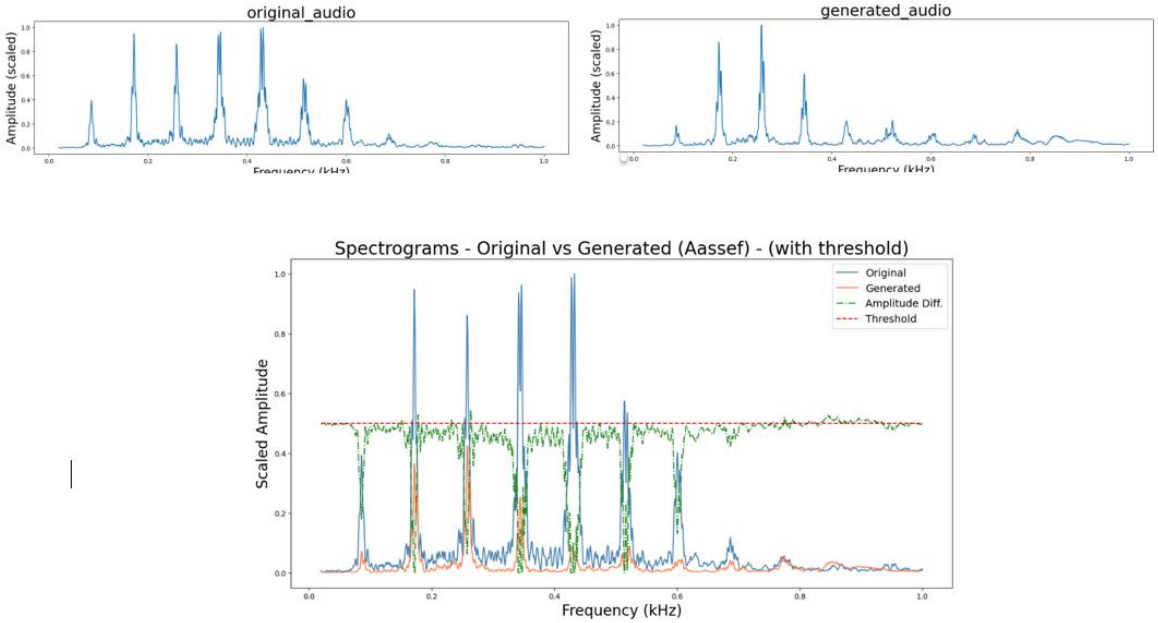


Figure 4.16: The difference between original and generated waves in terms of scaled amplitude word "آسف"

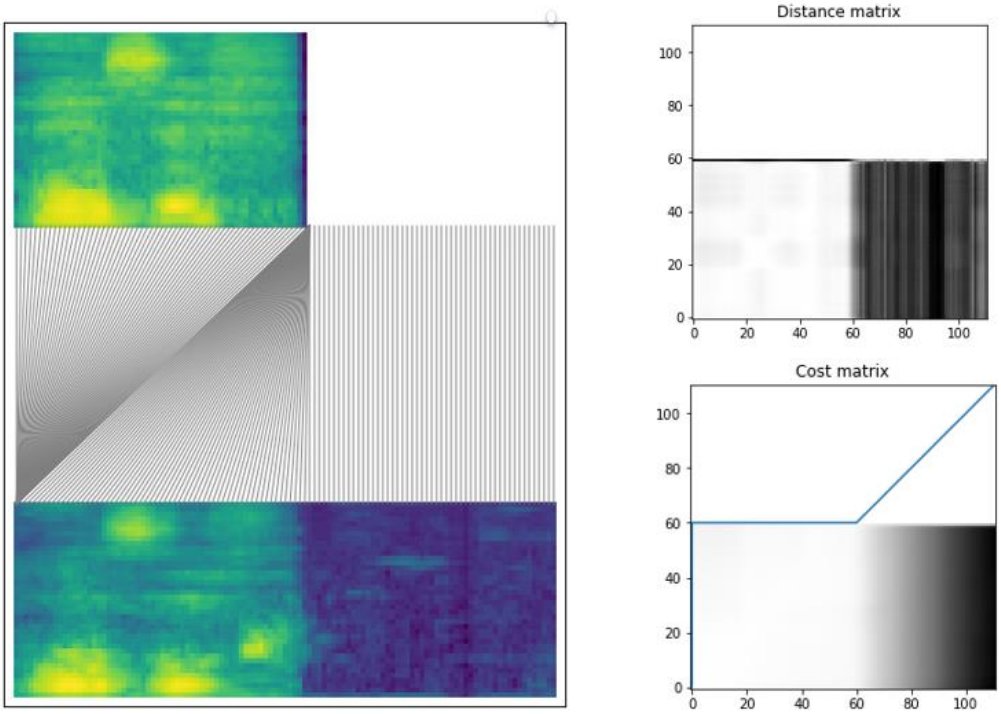


Figure 4.17: DTW between original and generated spectrograms

The scaled amplitude of the word "آسف" extracted from the original and generated audio signals: the figure shows that the difference in amplitude is below the threshold, which means that the two words are close.

We also used DTW to measure the similarity between the two spectrograms (original and generated of the word "آسف"); their alignment in Figure 4.17 shows they are very nearby.

4.10.2.1.2 Data Visualization

The latent space of a variational autoencoder (VAE) signifies a compressed, continuous, and organized representation of input data. Assessing the quality of this latent space is essential for comprehending the efficacy of the VAE in learning the fundamental data distribution. Qualitative evaluation with data visualization offers insights into the latent space's structure, facilitating the assessment of the model's efficacy in clustering similar data points and preserving significant relationships.

Thus, Common methods include t-SNE, PCA, and UMAP, which emphasize local clusters. These methods are used to visualize the latent space to detect clusters, evaluate continuity and structure, comprehend links between latent variables and input characteristics, and pinpoint areas of misrepresentation or inadequate reconstructions. Owing to the elevated

dimensionality of latent spaces, dimensionality reduction techniques are employed to project data into two or three dimensions for display. Standard methods include (t-SNE), (PCA), and (UMAP), which emphasize local clusters. [119] [120].

4.10.2.1.3 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a commonly employed method for dimensionality reduction and visualization, especially effective at handling high-dimensional datasets. PCA converts original variables into a new set of variables with no correlation called principal components, retaining the most essential features of the data while decreasing its dimensionality. This procedure not only simplifies data processing but also improves visualization, facilitating the identification of relationships and patterns within the data [121] [122].

The Mathematical Foundation of PCA is as follows: [121]

- **Data Representation:** Given a dataset with n samples, each with d dimensions, represented as $\mathrm{X} \in \mathbb{R}^{n \times d}$, where rows correspond to samples and columns to features.
- **Centering the Data:** Subtract the mean of each feature from the dataset

$$\mathrm{X}_{centered} = \mathrm{X} - \mu$$

Where μ is the mean vector of the dataset.

- **Covariance Matrix:** Compute the covariance matrix C

$$C = \frac{1}{n-1}\mathrm{X}_{centred}^{\mathrm{T}}\mathrm{X}_{centered}$$

This matrix captures the relationships between features.

- **Eigen Decomposition:** Compute the eigenvalues λ_i and eigenvectors $\mathbb{v}_i$ of C

$$C_{\mathbb{v}_i=}\ \lambda_i \mathbb{v}_i$$

Eigenvectors define the directions of the principal components, while eigenvalues determine the variance along each component.

- **Dimensionality Reduction:**

  Sort eigenvectors by their corresponding eigenvalues in descending order.

  Select the top k eigenvectors to form a projection matrix P:

$$\mathrm{P} = [\mathbb{v}_{i1}, \mathbb{v}_2, \dots, \mathbb{v}_k]$$

  - Transform the data to the new k-dimensional space:

$$\mathrm{X}_{reduced} = \mathrm{X}_{centered}\mathrm{P}$$

4.10.2.1.4 Data distribution visualization in VAE's bottleneck with PCA

We utilized the principal component analysis (PCA) in our proposal to visualize the data distribution in the VAE's latent space during both the training and testing stages. The PCA reduces the high-dimensional latent representations to a 3D lower-dimensional space. Hence, it enables us to interpret the structure of the data effectively.

PCA during training illustrates the clustering of normal pronunciations (Figure 4.18), offering insight into the efficacy of the VAE in encoding correct patterns. During the testing phase, PCA helps in the identification of mispronunciations by showing deviations or outliers from the distribution of the training data (Figure 4.19 & Figure 4.20). In Figure 4.19, the dataset is composed only of mispronunciation, whereas in Figure 4.20, we used latent space to display the distributions of 400 unseen samples: 200 normal data with "correct pronunciations" and 200 abnormal data with "incorrect pronunciations" for predictions. This method accurately evaluates the VAE's capacity to distinguish between standard and anomalous pronunciations.

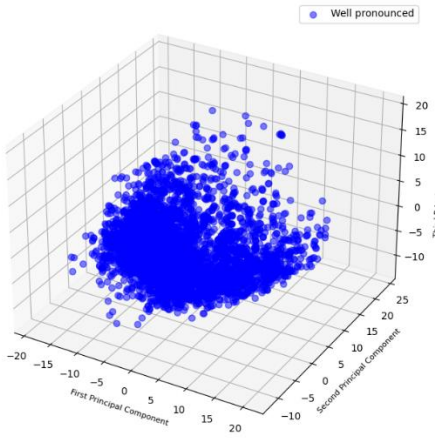


Figure 4.18: Data distribution visualization of the training stage (normal data)

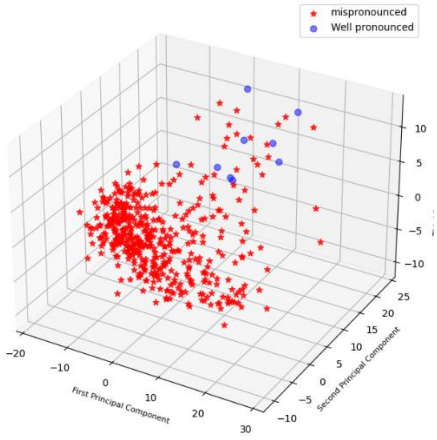


Figure 4.19: Visualization of abnormal data in the test stage (only abnormal data)

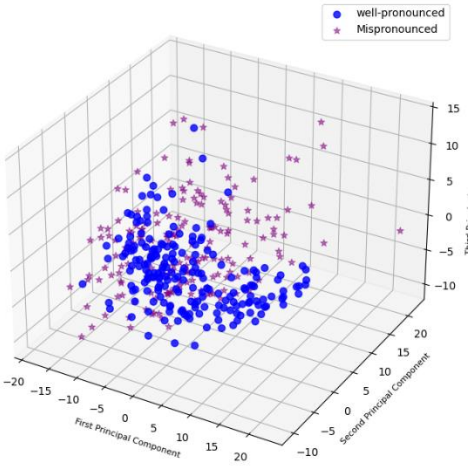


Figure 4.20: Visualization of unseen data during test 200 well-pronounced words+200 mispronounced words

To compare the variational autoencoder (VAE) with the autoencoder (AE), an AE was implemented using the same encoding and decoding architecture as the VAE but without the mu and log-variance layers. The result is a 128-dimensional latent space without the normal distribution constraint inherent in VAE. To ensure a fair comparison, both models were trained with identical hyperparameters, including learning rate and number of epochs.

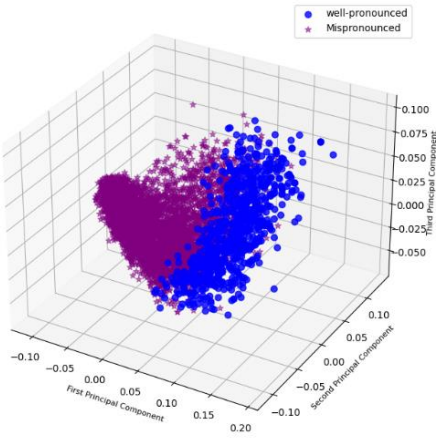


Figure 4.21: Visualization of the dataset samples in AE latent space with PCA

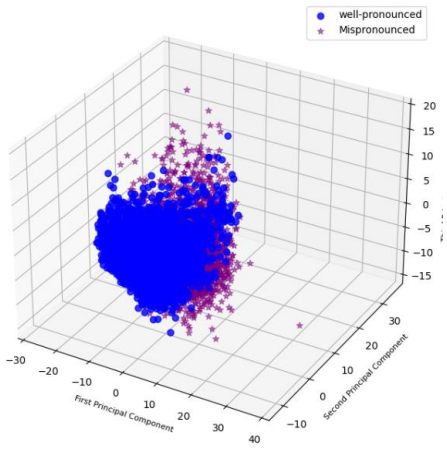
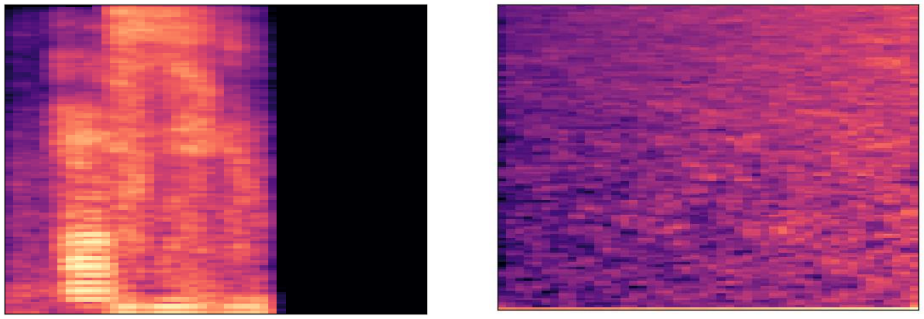


Figure 4.22: Visualization of the dataset samples in VAE latent space with PCA

The latent space visualizations in the figures above reveals significant differences. The AE latent space (Figure 4.21) shows a less structured distribution with more anomalies than in the dataset, with considerable overlap between ‘well pronounced’ and ‘poorly pronounced’ data points. This is because AE finds it challenging to distinguish these categories accurately, as shown by the generation of noisy and imprecise reconstructions. In contrast, the VAE latent space (Figure 4.22) is more organized and has a more explicit distribution, demonstrating the advantage of its probabilistic approach. VAE imposes a structured and meaningful latent space, which allows for better visualization and interpretation, ultimately leading to improved performance in distinguishing and reconstructing the input data discussed in the next section.



