Bitlis Eren Üniversitesi Fen Bilimleri Dergisi
BİTLİS EREN UNIVERSITY JOURNAL OF SCIENCE
ISSN: 2147-3129/e-ISSN: 2147-3188
VOLUME: 13 NO: 2 PAGE: 492-504 YEAR: 2024
DOI:10.17798/bitlisfen.1434925

Optimizing Speech to Text Conversion in Turkish: An Analysis of Machine Learning Approaches

İzel Zeynep GENÇYILMAZ [1], Kürşat Mustafa KARAOĞLAN [1*]

[1]Department of Computer Engineering, Faculty of Engineering, Karabuk University, Karabuk (ORCID: 0009-0009-0025-3394) (ORCID: 0000-0001-9830-7622)

Keywords: Speech to Text Conversion, Natural Language Processing, Convolutional Neural Network, Convolutional Recurrent Neural Network, Machine Learning, Deep Learning.

Abstract
The Conversion of Speech to Text (CoST) is crucial for developing automated systems to understand and process voice commands. Studies have focused on developing this task, especially for Turkish-specific voice commands, a strategic language in the international arena. However, researchers face various challenges, such as Turkish's suffixed structure, phonological features and unique letters, dialect and accent differences, word stress, word-initial vowel effects, background noise, gender-based sound variations, and dialectal differences. To address the challenges above, this study aims to convert speech data consisting of Turkish-specific audio clips, which have been limitedly researched in the literature, into texts with high-performance accuracy using different Machine Learning (ML) models, especially models such as Convolutional Neural Network and Convolutional Recurrent Neural Network (CRNN). For this purpose, experimental studies were conducted on a dataset of 26,485 Turkish audio clips, and performance evaluation was performed with various metrics. In addition, hyperparameters were optimized to improve the model's performance in experimental studies. A performance of over 97% has been achieved according to the F1-score metric. The highest performance results were obtained with the CRNN approach. In conclusion, this study provides valuable insights into the strengths and limitations of various ML models applied to CoST. In addition to potentially contributing to a wide range of applications, such as supporting hard-of-hearing individuals, facilitating notetaking, automatic captioning, and improving voice command recognition systems, this study is one of the first in the literature on CoST in Turkish.

1. Introduction

Natural Language Processing (NLP) is a pivotal field in computer science, focusing on enabling computers to understand and interpret human languages [1]. Language can be classified into three primary types: spoken, written, and sign language [2]. The conversion of spoken words into written text not only assists individuals with hearing impairments in comprehending others but also enhances our capacity to concentrate on presentations or lectures without note-taking. Conversion of Speech to Text (CoST) integration with smart devices and home systems, like Siri or Alexa, further enriches our daily interactions. Historically, speech analysis techniques have evolved significantly. In the early 2000s, speech analysis heavily relied on the utilization of the Hidden Markov Model (HMM) and Gaussian Mixture Model techniques [3]. HMMs employed a statistical modeling approach to comprehend the correlation between hidden states and transition probabilities, necessitating extensive data to achieve satisfactory results in tasks like classification or recognition [4]. In the context of speech processing, HMMs exhibited limited accuracy rates, particularly in terms of accuracy in data-scarce environments [5].

Consequently, attaining a high success rate in CoST presented challenges in the 2000s. However, in the 2010s, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models emerged, known for their enhanced accuracy [6]. LSTM, a variation of Recurrent Neural Network (RNN), was designed to address the challenge of long-term dependencies, thus incorporating a memory cell controlled by gates that