Original Generated

Figure 4.23: An example of the original and generated spectrograms of the word "?asif" ("آسف") with AE

4.10.2.2 Quantitative evaluation

To evaluate our model, we compared it, first with Vanilla Autoencoder and second with a State-of-The-Art CNN, by measuring accuracy, precision, recall, and the F1 score as defined in the second chapter.

A. VAE Vs. Vanilla AE

Figure 4.24 shows the outcomes obtained from the proposed model and the AE-based Anomaly detection model during the test stage. The results demonstrate that AE's performance with complex data, like spectrograms, was very poor. The vanilla AE was unable to effectively capture the underlying structure of the data, which explains why its predictions were 53% wrong. Unlike AE, VAE learned the data's probabilistic distribution over the latent space instead of matching inputs to fixed points, which allowed it to attain 98% accuracy.

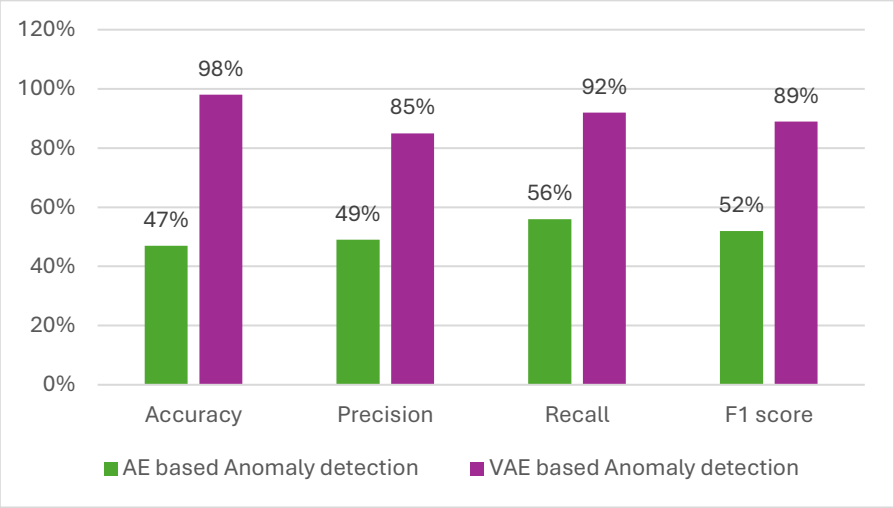


Figure 4.24: AE-based Anomaly detection Vs VAE-based Anomaly detection

B. VAE Vs. SOTA baseline

We also compared the results of our proposed VAE pronunciation error detection system with those provided by a supervised trained convolutional neural network. The dataset is inherently imbalanced in terms of classes, so the implemented CNN was also trained in this context.

The constructed CNN consists of three convolutional blocks, each comprising a Conv2D layer, a MaxPooling2D layer, and a BatchNormalization layer. These are followed by a Flatten layer and a Dense layer. To mitigate overfitting, a Dropout layer is included before the final output Dense layer. For binary classification, the output Dense layer uses a sigmoid activation function. The results achieved by the proposed model, as well as the baseline CNN during the test phase, are presented in the accompanying figure.

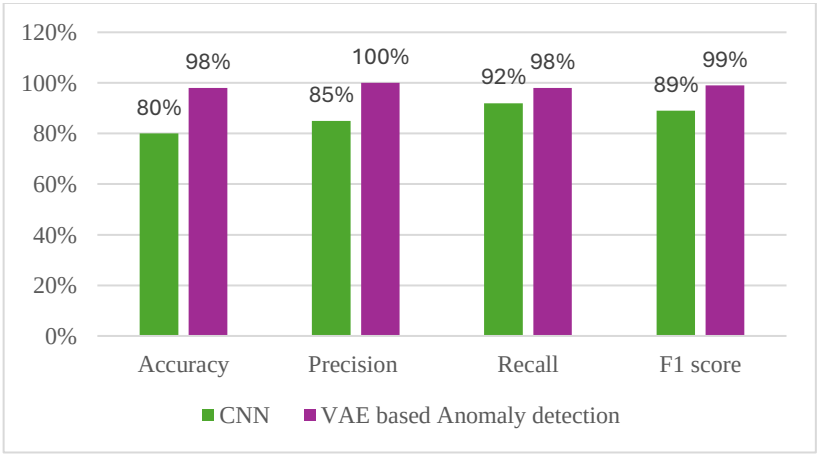


Figure 4.25: Comparison of the proposed method with SOTA CNN for binary classification

Figure 4.25 shows that the VAE-based AD approach outperforms the SOTA baseline CNN for all the considered measures. As we can see, there was an 18%, 15%, 6%, and 10% increase in accuracy, precision, recall, and F1-score, respectively.

The best achievement is seen in precision measure and the best improvement in accuracy. To better understand CNN's drawbacks, Figure 4.26 shows the confusion matrix of the CNN classification.

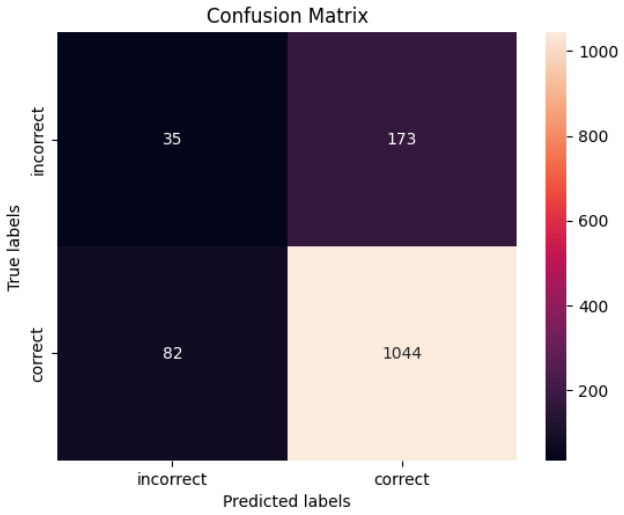


Figure 4.26: Confusion Matrix over the test data for CNN classifier

4.11 Chapter summary

In this chapter, we presented our contribution to the mispronunciation detection area relying to CAPT systems. The proposition considers the mispronunciation as an anomaly and thus addresses the mispronunciation detection problem as an anomaly detection task. For that purpose, we adopt the density-based approach which we implement using a variational autoencoder. To support our thesis, we compared its performances to those of Autoencoders (AEs) and Convolutional Neural Networks (CNNs).

The carried experiments over the largest available dedicated Arabic dataset confirmed the VAE's superiority in capturing meaningful latent representations that differentiate well-pronounced from mispronounced samples. Moreover, PCA visualizations of the latent space showed that the VAE's probabilistic nature enables it to encode structured and smooth distributions, improving error detection accuracy. On the other hand, the vanilla AE lacked this probabilistic constraint, leading to noisier latent spaces and poorer reconstructions, thereby limiting its effectiveness in distinguishing anomalies.

In additional experiments, the VAE-based proposed anomaly detection framework was compared to a standard CNN-based classification model. While CNN showed good classification performance, it relied tightly on labeled data and struggled to generalize to nuanced pronunciation errors. In contrast, the VAE leveraged unsupervised learning to detect subtle variations and anomalies, making it more adaptable to imbalanced or sparse datasets where mispronunciations are less frequent.

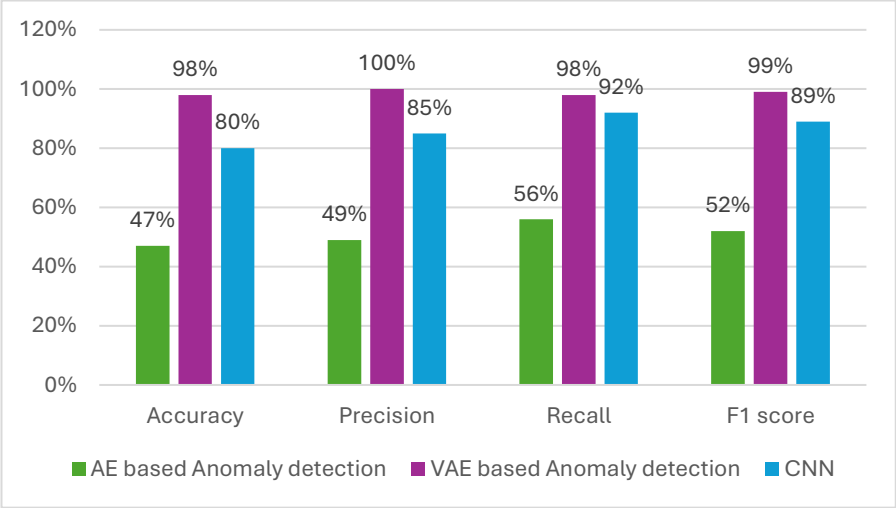


Figure 4.27: Comparison of the proposed method with CNN for binary classification

Additionally, further experiments showed that the VAE could reconstruct clean representations from its latent space, even for mispronounced samples, providing insights into the nature of the detected anomalies. The AE, however, failed to reconstruct meaningful outputs, often producing noisy spectrograms. These results highlight the VAE's effectiveness in learning robust representations for error detection.

In summary, this chapter underlines the potential of VAEs as a powerful tool for anomaly detection in speech contexts. Indeed, when compared to CNNs and vanilla AEs, the VAE showed more efficiency in learning latent representations of the available samples and consequently in identifying deviations, this highlights its added value in scenarios where detailed, unsupervised learning of complex patterns is required.

5 One-class classification and data augmentation for Arabic MDD

5.1 Introduction

Low-resource languages face significant challenges in the field of speech processing due to limited labeled datasets, class imbalance, and a lack of technological infrastructure. These obstacles hinder the development of robust pronunciation error detection systems, which are crucial for educational applications in CAPT systems. To address these limitations, innovative approaches are required to maximize the available data's utility while compensating for the scarcity of labeled examples.

This study investigates two alternative approaches to addressing these difficulties. First, we provide a One-Class Classification (OCC) model based on convolutional neural networks (CNNs), designed for contexts where mispronunciation data are uncommon. By focusing solely on well-pronounced utterances, the OCC model uses the "correct" class distribution to detect anomalies during testing, such as mispronunciations.

Second, we propose a supervised method based on Support Vector Machines (SVMs). Given the dataset's short size and class imbalance, we used data augmentation to create a larger and more diverse dataset. This augmentation improves the SVM's capacity to generalize and classify both well-pronounced and mispronounced utterances.

This study offers complementary solutions to the challenge of low-resource language pronunciation evaluation systems by addressing it using unsupervised OCC and SVM approaches. These approaches not only make better use of sparse data but also point to potential pathways for scalable and efficient language technology development.

5.2 Under-resources Languages:

“Under-resourced languages,” with synonyms for the same notion, are “low-density language”, “resource-poor languages”, “low-data languages”, and “less-resourced languages,” which denotes a language characterized by one or more of the following features [123]:

- Absence of a distinct writing system or consistent orthography
- Minimal online access
- Insufficient linguistic expertise
- A scarcity of electronic resources for speech and language processing, including L1 and L2 corpora, speech data transcriptions, pronunciation dictionaries, bilingual electronic dictionaries, and vocabulary lists.

It is crucial to distinguish that “under-resourced languages” differ from a "minority language,” which is a language utilized by a minority of the population inside a territory. Specific under-resourced languages serve as official languages within their respective countries and are spoken by substantial populations. Conversely, specific minority languages may be regarded as rather well-resourced, exemplified by the availability of the Catalan language in Google Search and Google Translate. Thus, under-resourced languages are not inherently endangered, although the converse is typically accurate.

5.3 Arabic as an under-resourced language

Although Arabic is the fifth most widely spoken language in the world, it is considered an “under-resourced language” for several reasons [124] [125]:

- *Dialectal Diversity:* Arabic is not a single language but a macro-language with many regional dialects (e.g., Egyptian, Levantine, Gulf, Maghreb). These dialects are often mutually unintelligible and do not have standardized orthographies, which complicates data collection and processing

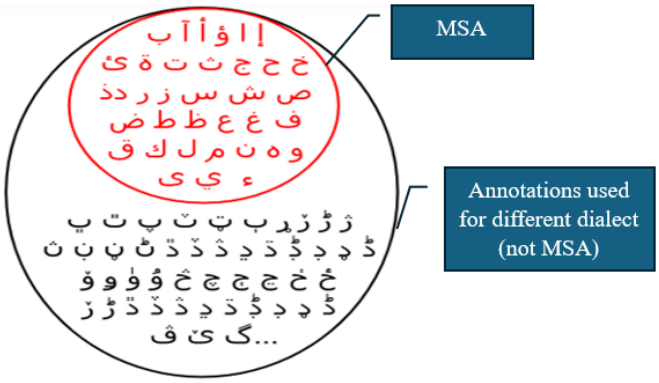


Figure 5.1: MSA Vs. Not MSA annotations

[124]

- *Limited Digitized Resources:* Modern Standard Arabic (MSA) has a more extensive digital presence than dialects, but overall, Arabic lacks the abundance of curated corpora, labeled datasets, and speech resources available for languages like English or Chinese.
- *Limited digital resources:* Modern Standard Arabic (MSA) has a more significant digital presence than dialects. However, Arabic lacks the abundance of corpora, labeled datasets, and speech resources available for languages such as English or Chinese.
- *Bias favoring English and high-resource languages:* Many language technology efforts prioritize English and other high-resource languages due to economic and research incentives. As a result, Arabic NLP tools and resources lag behind.
- *Minimal Annotation of Dialectal Data:* Existing Arabic datasets predominantly concentrate on official writings in Modern Standard Arabic or classical Arabic, ignoring informal dialectal information frequently utilized in social media and spoken interactions.

In this chapter, we propose two techniques to tackle the problem of imbalanced datasets in low-resource languages for mispronunciation detection. First, the One-class classification approach focuses on training models to recognize patterns based on limited available data without requiring negative samples. On the other hand, we propose data augmentation that generates a wide variety of synthetic data, which enriches the training set to increase the model's performance. When used independently, each strategy offers a way to address the limited availability of resources and successfully support language processing activities.

5.4 One-Class Classification

One-class classification (OCC) is a machine learning approach that is particularly suitable for low-resource languages, where only a limited amount of data from a single class is available. OCC concentrates on detecting anomalies that deviate from defined categories, thereby facilitating efficient classification with limited data. OCC includes one-class random forest (OCRF), one-class clustering-based ensemble (OCClustE), graph-based strategies, weighted loss functions, and pre-trained model refinement that can be applied in many low-resource language situations, including text classification and speech recognition. OCC can also be integrated with sophisticated domains such as cross-linguistic transfer learning and synthetic data generation [126].

5.4.1 Mechanism of One-Class Classification

The one-class classification (OCC) approach recognizes instances of a particular class and differentiates them from all other observations; the other observations are called outliers or anomalies. This approach is particularly suitable in scenarios where there is a large amount of data for the target class, while data for other classes is limited or nonexistent. The process of one-class classification is detailed as follows [127].

1. ***Training Phase:***

Unlike, conventional classification techniques that require instances from the involved classes during training, the OCC model is trained exclusively using data from the target class, referred to as the positive class. The main objective during training is to identify the fundamental characteristics and distribution of the target class. This goal is reached by approximating the density of probability of the target class or by drawing a decision boundary that encompasses it.

2. ***Modeling Techniques*** [128]***:***

a. Density-Based Methods: These techniques assess the density of the target class and classify instances that fall outside a specified density threshold as outliers. Kernel density estimation and Gaussian mixture models are frequently employed.
b. Distance-based methods: establish a distance metric to quantify the separation of new examples from the target class specifying those that surpass a specified distance as anomalies.
c. Reconstruction-Based Methods: Models like autoencoders are designed to recreate input data from the designated class. Instances that cannot be precisely rebuilt are identified as anomalies.
d. Boundary-Based Methods: These approaches establish a boundary containing the target class, with instances beyond this boundary categorized as outliers. One-Class Support Vector Machines (SVM) are a prevalent option in this domain.
e. Ensemble Methods: By integrating various one-class classifiers, ensemble methods can improve adaptability and performance, rendering them appropriate for intricate datasets.

3. ***Inference Phase:***

Once the OCC model is trained, the model can classify new samples as either part of the target class or outliers. This classification relies on the degree to which the new observations match the learned characteristics of the target class.

5.4.2 Convolutional Neural Networks (CNN) for One-Class Classification

Convolutional Neural Networks (CNNs) have become an incredible tool for One-Class Classification (OCC), especially in fields like image processing, where they effectively detect patterns and features [129] [130]. The CNN model acquires the ability to extract and express the fundamental features that define a positive class to reduce reconstruction errors or classification loss for such instances. While training, a decision boundary is created based on these features, facilitating the classification of additional instances. Upon completion of training, the CNN can classify new instances by establishing their affiliation with the target class or identifying them as anomalies. This is generally accomplished by assessing the compatibility of new inputs with the established feature space.

5.4.3 Benefits of Employing CNNs for OCC

- ***Efficient Feature learning:*** CNNs autonomously extract pertinent features from unprocessed data, minimizing the necessity for manual feature engineering.
- ***Scalability:*** Their hierarchical structure and parameter sharing enable effective control of big datasets.
- ***Flexibility:*** With suitable preprocessing techniques, CNNs can be modified for many data types beyond images, such as time series and textual data.

5.4.4 The use of CNN in MDD

Convolutional Neural Networks (CNNs) have become powerful tools in mispronunciation detection and diagnosis due to their proficiency in extracting and representing intricate features from data. These neural networks are proficient in processing grid-structured data such as spectrograms or Mel-frequency cepstral coefficients (MFCCs), which are typical representations of speech signals. Therefore, this ability makes them particularly useful for tasks associated with speech, covering feature extraction, classification, and constitution of end-to-end frameworks (covered in the second chapter).

Despite their successes in multiple areas, CNNs have yet to be utilized for one-class classification in mispronunciation detection, where the model could be trained solely on accurately spoken instances. This would allow the system to, on the one hand, identify anomalies in correct pronunciation during assessments, marking them as possible errors. On the other hand, it concentrates on depicting precise phonetic patterns, improving its generalization capacity across various speaker accents and tone differences.

5.4.5 Proposed Method

The increasing availability of unlabeled data and the difficulties associated with class imbalance in low-resource languages in labeled datasets have positioned deep learning for imbalanced classes as a significant area of research. This issue is especially important in the realm of mispronunciation detection. Datasets in this field are intrinsically imbalanced, as well-pronounced utterances significantly exceed mispronounced ones. This imbalance constitutes a considerable challenge to conventional supervised learning approaches, which often depend on balanced datasets to attain optimal efficacy [131].

To tackle this difficulty, we present a one-class classification algorithm developed for mispronunciation detection. Our methodology employs a convolutional neural network (CNN) architecture, trained using a semi-supervised approach. In contrast to traditional methods that necessitate balanced datasets with labeled samples from each class, our model is trained solely on good pronunciations ("class 1"). By focusing exclusively on this class during the training phase, the model acquires the inherent traits and patterns of the positive class.

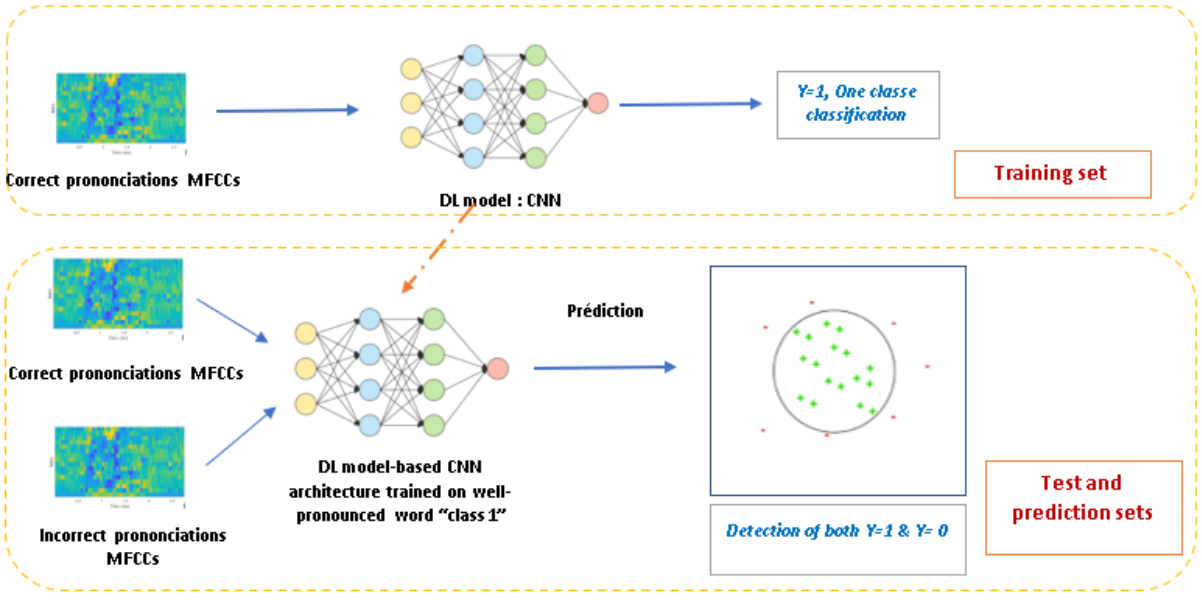


Figure 5.2: Overview of the proposed OCC-CNN classifier

During the testing phase, the trained model differentiates between well-pronounced and mispronounced utterances by recognizing divergences from the established patterns. This method is very effective in resolving the imbalance issue, as it removes the dependence on an important number of labeled mispronunciation instances. Figure 5.2 depicts the workflow of our proposed system, wherein the CNN extracts and learns features pertinent to good pronunciations, facilitating robust classification despite unseen mispronunciations during testing.

5.4.5.1 CNN architecture

In Figure 5.3, we illustrate the architecture of the proposed method. We implemented a Convolutional Neural Network (CNN) to classify and differentiate between well-pronounced and mispronounced words. In our approach, we implemented a CNN model that comprises five layers, including three convolutional layers and two fully connected layers. Each convolutional layer (Conv2D) is succeeded by Maxpooling and BatchNormalization layers. Flatten layers consolidate every feature map information into a single column. A dropout layer reduces overfitting by assigning random zero weights to a subset of the data. The final dense layer produces the model's output. The Rectified Linear Unit (ReLU) activation function enables the model to incorporate nonlinearity into the pattern. The activation function utilized in the final dense layer is the Sigmoid. The sigmoid function assigns a score between 0 and 1 to each output neuron, indicating the probability that the input observation is associated with the respective neuron class.

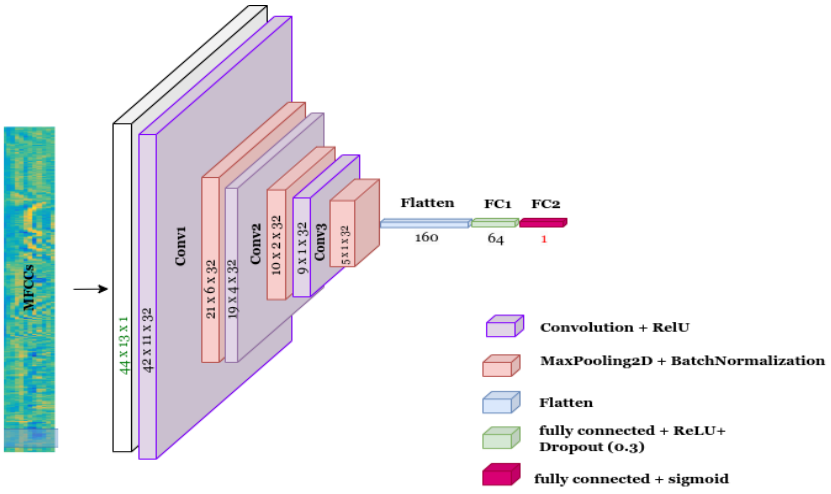


Figure 5.3: CNN Architecture for One-Class Classification

5.4.5.2 Preprocessing pipeline

Because Extracting fundamental characteristics of an audio signal is necessary for tasks like recognition, classification, or detection, relevant acoustic features from speech signals are crucial to developing an effective speech-processing system.