[*] Corresponding author: kkaraoglan@karabuk.edu.tr

Received:10.02.2024, Accepted:20.03.2024

determine the information to be retained or forgotten within the network [7]. Leveraging this attribute empowers LSTM to keep past information proficiently, amplifying its efficacy when dealing with intricate sequential assignments like natural language or speech signal processing [8]. On the other hand, GRU, a more efficient counterpart to LSTM, offers quicker processing and reduced computational demands. Nevertheless, despite these advantages, GRU sometimes they may fall short in NLP tasks that require the recognition of extended sequences, like CoST, or context-dependent aspects of language where LSTM networks often excel due to their enhanced ability to retain long-term data dependencies [9]. This discrepancy arises from its limited ability to capture long-term dependencies as effectively as LSTM. Over the past few years, significant progress has been witnessed in deep learning techniques and their integration into NLP, leading to rapid improvement in success rates [10]. Through the strategic utilization of methodologies like deep neural network (DNN), Convolutional Neural Network (CNN), and RNN, remarkable strides have been made, resulting in notably elevated levels of performance [11], [12]. While significant strides have been achieved in deep learning for speech processing, several challenges must be addressed. These hurdles encompass a need for substantial labelled data, the pursuit of model interpretability, and varying environmental conditions [13]. These environmental factors contain various variables, ranging from word count, gender-specific voices, and ethnic origins to audio quality, duration, and vocabulary representation within the dataset. Addressing these intricacies is pivotal for further advancement in the field. Our study tackles these challenges by employing and comparing different ML techniques. Utilizing a comprehensive Kaggle dataset, curated by Kurtkaya, which comprises 26,485 one-second Turkish audio recordings across 14 various commands, our objectives are twofold [14].

This study aims to enhance the accuracy and efficiency of CoST in Turkish, thereby significantly contributing to the broader field of NLP and automated voice recognition systems. The comparison of various Machine Learning (ML) techniques, focusing on deep learning and CNN, endeavors to find the most effective and accurate results. By optimizing parameters such as the number of epochs and other hyperparameters, an improvement in the model's performance is sought, and novel insights are provided by contrasting the findings of this study with other similar works.

This study is organized into four main sections. The first section provides an introduction. The second section is a comprehensive literature review that sets the stage for the research, delves into the goals and overarching framework, gives details of the methodology, and explores the approaches and techniques employed. The third section evaluates criteria, hyperparameters, and comparative results. The fourth and final section discusses the detailed findings and offers suggestions for future research.

2. Material and Method

The dataset details are outlined initially, followed by a discussion on the pre-processing steps and the proposed approach to speech recognition. The architecture of the proposed model is detailed after that, with a specific focus on constructing the CNN and CRNN models. The dataset encompasses a collection of 26,485 audio recordings featuring a spectrum of 14 distinct commands. Each audio recording has been meticulously standardized to precisely 1 second, with a chosen sampling frequency of 16 kHz. Table 1 provides a comprehensive statistical overview, encapsulating key aspects and metrics that define the dataset's composition and characteristics.

Table 1. Summary of Dataset Characteristics and Features

Feature	Description
Total Audio Files	26485
Command Instances	Open - (Aç): 1995 Cancel - (İptal): 1952 Left - (Sol) : 1910 Up - (Yukarı) : 1892 Stop - (Dur) : 1887 Forward - (İleri) : 1886 Yes - (Evet) : 1885 Right - (Sağ) : 1883 Close - (Kapa) : 1882 Continue - (Devam) : 1880 Start - (Başlat) : 1879 Down - (Aşağı) : 1870 No - (Hayır) : 1843
Sampling Frequency	16 kHz

2.1. Speech Classification

In this section, the description of speech classification is provided, along with some fundamental terminologies encountered during our speech processing work. Waveform is a diagrammatic representation that aids in examining the displacement of sound waves over time alongside other essential parameters [15]. Frequency pertains to the rate at which the waveform repeats within one second. The highest point on the waveform is called the crest, while the lowest point is known as the trough. Amplitude signifies the distance from the center line to the crest or trough [16]. Spectrogram visually depicts an audio signal's evolving frequency spectrum over time [17]. This powerful tool is employed for sound signal analysis and finds utility in tasks such as identifying different sounds in a recording, analyzing pitch and timbre, monitoring the temporal evolution of sound, and detecting and categorizing noise [18].