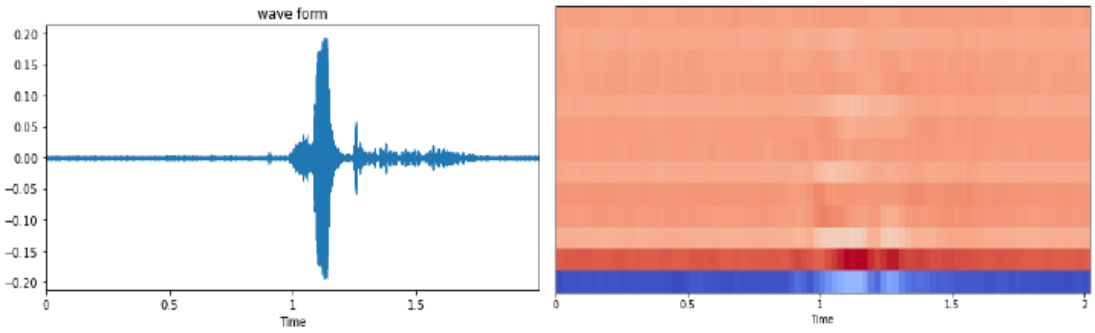


Figure 5.4: Waveform representation Vs. MFCCs representation of the Arabic word /ʃukran /

In this study, we selected MFCCs (Mel-frequency cepstrum coefficients) as the main acoustic features for their effectiveness in speech-processing applications. MFCCs are especially effective for pronunciation assessment tasks, as they accurately capture the nuanced frequency characteristics that differentiate correct pronunciations from errors [132]. Obtained through a series of transformations, MFCCs are applied to the speech signal. The process of their extraction is as follows:

- The short-term energy spectrum of the signal is obtained to capture the frequency content in short time intervals.
- The spectrum is then transformed into the Mel-frequency scale, which is a perceptual scale that reflects human pitch perception.
- The logarithm of the Mel-scaled spectrum is computed, followed by applying a cosine transform to generate the MFCCs.

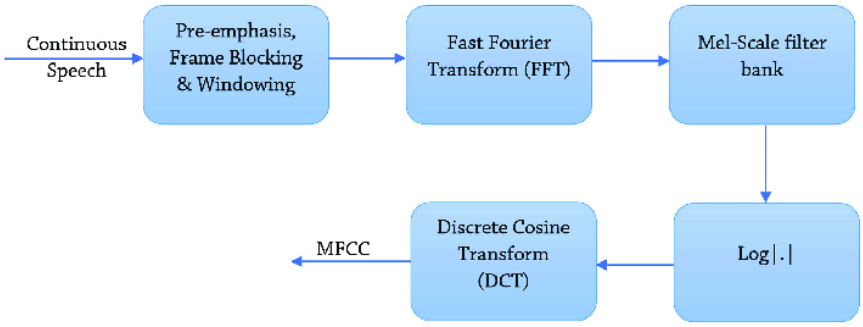


Figure 5.5: Process for MFCCs extraction

The outcome is a collection of coefficients that effectively define the signal's spectral envelope, which is an essential component of speech sounds.

5.4.5.3 Results and discussion

A CNN-based one-class classifier was built and fitted to a training dataset that only included examples from the normal class from the largest dataset of Arabic speech pronunciation errors in Egyptian dialogues, presented in Chapter IV, to assess the efficacy of the suggested approach. Once the model was ready, new instances were classified as either normal (well-pronounced) or not normal (mispronounced).

It was essential to perform a manual split to ascertain the number of samples in the training, validation, and test datasets because our dataset is totally imbalanced between the two classes, both overall and per word. Specifically, the test dataset comprises 480 recordings, 200 of which are well-pronounced and 280 of which are poorly pronounced.

Figure 5.6 depicts the model's performance during training, and Table 2 reports the outcomes on the test dataset in terms of DetAcc, FRR, and FAR.

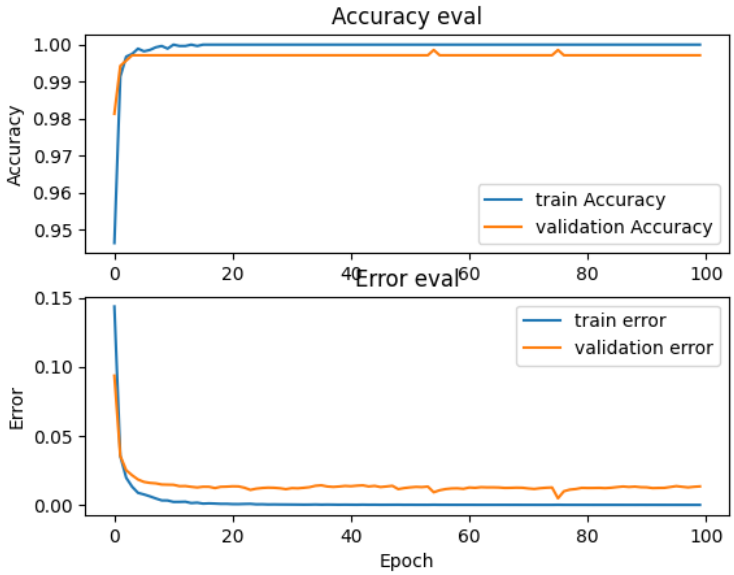


Figure 5.6: The training CNN performances

Table 5.1: Performances during the test stage

Corpus	False acceptance rate (FAR)	False Rejection Fate (FRR)	Det Accuracy
200 good pronunciations + 280 bad pronunciations	0%	39%	84%

Figure 5.7 presents the details in terms of the confusion matrix to better clarify the interpretation of these results.

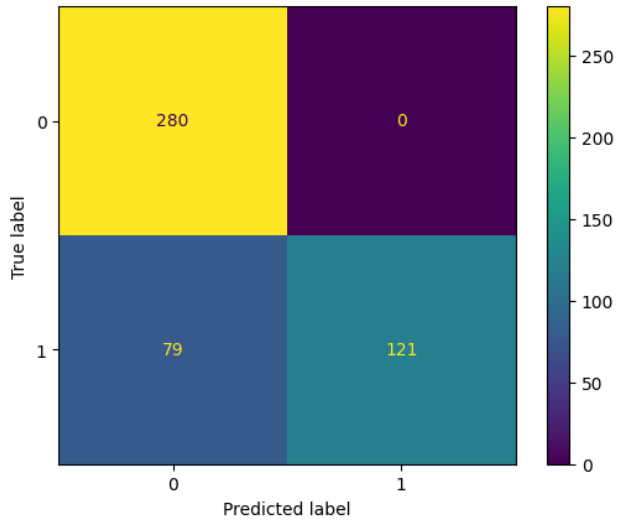


Figure 5.7: The confusion matrix on the test dataset

We use the terms الحقيقة ,الوقت, and بالضبط to demonstrate the mispronunciation detection within the dataset for examination. These words were intentionally chosen to characterize the segmental faults made by the speakers. These mistakes reflect spontaneous substitutions due to the influence of the speakers' Egyptian dialect. Table 5.2 summarizes the various speakers' alterations to the three words.

Table 5.2: Substitution errors of Arabic letters are illustrated with the use of IPA symbols for the transcription

Id	Word in Arabic letters	Det Accuracy	Substitution's type
5	الوقت	88.24%	Native Egyptian pronunciation/ʔ/ instead of /q/
18	الحقيقة	80.00%	Native pronunciation /ʔ/ instead of /q/
50	بالضبط	76,67%	Native pronunciation /z/ instead of / dˤ/

To address the challenge of pronunciation assessment in Arabic, particularly in the context of imbalanced datasets where accurately pronounced utterances significantly exceed inaccurately pronounced ones, we proposed a one-class classification convolutional neural network (OCC-CNN) trained in a semi-supervised manner to identify pronunciation deviations.

When experimenting on the ASMDD dataset, the results underline the OCC-CNN's efficacy in mispronunciation detection by demonstrating its ability to distinguish between correct and incorrect pronunciations. This study highlights the significance of employing one-

class classification algorithms to address the challenges posed by data imbalance in pronunciation assessment.

5.5 Data augmentation for imbalanced datasets in MDD

In deep learning architectures, to perform supervised tasks based on observed data, the need for sufficient and qualitative labeled data is the most important factor that determines the performance of the machine learning model [133]. The collection and annotation processes require time, effort, and money. An effective way to overcome data scarcity is data augmentation. The concept of augmenting data is (i). To increase the amount of existing data using different techniques to generate artificial data (ii). Add them to the original ones to have a more data-intensive training dataset by considering label-preserving [134].

In several situations, scholars give priority to corpus development to guarantee the accuracy and the quality of their deep learning architectures in order to increase their generalization performance [135]. This has led to evolve more complicated architectures between 2015-2017 [136], [134] as AlexNet [137], VGG-16 [85], ResNet [138], Inception-V3 [139] and DenseNet [140]. Since then, data augmentation were widely used in different domains; Time series[141], videos [142] images [143] [136] [144], Text classification [135], anomaly detection [145], and also sound augmentation [146] [147]

It is essential to recognize the lack of non-native labeled corpora, particularly for low-resource languages in CAPT, such as Arabic. In this case, it is more probable to encounter well-pronounced speech rather than deviant speech, resulting in the issue of class imbalance; a dataset is considered imbalanced if the grouped categories are unequally represented. In this section, we propose a study employing speech augmentation techniques to address the deficiency of dedicated nonnative speech corpora and the imbalanced dataset [148]. Thus, data augmentation is a method that enhances the dataset by generating synthetic data from the training data [149]. Its objective is to extensively cover the problem space by augmenting the data and producing additional samples from the original dataset, thereby improving the model's generalization [150].

5.5.1 Data Augmentation for audio

Computer vision and image processing extensively utilize data augmentation, but regrettably, its application in audio data remains limited. [151] pioneered the use of DA for speech recognition, enhancing the speech dataset by transforming spectrograms and generating a

random warp factor for each utterance during training. [152] proposed two data augmentation schemes: semi-supervised training and vocal tract length perturbation. In 2015 [153], focusing on increasing speaker and speech variations, the authors proposed two data augmentation approaches, vocal tract length perturbation (VTLP) and stochastic feature mapping (SFM), in their article “Data Augmentation for Deep Neural Network Acoustic Modeling”. [154] proved that by using data augmentation techniques, the recognition of Latin American and Asian accented speech was significantly improved. In 2019, [155] applied pitch shift, time stretch, and time shift to implement DNN-based real-time voice conversion. In 2021, [156] recently tested three augmentation protocols for audio. Standard Signal Augmentation relies on the MATLAB built-in data augmentation methods for audio signals. Spectrogram Augmentation produces six transformed versions of each original spectrogram, and Signal Augmentation works directly on the raw audio signals.

5.5.2 Data augmentation for MDD

In their work, [68] attempted to improve Arabic audio dataset via modifying their private. The authors augmented 20 samples of each alphabet by employing a pitch variation factor to preserve originality and reduce pronunciation effects. By reviewing the audio recordings audibly, they judged that this technique was appropriate and had no negative impact on the Arabic audio dataset. They acquired six modified recordings from each letter by changing pitch levels between -0.3 and 0.3.

The research on [67] employed noise injection, time-shifting, and audio speed alteration through time stretching to gradually enhance the audio instances in their proprietary dataset, hence improving the validation accuracy of the fundamental model. The first method they used is near noise injection, which incorporates white noise as a proportion of the signal to noise, which is appropriate for their model since user input is typically not devoid of noise. Random noise augmentation was implemented by incorporating two noise values, X = 0.005 and X = 0.0005, into audio files utilizing the NumPy module in Python. The second DA method used by authors is time-shifting, this method advances the audio either forward or backward with the roll method in Python's Numpy module. The roll method displaces the commencement of an audio file by S milliseconds, advancing the audio file by 2 milliseconds at the graph's outset, substituting it with silence. in the third method, changing the speed modifies the audio stream S at a specified rate R, with R values of [1.25, 1.4, 1.5, 1.6]. The study also investigates the variation of short vowels on shorter and longer durations.

5.5.3 Online and offline augmentation

After dividing the dataset into training, testing, and validation sets, only the training set should be subjected to data augmentation. This procedure can be executed via offline or online augmentation [136].

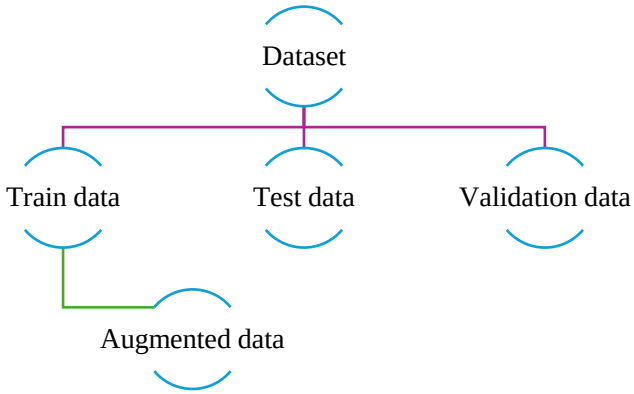


Figure 5.8: Splitting the dataset to augment only the train data

In offline augmentation, transformations are pre-calculated prior to training. This method provides the benefit of conserving computing resources over time while ensuring a distinct separation between augmentation and model code. Nonetheless, offline augmentation is generally executed on the CPU, which is less efficient than GPU processing, and necessitates supplementary storage for the augmented data.

Conversely, online augmentation implements transformations dynamically during training by applying deep learning libraries. Thus, it leverages accelerated GPU processing and facilitates seamless integration with training operations, simplifying implementation. Nonetheless, it may require significant computer resources during long training sessions and tightly integrates the augmentation process with the model code.

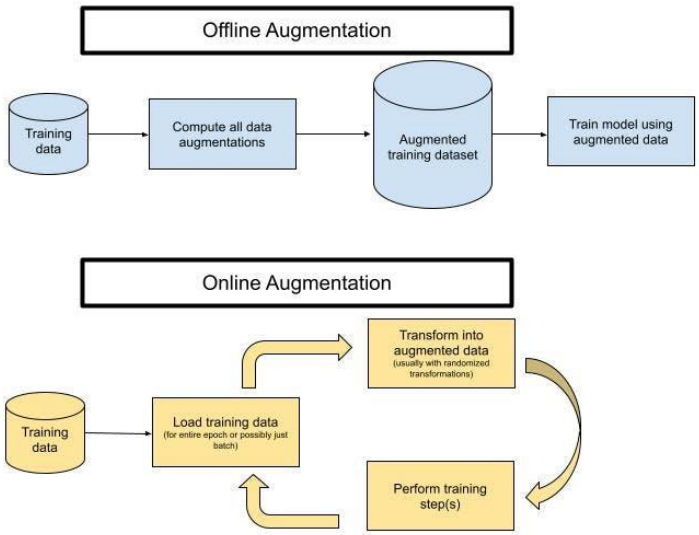


Figure 5.9: Online Vs. Offline DA

The fundamental principle of data augmentation is that the enhanced samples must stay "credible," preserving realistic and significant attributes to enhance the model's performance effectively.

5.5.4 Data augmentations techniques

The growing number of sound classification studies based on data augmentation methods shows the importance of these techniques in different domains [146], such as speech recognition[157], music classification [158], environmental sound classification [159]…etc. The choice of features directly affects the quality of the generated data. Thus, audio augmentation can be applied to two audio representations: 1) Raw audio in waveform representation and 2) frequency representation of audio, namely, Spectrogram [156] [146].

5.5.4.1 Spectrogram augmentation

Data augmentation based on spectrograms is a helpful trend that improves the performances of DL models as it considerably increases the amount of training data, particularly in speech and audio processing. Some of the prevalent techniques are presented below:

- *Time Masking and Frequency Masking:* These techniques entail hiding a random segment of time or frequency bands inside the spectrogram. It enhances model robustness, forcing it to learn representations that are less reliant on certain spectral properties.
- *Time Warping:* This method modifies the spectrogram by extending or compressing it along the temporal axis, so replicating variations in speech rate or tempo.

- *Adding Noise:* Random noise may be superimposed onto the spectrogram, simulating background noise. This enhances the model's resilience to noisy real-world situations.
- *Shifting:* Displacing the spectrogram along the temporal axis emulates misalignment or latency in the audio input.

5.5.4.2 Waveform augmentation

The performances of DL models and their adaptability are highly boosted by the use of audio data augmentation techniques. Notable methods for enhancing raw audio waveforms comprise transformations in the time domain and noise injection. We can notice some prominent techniques for augmenting raw audio waveforms as illustrated bellow:

- *Time-stretching:* Modifies the audio tempo without affecting its pitch [160].
- *Pitch-shifting:* Modifies the pitch of the audio while maintaining its duration [160].
- *Random cropping or padding:* Adjusts the temporal duration of audio snippets to standardize or enhance samples.
- *Dynamic range compression:* Implements effects to augment or diminish audio dynamics.
- *Reverberation and echo:* Incorporates spatial sound effects such as reverbs to replicate various surroundings.
- *Incorporating background noise:* such as white noise or environmental sounds, to replicate situations in the real world. Methods encompass the incorporation of Gaussian noise and the addition of pink or brown noise.

5.5.5 Proposed Method

As depicted in the figure below, this section outlines a supervised approach to detect pronunciation errors, focusing on the use of Support Vector Machines (SVM) as a classification technique. SVM was selected for its proven efficacy in handling various classification tasks. To enhance the model's generalization capability, and due to the dataset's limited size, we performed an offline data augmentation procedure implemented on the training set to augment the dataset by incorporating synthetic variations. The increased dataset was utilized to train three distinct SVM kernels: linear, polynomial, and Gaussian, so a comparative study was established to assess their accuracy in identifying mispronunciations.

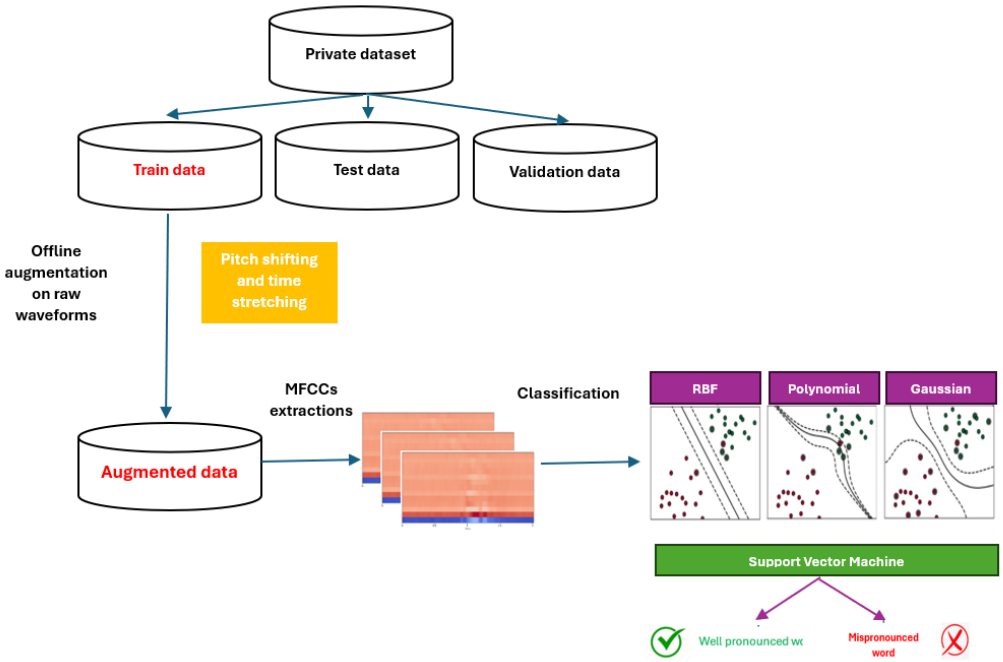


Figure 5.10: System outline

5.5.5.1 Support Vector Machine (SVM)

Support vector machines appeared in the mid-1990s [161]. They are based on the intuitive concept of maximizing the margin of separation between two rival classes, which is the distance between the decision hyperplane and the support vector closest to the training instances. Given two classes of instances, the SVM aims to create a hyperplane separating the data while maximizing the distance between the two classes. When the training data is linearly separable, SVMs provide linear separation, ensuring that all examples in the training set are accurately classified. Otherwise, data are translated to a new, higher-dimensional space, allowing them to be separated by a hyperplane; this transformation requires a kernel function. Common kernel functions include linear, polynomial, and Gaussian. This allows SVM to attain excellent accuracy in high-dimensional spaces while reducing computing complexity. When SVM is trained on big datasets, it achieves excellent accuracy on small datasets and is resilient to noise and outliers.

This study evaluated three kernels: linear, polynomial, and radial basis function (RBF). The linear kernel is mostly utilized in text classification, but the polynomial kernel is ideal for image processing. The RBF is a multipurpose kernel. However, a large number of labelled data is necessary to train an SVM model, and in CAPT applications, the lack of specialized corpora is a critical issue.

5.5.5.2 Private Arabic dataset

This study's dataset comprises both "correct" and "wrong" non-artificial pronunciations [62]. The pronunciations are from nine students aged five to eight years old, each of them recited a set of sixteen sequences (words or groupings of syllables). The chosen words present certain obstacles for learners, such as lengthy vowels and words written with more than one connected component. The sequences are not excessively long and do not contain unusual words (Table 5.3).

Table 5.3: List of the considered words in the private Arabic dataset

#	Sequences in Arabic	Phonetic transcription	Translation	#	Sequences in Arabic	Phonetic transcription	Translation
1	صباح الخير	s`aba:ħu ʔalxajr	Good Morning	9	مسّن	Mussin	Aged
2	إلى اللقاء	ʔila ʔalliqa:ʔ	Good bye	10	متأخر	mutaʔaxir	Late
3	ليلة سعيده	lajlatun saʕi:datum	Happy Night	11	فارغ	fa:riɣ	Empty
4	من فضلك	min fad`lik	Please	12	ثقيل	θaqi:l	Heavy
5	شكرا	ʃukran	Thanks	13	أسفل	ʔasfal	Down
6	جميل	ʤami:l	Beautiful	14	داخل	da:xil	Inside
7	قبيح	qabi: ħ	Ugly	15	بداخل	bida:xil	Inside of
8	قريب	qari:b	Near (close)	16	خارج	xa:riʤ	Outside

Figure 5.11 represents the distribution of the samples based on the available speech sequences. It demonstrates the disparity between the two classes (well-pronounced versus mispronounced).

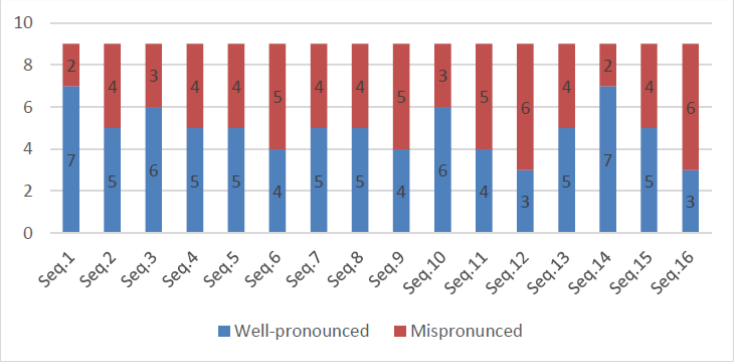


Figure 5.11: Distribution of samples for the both classes over the dataset

5.5.5.3 Speech augmentation

To address restrictions caused by the lack of ground truth samples, including both "good" and "bad" pronunciations, pitch shifting and time stretching audio augmentation techniques were used to artificially and accurately enlarge the training dataset. In the present case, the new samples help us to rebalance the dataset, as illustrated in Figure 5.12.

Figure 5.12: Distribution of the samples in training / test datasets

5.5.5.4 Results and Discussion

Accuracy is an important metric for evaluating classification tasks when leveraging data augmentation. It assesses the proportion of valid predictions across all samples. First, we trained the SVM on the ground-truth training dataset; Table 5.4 shows the results according to the three kernels.

Table 5.4: Results of the detection with the initial training data set

Accuracy	**Linear kernel**	**Polynomial kernel**	**RBF kernel**
Traning datset	100%	81%	80%
Test dataset	48%	58%	58%

Figure 5.13 depicts the confusion matrices for the three kernels across the test samples. It is clear that the mispronounced samples are primarily classed as well-pronounced, which can be attributed to an imbalanced dataset in favor of well-pronounced samples. Thus, the SVM mis predicts minority class instances.

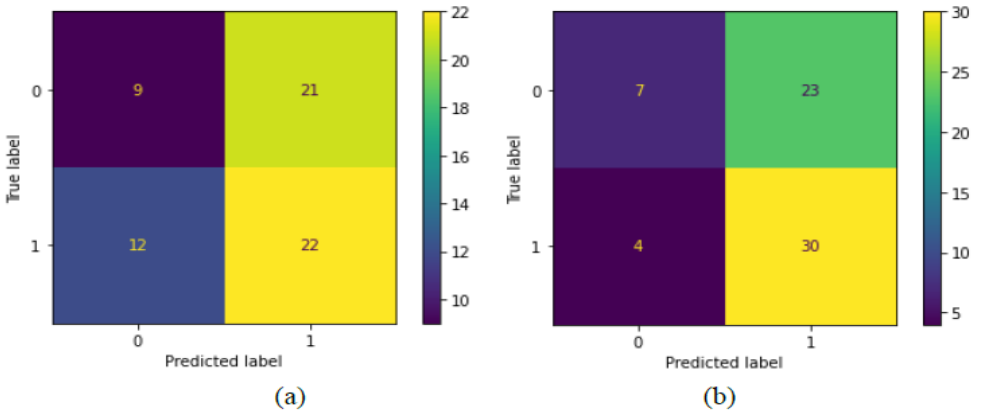


Figure 5.13: Confusion matrices over the test data for the (a) linear kernel and (b) the polynomial and RBF kernels

In the subsequent tests, we enhanced the training dataset by various audio augmentation approaches; Figure 5.14 presents the results based on the size of the training set.