Mel-Frequency Cepstral Coefficient (MFCC) denotes a method of extracting features that represent the spectral characteristics of an audio signal [19]. MFCCs apply Mel filtering to the logarithm of the signal's power spectrum. Subsequently, they compute the discrete cosine transform of the filtered signal. Popular libraries like Librosa or Sox are commonly utilized to extract MFCC coefficients from audio signals [20]. Sound classification encompasses the automated attribution of labels to audio recordings.
Diverse techniques are employed to accomplish this objective, with spectrograms and MFCCs emerging as the most prevalent approaches [21]. Spectrograms are leveraged to distil distinctive attributes from assorted sound recordings. In the subsequent feature extraction stage, our model leverages MFCCs to apply ML techniques. Several vital advantages underpin the choice of MFCCs over spectrograms.
Firstly, spectrograms often possess high-dimensional characteristics, slowing the model training process and requiring powerful computational resources. In contrast, MFCCs offer a more compact and efficient representation. Secondly, while spectrograms linearly process frequencies, MFCCs are designed to emulate the non-linear auditory perception of the human ear. This is particularly beneficial in CoST, as it allows for a more natural interpretation of audio data. However, it is noteworthy that spectrograms can be more effective than MFCCs in handling high frequencies [22]. Lastly, spectrograms' critical limitation is their reduced capability to accurately represent the temporal variations in sound within specific time intervals [23]. MFCCs, by design, provide a more detailed and precise reflection of these material changes, making them a more suitable choice for our CNN model's requirements [24].

In the study, one of the procedural steps involved using the 'path' column in the 'dataBase.xlsx' file, primarily to reference and retrieve audio files with .wav and .mp3 extensions from various directories using the' os.walk(directory_path)' command. For organizational purposes, these files were initially stored on an empty list. Subsequently, we prepared additional empty lists for labeling and calculating MFCC. The MFCC for each audio file was computed using the 'librosa.feature.mfcc' command. The labelling process utilized the file paths of the audio files, a method chosen for its straightforwardness as each audio file was systematically stored in its corresponding folder. Labelling was executed using the' os.path.

Since the folder names and labels encompass the names of 14 distinct commands, these were initially in string format. However, we converted these string labels into integer values to enhance the model's performance. This numerical representation is crucial as it facilitates the model's ability to process and understand the data more effectively.

Another vital aspect of our methodology was dividing data into training and test sets, with the test size set at 20%. This split is essential for evaluating the model's performance under varied conditions [25]. The research encompassed an array of ML models: Support Vector Machine (SVM), K-Nearest Neighbors, Decision Tree (DT), Random Forest (RF), Gaussian Naive Bayes (GNB), LSTM, GRU, CNN, and Convolutional Recurrent Neural Networks (CRNN). The prediction phase involved using the trained models to classify audio clips into their respective voice command categories. This step has been performed to determine the practical applicability of the models in real-world scenarios, such as voice command recognition systems. In the final phase, the performance of each model was compared using key metrics such as Accuracy, Precision, Recall, F1-Score, Receiver Operating Characteristic (ROC) curve, and Area Under the Curve (AUC).

Furthermore, hyperparameters were optimized to improve the performance of the models. This phase yielded insights into the strengths and weaknesses of each model, while also elucidating the potential and limitations of various ML approaches in CoST.

To concisely encapsulate, the steps in our study were as follows:

- Retrieval of audio files from the designated directory.
- Calculation of MFCCs for each audio file using Librosa.
- Utilization of the folder names as labels for the audio files.
- Segregation of the dataset into training and testing subsets.
- Employing various ML techniques.

- Comparing the results and optimizing hyperparameters for improved outcomes.

Figure 1 is provided to enhance understanding of the proposed approach structure.

2.2. Architecture of CNN Model

The CNN architecture is a deep learning model widely utilized for tasks like image classification [26], [27].

.