Figure 5.14: Accuracy of the mispronunciation detection according to the training size

Figure 5.14 demonstrates a rise in accuracy for all kernels across the test dataset. Specifically, the linear kernel's accuracy increases from 48% to 97%. The subsequent table presents the false rejection rate throughout the test samples. It is evident that as the training dataset size increases, the false rejection rate (FRR) diminishes to zero.

Table 5.5: The false rejection rate on the test samples according to the training set size

Kernel	80 samples	194 samples	308 samples	650 samples
Linear	35%	0%	0%	0%
Polynomial	12%	6%	0%	0%
RBF	12%	0%	0%	0%

This study addresses the deficiency of CAPT-specific speech corpora and the associated class imbalance problem. The suggested approach is to augment the training dataset artificially. To evaluate the proposition, we examine the impact of data augmentation on classification performance in terms of accuracy and false rejection rate. Data augmentation enhances the training dataset by producing new data points and mitigating overfitting. The findings demonstrate the advantageous effects of speech augmentation strategies that improve accuracy and substantially decrease the false error rate.

5.6 Chapter Summary

In this chapter, we addressed two important issues in detecting pronunciation errors in low-resource languages: the lack of labeled data and the occurrence of class imbalance. First, we used a convolutional neural network (CNN) to implement a one-class classification (OCC) technique. This model was trained only on well-pronounced utterances so it can detect mispronunciations during testing. The OCC technique demonstrated its usefulness as a solution customized to settings where mispronunciation examples are limited or unavailable, emphasizing its potential for practical applications.

We also explored data augmentation techniques to address dataset scarcity. Offline augmentation methods were employed to increase the training data, as a results, various and credible variants of the available samples were created. These augmentation procedures considerably improved SVM model performance by boosting the resilience and generalization capacities of the trained RBF, Polynomial, and Gaussian classifiers.

6 Conclusion and future work

6.1 Summary of Contributions

This dissertation has explored the challenge of mispronunciation detection for Arabic within the context of Computer-Assisted Pronunciation Training (CAPT). Addressing this issue is particularly critical due to the scarcity of balanced, labeled datasets in this field. To overcome these constraints, we proposed solutions rooted in Deep Learning methodologies, with a specific focus on generative that present the central model of our proposed solution due to their ability to learn meaningful latent representations in an unsupervised manner.

In Chapter 2, we conducted a comprehensive review of mispronunciation detection and diagnosis using deep learning models. This review highlighted existing methods and their respective limitations, particularly concerning data dependency. In Chapter 3**,** We explored deep learning approaches by categorizing them into discriminative and generative models. The two subsequent chapters provide our contributions in this field, the main one was in the chapter 4.

6.2 Proposed Approaches

6.2.1 Variational autoencoder for anomaly detection

In Chapter 4, we presented our primary contribution: the use of VAEs for mispronunciation detection. This approach framed the problem as an anomaly detection task. The VAE was trained exclusively on correctly pronounced utterances, learning to encode their latent representations. Mispronounced utterances were detected as deviations from this learned distribution. This technique proved effective, especially considering our imbalanced datasets.

The strength of the VAE lies in its capacity to reconstruct correct inputs while producing higher reconstruction errors for outliers. This characteristic allowed us to sidestep the need for extensive labeled corpora, making the method particularly suitable for low-resource languages such as Arabic.

6.2.2 One-Class CNN for Pronunciation Error Detection

In the fourth subtitle of the Chapter 5 introduced a discriminative deep learning model using a one-class Convolutional Neural Network (CNN). This model was designed to extract robust speech features while performing direct classification. The CNN-based approach performs

explicit classification. Despite working with an imbalanced dataset, the one-class CNN yielded promising results due to its capacity for feature extraction and adaptability to limited data.

6.2.3 Data Augmentation for Supervised Learning

Given the persistent issue of data sparsity, we explored audio data augmentation techniques in the fifth subtitle of the Chapter 6. By augmenting the existing dataset with synthetic variations, we expanded the training set and rebalanced its classes. These techniques allowed us to train a Support Vector Machine (SVM) as a binary classifier. The augmented dataset significantly improved the SVM's performance, validating the effectiveness of this method even with limited original data.

The results from our experiments demonstrated that generative models in AD approach, discriminative classifiers in OCC approach, and data augmentation techniques can address core challenges in CAPT for under-resourced languages like Arabic. Each method contributed uniquely: VAEs excelled in unsupervised detection, CNNs enhanced feature-based classification, and data augmentation strengthened supervised learning models.

6.3 Implications and Future Work

6.3.1 At backend level

Looking forward, future research could explore advanced generative models such as Generative Adversarial Networks (GANs) and diffusion models. These models could create more realistic synthetic audio samples, further addressing data limitations. Additionally, self-supervised learning approaches could be incorporated to leverage large unlabeled speech corpora, enabling more robust feature extraction without extensive manual labeling.

The integration of multimodal learning, combining audio with visual articulatory data, also presents a promising direction. This could enhance pronunciation detection by providing complementary cues, especially in complex speech contexts.

6.3.2 At frontend level

Computer-Assisted Pronunciation Training (CAPT) applications require effective feedback mechanisms. potential research directions to enhance feedback mechanisms may include, multimodal feedback integration, explainable feedback, real-time feedback systems, generative feedback models, multilingual and cross-language transfer, and gamification and engagement tools.

Adaptive algorithms can adjust feedback based on learners' progress, while multimodal feedback integrates visual aids and tactile devices. Explainable feedback incorporates linguistic insights and pronunciation guides. Real-time feedback systems provide contextual feedback during live conversations or interactive dialogues. Gamification and engagement tools reward correct pronunciation and persistent improvement.

In conclusion, our work demonstrates that addressing data sparsity and imbalance through innovative deep learning models is a viable pathway toward advancing CAPT systems for low-resource languages. By bridging generative and discriminative methods and leveraging data augmentation, we have laid a strong foundation for future research in this evolving field.

7 References

[1] Dionisotti, A. C. (1982). From Ausonius' schooldays? A schoolbook and its relatives. *The Journal of Roman Studies*, *72*, 83-125.

[2] Korhonen, K. (1996). On the composition of the Hermeneumata Language Manuals. *Arctos–Acta Philologica Fennica*, *30*, 101-119.

[3] Bravo-Agapito, J., Bonilla, C. F., & Seoane, I. (2020). Data mining in foreign language learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *10*(1), e1287.

[4] Bitzer, D., Braunfeld, P., & Lichtenberger, W. (1961). PLATO: An automatic teaching device. *IRE Transactions on Education*, *4*(4), 157-161.

[5] Kalikow, D., & Swets, J. (1972). Experiments with computer-controlled displays in second-language learning. *IEEE Transactions on Audio and Electroacoustics*, *20*(1), 23-28.

[6] Ghashghaei, T. B., Kashefian-Naeeini, S., & Marzban, A. (2020). The Relationship between the Use of Computer Assisted Language Learning (CALL) and Computer Knowledge and Facilities: A Mixed-Method Study. *International Journal of Multicultural and Multireligious Understanding*, *7*(7), 323-343.

[7] Gillespie, J. (2020). CALL research: Where are we now?. *ReCALL*, *32*(2), 127-144. Doi: 10.1017/S0958344020000051.

[8] Chen, X. L., Zou, D., Xie, H. R., & Su, F. (2021). Twenty-five years of computer-assisted language learning: A topic modeling analysis. *Language Learning & Technology*, Vol. 25(3), 151–185. Doi: http://hdl.handle.net/10125/73454.

[9] Lee, K. W. (2000). English Teachers' Barriers To The Use Of Computer-Assisted Language Learning. *The Internet TESL Journal*. Vol *6*(12), 1-8.

[10] Jeong, K. O. (2022). Facilitating sustainable self-directed learning experience with the use of mobile-assisted language learning. *Sustainability*, *14*(5), 2894. Doi : https://doi.org/10.3390/su14052894.

[11] Bax, S. (2003). CALL—past, present and future. *System*, Vol *31*(1), 13-28. Doi : https://doi.org/10.1016/S0346-251X(02)00071-4.

[12] Godwin-Jones, R. (2019). Riding the digital wilds: Learner autonomy and informal language learning. *Language Learning & Technology, 23*(1), 8–25.

[13] Rogerson-Revell, P. M. (2021). Computer-assisted pronunciation training (CAPT): Current issues and future directions. *Relc Journal*, *52*(1), 189-205. Doi: 10.1177/0033688220977406.

[14] Lounis, M., Dendani, B., & Bahi, H. (2024). Mispronunciation detection and diagnosis using deep neural networks: a systematic review. *Multimedia Tools and Applications*, 1-35. Doi: 10.1007/s11042-023-17899-x.

[15] Godwin-Jones, R. (2021). Big data and language learning: Opportunities and challenges. *Language Learning & Technology*, 25(1), 4–19. http://hdl.handle.net/10125/44747.

[16] Qiu, J., Wu, Q., Ding, G., Xu, Y., & Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, Vol *2016*, 1-16. Doi: 10.1186/s13634-016-0355-x.

[17] Miao, X., & Wang, P. (2023). A literature review on factors affecting motivation for learning Arabic as a foreign language. *Open Journal of Social Sciences*, *11*(6), 203-211. doi: 10.4236/jss.2023.116014.

[18] Cengiz, B. C. (2023). Computer-assisted pronunciation teaching: An analysis of empirical research. *Participatory Educational Research*, *10*(3), 72-88. Doi: 10.17275/per.23.45.10.3.

[19] Harrison, A. M., Lo, W. K., Qian, X., & Meng, H. (2009). Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training. In *SLaTE* (pp. 45-48).

[20] Lee, A. (2016). Language-independent methods for computer-assisted pronunciation training (Doctoral dissertation, Massachusetts Institute of Technology).

[21] Shahin, M., & Ahmed, B. (2019). Anomaly detection based pronunciation verification approach using speech attribute features. *Speech Communication*, *111*, 29-43. Doi : 10.1016/j.specom.2019.06.003.

[22] Xu, Y. (2022). English speech recognition and evaluation of pronunciation quality using deep learning. *Mobile Information Systems*, *2022*(1), 7186375. Doi: 10.1155/2022/7186375.

[23] Witt, S. M., & Young, S. J. (1997). Language learning based on non-native speech recognition. In *Eurospeech* (pp. 633-636). Doi: 10.21437/Eurospeech.1997-227.

[24] Kheir, Y. E., Ali, A., & Chowdhury, S. A. (2023). Automatic Pronunciation Assessment--A Review. in *Findings of the Association for Computational Linguistics: EMNLP 2023*, p. 8304-8324. Doi: 10.18653/v1/2023.findings-emnlp.557.

[25] Harrison, A. M., Lo, W. K., Qian, X., & Meng, H. (2009). Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training. In *SLaTE* (pp. 45-48).

[26] Li, K., Qian, X., & Meng, H. (2016). Mispronunciation detection and diagnosis in l2 english speech using multidistribution deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *25*(1), 193-207. Doi: 10.1109/TASLP.2016.2621675.

[27] Truong, K. P., Neri, A., Cucchiarini, C., & Strik, H. (2004). Automatic pronunciation error detection: an acoustic-phonetic approach. in InSTIL/ICALL Symposium 2004.

[28] Franco, H., Ferrer, L., & Bratt, H. (2014). Adaptive and discriminative modeling for improved mispronunciation detection. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 7709-7713). IEEE. Doi: 10.1109/ICASSP.2014.6855100.

[29] Necibi, K., Frihia, H., & Bahi, H. (2015). On the use of decision trees for arabic pronunciation assessment. In *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication* (pp. 1-6). Doi: 10.1145/2816839.2816866.

[30] Ryu, H., & Chung, M. (2017). Mispronunciation Diagnosis of L2 English at Articulatory Level Using Articulatory Goodness-Of-Pronunciation Features. In *SLaTE* (pp. 65-70). Doi: 10.21437/SLaTE.2017-12.

[31] Li, W., Li, K., Siniscalchi, S. M., Chen, N. F., & Lee, C. H. (2016). Detecting Mispronunciations of L2 Learners and Providing Corrective Feedback Using Knowledge-Guided and Data-Driven Decision Trees. In *Interspeech* (Vol. 2016, pp. 3127-3131). Doi: 10.21437/Interspeech.2016-517.

[32] Duan, R., Kawahara, T., Dantsuji, M., & Nanjo, H. (2017). Transfer Learning based Non-native Acoustic Modeling for Pronunciation Error Detection. In *SLaTE* (pp. 42-46). Doi: 10.21437/SLaTE.2017-8.

[33] Duan, R., Kawahara, T., Dantsuji, M., & Zhang, J. (2017). Effective articulatory modeling for pronunciation error detection of L2 learner without non-native training data. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5815-5819). IEEE. Doi: 10.1109/ICASSP.2017.7953271.

[34] Li, W., Chen, N. F., Siniscalchi, S. M., & Lee, C. H. (2017). Improving Mispronunciation Detection for Non-Native Learners with Multisource Information and LSTM-Based Deep Models. In *Interspeech* (pp. 2759-2763). Doi: 10.21437/Interspeech.2017-464.

[35] Li, K., Wu, X., & Meng, H. (2017). Intonation classification for L2 English speech using multi-distribution deep neural networks. *Computer Speech & Language*, *43*, 18-33. Doi: 10.1016/j.csl.2016.11.006.

[36] Ye, W., Mao, S., Soong, F., Wu, W., Xia, Y., Tien, J., & Wu, Z. (2022). An approach to mispronunciation detection and diagnosis with Acoustic, Phonetic and Linguistic (APL) embeddings. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6827-6831). IEEE.

[37] Wu, Y., Zhang, J., & Dong, Q. (2019). The use of SDAE in noisy English mispronunciation detection and diagnosis towards application in mobile learning. In *Proceedings of the 2019 International Symposium on Signal Processing Systems* (pp. 176-180). Doi: 10.1145/3364908.3365302.

[38] Yan, B. C., Wang, H. W., & Chen, B. (2023). Peppanet: Effective mispronunciation detection and diagnosis leveraging phonetic, phonological, and acoustic cues. In *2022 IEEE Spoken Language Technology Workshop (SLT)* (pp. 1045-1051). IEEE. Doi: 10.1109/SLT54892.2023.10022472.

[39] Akhtar, S., Hussain, F., Raja, F. R., Ehatisham-ul-haq, M., Baloch, N. K., Ishmanov, F., & Zikria, Y. B. (2020). Improving mispronunciation detection of arabic words for non-native learners using deep convolutional neural network features. *Electronics*, *9*(6), 963. Doi: 10.3390/electronics9060963.

[40] Shahin, M., Epps, J., & Ahmed, B. (2023). Phonological Level wav2vec2-based Mispronunciation Detection and Diagnosis Method. *arXiv preprint arXiv:2311.07037*.

[41] Zhang, D. Y., Saha, S., & Campbell, S. (2023). Phonetic RNN-transducer for mispronunciation diagnosis. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1-5). IEEE. Doi: 10.1109/ICASSP49357.2023.10094945.

[43] Guo, S., Kadeer, Z., Wumaier, A., Wang, L., & Fan, C. (2023). Multi-Feature and Multi-Modal Mispronunciation Detection and Diagnosis Method Based on the Squeezeformer Encoder. *IEEE Access*, *11*, 66245-66256. Doi: 10.1109/ACCESS.2023.3278837.

[44] Xu, X., Kang, Y., Cao, S., Lin, B., & Ma, L. (2021, August). Explore wav2vec 2.0 for Mispronunciation Detection. In *Interspeech* (pp. 4428-4432). Doi: 10.21437/Interspeech.2021-777.

[45] Yang, M., Hirschi, K., Looney, S. D., Kang, O., & Hansen, J. H. (2022). Improving mispronunciation detection with wav2vec2-based momentum pseudo-labeling for accentedness and intelligibility assessment. *arXiv preprint.* Doi: https://doi.org/10.48550/arXiv.2203.15937

[46] Pennington, M. C., Rogerson-Revell, P., Pennington, M. C., & Rogerson-Revell, P. (2019). Using technology for pronunciation teaching, learning, and assessment. *English Pronunciation Teaching and Research: Contemporary Perspectives*, 235-286. Doi: 10.1057/978-1-137-47677-7_5.

[47] Bahi, H., Dendani, B., & Lounis, M. (2024). Automatic Pronunciation Assessment and Feedback for Arabic Learners: A Review. *International Journal of Asian Language Processing*. Doi: 10.1142/S2717554524300019.

[48] Agarwal, C., & Chakraborty, P. (2019). A review of tools and techniques for computer aided pronunciation training (CAPT) in English. *Education and Information Technologies*, *24*(6), 3731-3743. Doi: 10.1007/s10639-019-09955-7.

[49] Alsabaan, M. (2015). Pronunciation support for Arabic learners. The University of Manchester (United Kingdom).

[50] Tejedor-García, C., Escudero-Mancebo, D., Cardeñoso-Payo, V., & González-Ferreras, C. (2020). Using challenges to enhance a learning game for pronunciation training of English as a second language. *IEEE Access*, *8*, 74250-74266. Doi: 10.1109/ACCESS.2020.2988406.

[51] Erizara, B. V., & Wijirahayu, S. (2024). The Exploration of Duolingo Application for Vocabulary Building and Pronunciation of Pre-Service Teachers. *Scripta: English Department Journal*, *11*(1), 95-105. Doi: 10.37729/scripta.v11i1.5081.

[52] Nazir, F., Majeed, M. N., Ghazanfar, M. A., & Maqsood, M. (2019). Mispronunciation detection using deep convolutional neural network features and transfer learning-based model for Arabic phonemes. *IEEE Access*, *7*, 52589-52608. Doi: 10.1109/ACCESS.2019.2912648.

[53] Touchie, H. Y. (1986). Second language learning errors: Their types, causes, and treatment. *JALT journal*, *8*(1), 75-80

[54] Meng, H., Lo, Y. Y., Wang, L., & Lau, W. Y. (2007). Deriving salient learners' mispronunciations from cross-language phonological comparisons. In *2007 IEEE*

Workshop on Automatic Speech Recognition & Understanding (ASRU) (pp. 437-442). IEEE. Doi: 10.1109/ASRU.2007.4430152.

[55] Zhao, G., Chukharev-Hudilainen, E., Sonsaat, S., Silpachai, A., Lucic, I., Gutierrez-Osuna, R., & Levis, J. (2018). L2-arctic: A non-native english speech corpus. In *Interspeech 2018*, ISCA, sept. 2018, p. 2783-2787. doi: 10.21437/Interspeech.2018-1110.

[56] Chen, N. F., Tong, R., Wee, D., Lee, P. X., Ma, B., & Li, H. (2015). iCALL corpus: Mandarin Chinese spoken by non-native speakers of European descent. In *INTERSPEECH* (pp. 324-328). Doi: 10.21437/Interspeech.2015-148.

[57] Algabri, M., Mathkour, H., Alsulaiman, M., & Bencherif, M. A. (2022). Mispronunciation detection and diagnosis with articulatory-level feedback generation for non-native arabic speech. *Mathematics*, *10*(15), 2727. Doi: 10.3390/math10152727.

[58] Alotaibi, Y., & Meftah, A. (2013). Review of distinctive phonetic features and the Arabic share in related modern research. *Turkish Journal of Electrical Engineering and Computer Sciences*, *21*(5), 1426-1439. Doi: 10.3906/elk-1112-29.

[59] Al-Marri, M., Raafat, H., Abdallah, M., Abdou, S., & Rashwan, M. (2018). Computer aided qur'an pronunciation using dnn. *Journal of Intelligent & Fuzzy Systems*, *34*(5), 3257-3271.

[60] Alotaibi, Y. A., & Muhammad, G. (2010). Study on pharyngeal and uvular consonants in foreign accented Arabic for ASR. *Computer Speech & Language*, *24*(2), 219-231. Doi: 10.1016/j.csl.2009.04.005.

[61] Khan, A. F. A., Mourad, O., Mannan, A. M. K. B., Dahan, H. B. A. M., & Abushariah, M. A. (2013). Automatic Arabic pronunciation scoring for computer aided language learning. In *2013 1st international conference on communications, signal processing, and their applications (ICCSPA)* (pp. 1-6). IEEE. Doi: 10.1109/ICCSPA.2013.6487246.

[62] Bahi, H., & Necibi, K. (2020). Fuzzy logic applied for pronunciation assessment. *International Journal of Computer-Assisted Language Learning and Teaching (IJCALLT)*, *10*(1), 60-72. Doi: 10.4018/IJCALLT.2020010105.

[63] Abdou, S. M., Hamid, S. E., Rashwan, M., Samir, A., Abdel-Hamid, O., Shahin, M., & Nazih, W. (2006). Computer aided pronunciation learning system using speech recognition techniques. In *Ninth International Conference on Spoken Language Processing*.

[64] Abdou, S. M., & Rashwan, M. (2014). A Computer Aided Pronunciation Learning system for teaching the holy quran Recitation rules. In *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)* (pp. 543-550). IEEE. Doi: 10.1109/AICCSA.2014.7073246.

[65] Al Hindi, A., Alsulaiman, M., Muhammad, G., & Al-Kahtani, S. (2014). Automatic pronunciation error detection of nonnative Arabic Speech. In *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)* (pp. 190-197). IEEE. Doi: 10.1109/AICCSA.2014.7073198.

[66] Maqsood, M., Habib, H. A., Nawaz, T., & Haider, K. Z. (2016). A complete mispronunciation detection system for Arabic phonemes using SVM. *International Journal of Computer Science and Network Security (IJCSNS)*, *16*(3), 30.

[67] Asif, A., Mukhtar, H., Alqadheeb, F., Ahmad, H. F., & Alhumam, A. (2021). An approach for pronunciation classification of classical arabic phonemes using deep learning. *Applied Sciences*, *12*(1), 238. Doi: 10.3390/app12010238.

[68] Ziafat, N., Ahmad, H. F., Fatima, I., Zia, M., Alhumam, A., & Rajpoot, K. (2021). Correct pronunciation detection of the arabic alphabet using deep learning. *Applied Sciences*, *11*(6), 2508. Doi: 10.3390/app11062508.

[69] Ahmed, A., Bader, M., Shahin, I., Nassif, A. B., Werghi, N., & Basel, M. (2023). Arabic Mispronunciation Recognition System Using LSTM Network. *Information*, *14*(7), 413.

[70] Harere, A. A., & Jallad, K. A. (2023). Mispronunciation detection of basic quranic recitation rules using deep learning. *arXiv preprint arXiv:2305.06429*.

[71] Jebara, T., & Jebara, T. (2004). Generative versus discriminative learning. *Machine learning: discriminative and generative*, 17-60. Doi: 10.1007/978-1-4419-9011-2_2.

[72] Foster, D. (2022). Generative deep learning. " O'Reilly Media, Inc.".

[73] Gheisari, M., Ebrahimzadeh, F., Rahimi, M., Moazzamigodarzi, M., Liu, Y., Dutta Pramanik, P. K., ... & Kosari, S. (2023). Deep learning: Applications, architectures, models, tools, and frameworks: A comprehensive survey. *CAAI Transactions on Intelligence Technology*, *8*(3), 581-606. Doi: 10.1049/cit2.12180.

[74] Bengio, Y., Goodfellow, I., & Courville, A. (2017). *Deep learning* (Vol. 1). Cambridge, MA, USA: MIT Press.

[75] Hijazi, S., Kumar, R., & Rowen, C. (2015). Using convolutional neural networks for image recognition. *Cadence Design Systems Inc.: San Jose, CA, USA*, *9*(1).

[76] Graupe, D. (2013). *Principles of artificial neural networks* (Vol. 7). World Scientific.

[77] Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, *2*(5), 359-366. Doi: 10.1016/0893-6080(89)90020-8.

[78] Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation*, *31*(7), 1235-1270. Doi: 10.1162/neco_a_01199.

[79] Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

[80] Niu, Z., Zhong, G., & Yu, H. (2021). A review on the attention mechanism of deep learning. *Neurocomputing*, *452*, 48-62. Doi: 10.1016/j.neucom.2021.03.091.

[81] LeCun, Y., Denker, J., & Solla, S. (1989). Optimal brain damage. *Advances in neural information processing systems*, *2*.

[82] Roffo, G. (2017). Ranking to learn and learning to rank: On the role of ranking in pattern recognition applications. *arXiv preprint.* Doi: 10.48550/arXiv.1706.05933.

[83] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541-551. Doi: 10.1162/neco.1989.1.4.541.

[84] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

[85] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[86] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

[87] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139-144. Doi: 10.1145/3422622.

[88] Kingma, D. P. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[89] Kingma, D. P., & Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, *12*(4), 307-392.doi: 10.1561/2200000056.

[90] Wei, W., Hengguan, H., Xiangming, G., Hao, W., & Ye, W. (2022). Unsupervised Mismatch Localization in Cross-Modal Sequential Data with Application to Mispronunciations Localization. *arXiv preprint arXiv:2205.02670*.

[91] Shahin, M., & Ahmed, B. (2024). Phonological-Level Mispronunciation Detection and Diagnosis. In *Proc. Interspeech 2024* (pp. 307-311). Doi: 10.21437/Interspeech.2024-2217.

[92] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798-1828.

[93] Zhong, G., Wang, L. N., Ling, X., & Dong, J. (2016). An overview on data representation learning: From traditional feature learning to recent deep learning. *The Journal of Finance and Data Science*, *2*(4), 265-278. Doi: 10.1016/j.jfds.2017.05.001.

[94] Zhang, D., Yin, J., Zhu, X., & Zhang, C. (2018). Network representation learning: A survey. *IEEE transactions on Big Data*, *6*(1), 3-28. Doi: 10.1109/TBDATA.2018.2850013.

[95] Liu, Z., Lin, Y., & Sun, M. (2023). *Representation learning for natural language processing* (p. 521). Springer Nature. Doi: 10.1007/978-981-99-1600-9.

[96] Peng, L., Gao, Y., Bao, R., Li, Y., & Zhang, J. (2023). End-to-End Mispronunciation Detection and Diagnosis Using Transfer Learning. *Applied Sciences*, *13*(11), 6793.

[97] Wadud, M. A. H., Alatiyyah, M., & Mridha, M. F. (2022). Non-autoregressive end-to-end neural modeling for automatic pronunciation error detection. *Applied Sciences*, *13*(1), 109. Doi: 10.3390/app13010109.

[98] Kheir, Y. E., Chowdhury, S. A., & Ali, A. (2023). Multi-view multi-task representation learning for mispronunciation detection. *arXiv preprint arXiv:2306.01845*.

[99] Charu C. Aggarwal, *Outlier Analysis*. 2016.

[100] Hawkins, D. (1980). Identification of outliers. Doi: 10.1007/978-94-015-3994-4.

[101] An, J., & Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, *2*(1), 1-18. Doi : https://doi.org/10.1002/tee.22868

[102] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 1-58. https://doi.org/10.1145/1541880.154188

[103] Cissokho, Y., Fadel, S., Millson, R., Pourhasan, R., & Boily, P. (2020). Anomaly Detection and Outlier Analysis. *Data Science Report Series*.

[104] de Albuquerque Filho, J. E., Brandão, L. C., Fernandes, B. J. T., & Maciel, A. M. (2022). A review of neural networks for anomaly detection. *IEEE Access*, *10*, 112342-112367. Doi: 10.1109/ACCESS.2022.3216007.

[105] Sulayman, I. I. A., & Ouda, A. (2018). Data analytics methods for anomaly detection: Evolution and recommendations. In *2018 International Conference on Signal Processing and Information Security (ICSPIS)* (pp. 1-4). IEEE. Doi: 10.1109/CSPIS.2018.8642713.

[106] Frihia, H., & Bahi, H. (2020). One-class training for intrusion detection. In *Proceedings of the 1st International Conference on Intelligent Systems and Pattern Recognition* (pp. 12-16). Doi: 10.1145/3432867.3432898.

[107] Shahin, M. A., Ahmed, B., Ji, J. X., & Ballard, K. J. (2018). Anomaly Detection Approach for Pronunciation Verification of Disordered Speech Using Speech Attribute Features. In *INTERSPEECH* (pp. 1671-1675). Doi: 10.21437/Interspeech.2018-1319.

[108] Shahin, M., Zafar, U., & Ahmed, B. (2019). The automatic detection of speech disorders in children: Challenges, opportunities, and preliminary results. *IEEE Journal of Selected Topics in Signal Processing*, *14*(2), 400-412. Doi: 10.1109/JSTSP.2019.2959393.

[109] Wei, R., Garcia, C., El-Sayed, A., Peterson, V., & Mahmood, A. (2020). Variations in variational autoencoders-a comparative evaluation. *Ieee Access*, *8*, 153651-153670. Doi: 10.1109/ACCESS.2020.3018151.

[110] Bank, D., Koenigstein, N., & Giryes, R. (2023). Autoencoders. *Machine learning for data science handbook: data mining and knowledge discovery handbook*, 353-374. Doi : https://doi.org/10.1007/978-3-031-24628-9_16.

[111]Sayed, H. M., ElDeeb, H. E., & Taie, S. A. (2023). Bimodal variational autoencoder for audiovisual speech recognition. *Machine Learning*, *112*(4), 1201-1226. Doi : https://doi.org/10.1007/s10994-021-06112-5.

[112] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1.

1986. *Biometrika*, *71*(599-607), 6. Doi: https://doi.org/10.7551/mitpress/4943.003.0128

[113] Baldi, P. (2012). Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning* (pp. 37-49). JMLR Workshop and Conference Proceedings.

[114] Tschannen, M., Bachem, O., & Lucic, M. (2018). Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*.

[115] Ehsani, N., Aminifar, F., & Mohsenian-Rad, H. (2022). Convolutional autoencoder anomaly detection and classification based on distribution PMU measurements. *IET Generation, Transmission & Distribution*, *16*(14), 2816-2828.

[116] Odaibo, S. (2019). Tutorial: Deriving the standard variational autoencoder (vae) loss function. *arXiv preprint arXiv:1907.08956*.

[117] Aly, S. A., Salah, A., & Eraqi, H. M. (2021). ASMDD: Arabic Speech Mispronunciation Detection Dataset. *arXiv preprint arXiv:2111.01136*.

[118] Nenov, R., Nguyen, D. K., Balazs, P., & Boţ, R. I. (2023). Accelerated Griffin-Lim algorithm: A fast and provably converging numerical method for phase retrieval. *IEEE Transactions on Signal Processing*.

[119] Kirk, A. (2019). Data visualisation: A handbook for data driven design.

[120] Khalid, Z. M., & Zeebaree, S. R. (2021). Big data analysis for data visualization: A review. *International Journal of Science and Business*, *5*(2), 64-75.

[121] Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4, No. 4, p. 738). New York: springer.

[122] Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, *2*(4), 433-459. Doi : https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.101

[123] Besacier, L., Barnard, E., Karpov, A., & Schultz, T. (2014). Automatic speech recognition for under-resourced languages: A survey. *Speech communication*, *56*, 85-100. Doi: 10.1016/j.specom.2013.07.008.

[124] Habash, N. Y. (2010). *Introduction to Arabic natural language processing*. Morgan & Claypool Publishers.

[125] Abdou, S. M., & Moussa, A. M. (2019). Arabic speech recognition: Challenges and state of the art. *Computational linguistics, speech and image processing for arabic language*, 1-27.

[126] Seliya, N., Abdollah Zadeh, A., & Khoshgoftaar, T. M. (2021). A literature review on one-class classification and its potential applications in big data. *Journal of Big Data*, *8*, 1-31. Doi :https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00514-x.

[127] Perera, P., Oza, P., & Patel, V. M. (2021). One-class classification: A survey. *arXiv preprint arXiv:2101.03064*.

[128] Wenzhu, S., Wenting, H., Zufeng, X., & Jianping, C. (2019, July). Overview of one-class classification. In *2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)* (pp. 6-10). IEEE. Doi: 10.1109/SIPROCESS.2019.8868559.

[129] Perera, P., & Patel, V. M. (2019). Learning deep features for one-class classification. *IEEE Transactions on Image Processing*, *28*(11), 5450-5463.

[130] Oza, P., & Patel, V. M. (2018). One-class convolutional neural network. *IEEE Signal Processing Letters*, *26*(2), 277-281. Doi: 10.1109/LSP.2018.2889273.

[131] Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of big data*, *6*(1), 1-54. Doi: 10.1186/s40537-019-0192-5.

[132] Tiwari, V. (2010). MFCC and its applications in speaker recognition. *International journal on emerging technologies*, *1*(1), 19-22.

[133] Mumuni, A., & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. *Array*, *16*, 100258.

[134] Khosla, C., & Saini, B. S. (2020). Enhancing performance of deep learning models with different data augmentation techniques: A survey. In *2020 International Conference on Intelligent Engineering and Management (ICIEM)* (pp. 79-85). IEEE.

[135] Bayer, M., Kaufhold, M. A., & Reuter, C. (2022). A survey on data augmentation for text classification. *ACM Computing Surveys*, *55*(7), 1-39.

[136] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, *6*(1), 1-48. Doi: 10.1186/s40537-019-0197-0.

[137] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84-90. Doi: 10.1145/3065386.

[138] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778). Doi: 10.1109/CVPR.2016.90.

[139] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826). Doi: 10.1109/CVPR.2016.308.

[140] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708). Doi: 10.1109/CVPR.2017.243.

[141] Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., & Xu, H. (2020). Time series data augmentation for deep learning: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, p. 4653-4660. Doi: 10.24963/ijcai.2021/631.

[142] Cauli, N., & Reforgiato Recupero, D. (2022). Survey on videos data augmentation for deep learning models. *Future Internet*, *14*(3), 93.

[143] Yang, Z., Sinnott, R. O., Bailey, J., & Ke, Q. (2023). A survey of automated data augmentation algorithms for deep learning-based image classification tasks. *Knowledge and Information Systems*, *65*(7), 2805-2861.

[144] Wang, Z., She, Q., & Ward, T. E. (2021). Generative adversarial networks in computer vision: A survey and taxonomy. *ACM Computing Surveys (CSUR)*, *54*(2), 1-38. Doi: 10.1145/3439723.

[145] Sabuhi, M., Zhou, M., Bezemer, C. P., & Musilek, P. (2021). Applications of generative adversarial networks in anomaly detection: a systematic literature review. *Ieee Access*, *9*, 161003-161029. Doi: 10.1109/ACCESS.2021.3131949.

[146] Abayomi-Alli, O. O., Damaševičius, R., Qazi, A., Adedoyin-Olowe, M., & Misra, S. (2022). Data augmentation and deep learning methods in sound classification: A systematic review. *Electronics*, *11*(22), 3795. Doi : https://doi.org/10.3390/electronics11223795.

[147] Ning, Y., He, S., Wu, Z., Xing, C., & Zhang, L. J. (2019). A review of deep learning based speech synthesis. *Applied Sciences*, *9*(19), 4050. Doi : https://doi.org/10.3390/app9194050.

[148] Ko, T., Peddinti, V., Povey, D., & Khudanpur, S. (2015). Audio augmentation for speech recognition. In *Interspeech* (Vol. 2015, p. 3586). Doi: 10.21437/Interspeech.2015-711.

[149] Li, B., Hou, Y., & Che, W. (2022). Data augmentation approaches in natural language processing: A survey. *Ai Open*, *3*, 71-90. Doi: 10.1016/j.aiopen.2022.03.001.

[150] Schlüter, J., & Grill, T. (2015, October). Exploring data augmentation for improved singing voice detection with neural networks. In *ISMIR* (pp. 121-126).

[151] Jaitly, N., & Hinton, G. E. (2013, June). Vocal tract length perturbation (VTLP) improves speech recognition. In *Proc. ICML workshop on deep learning for audio, speech and language* (Vol. 117, p. 21).

[152] Ragni, A., Knill, K. M., Rath, S. P., & Gales, M. J. (2014). Data augmentation for low resource languages. In *INTERSPEECH 2014: 15th annual conference of the international speech communication association* (pp. 810-814). International Speech Communication Association (ISCA). Doi: 10.21437/Interspeech.2014-207.

[153] Cui, X., Goel, V., & Kingsbury, B. (2015). Data augmentation for deep neural network acoustic modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(9), 1469-1477. Doi: 10.1109/TASLP.2015.2438544.

[154] Fukuda, T., Fernandez, R., Rosenberg, A., Thomas, S., Ramabhadran, B., Sorin, A., & Kurata, G. (2018, September). Data Augmentation Improves Recognition of Foreign Accented Speech. In *Interspeech* (No. September, pp. 2409-2413). Doi: 10.21437/Interspeech.2018-1211.

[155] Arakawa, R., Takamichi, S., & Saruwatari, H. (2019). Implementation of DNN-based real-time voice conversion and its improvements by audio data augmentation and mask-shaped device. *Proc. SSW10*, 93-98. Doi: 10.21437/SSW.2019-17.

[156] Nanni, L., Maguolo, G., & Paci, M. (2020). Data augmentation approaches for improving animal audio classification. *Ecological Informatics*, *57*, 101084. Doi : https://doi.org/10.1016/j.ecoinf.2020.101084.

[157] Wang, E. K., Yu, J., Chen, C. M., Kumari, S., & Rodrigues, J. J. (2022). Data augmentation for internet of things dialog system. *Mobile Networks and Applications*, 1-14. Doi: 10.1007/s11036-020-01638-9.

[158] Koszewski, D., & Kostek, B. (2020). Musical instrument tagging using data augmentation and effective noisy data processing. *Journal of the Audio Engineering Society*, *68*(1/2), 57-65.

[159] Mushtaq, Z., & Su, S. F. (2020). Environmental sound classification using a regularized deep convolutional neural network with data augmentation. *Applied Acoustics*, *167*, 107389. Doi: 10.1016/j.apacoust.2020.107389.

[160] Kharitonov, E., Rivière, M., Synnaeve, G., Wolf, L., Mazaré, P. E., Douze, M., & Dupoux, E. (2021, January). Data augmenting contrastive learning of speech representations in the time domain. In *2021 IEEE Spoken Language Technology Workshop (SLT)* (pp. 215-222). IEEE.

[161] Cortes, C. (1995). Support-Vector Networks. *Machine Learning*. vol. 20, nº 3, p. 273-297, sept. 1995, Doi: 10.1007/BF00994018.