Nevertheless, thanks to recent advancements in learning methodologies, CNN has also demonstrated remarkable achievements in speech processing [20], [28], [29], [30], [31]. Convolutional layers employ filters to process the input data, generating feature maps [32].

$$\mathbf{x_l}(\mathbf{t},\mathbf{f}) = \sum_{\mathbf{a}=-\mathbf{p}}^{\mathbf{p}} \sum_{\mathbf{b}=-\mathbf{p}}^{\mathbf{p}} \mathbf{w_{ab}} \mathbf{x_{l-1}}(\mathbf{t}+\mathbf{a})(\mathbf{f}+\mathbf{b}) + \mathbf{b} \quad (1)$$

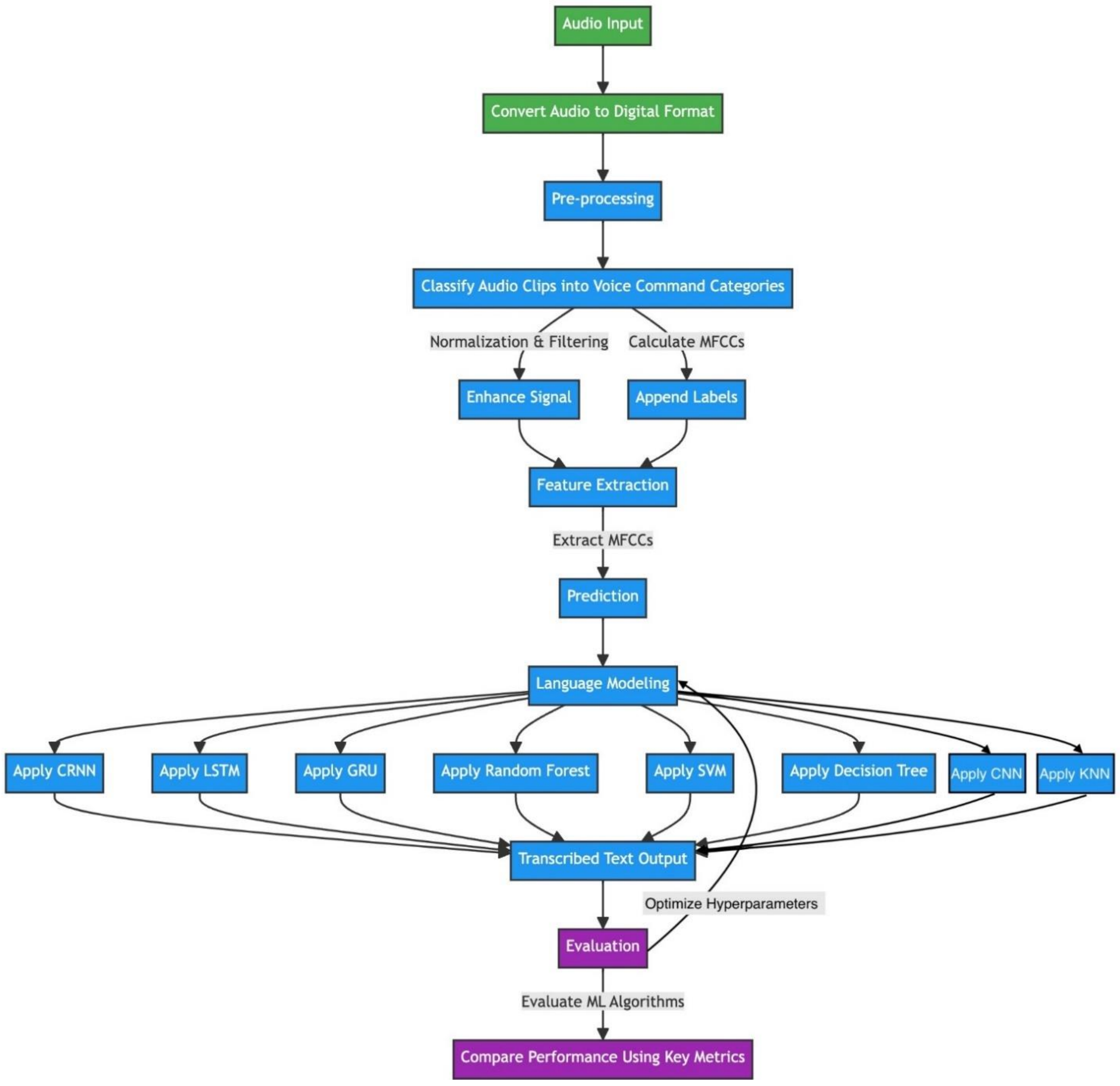

Figure 1. Structure of the proposed CoST approach.

Formula (1) is a mathematical expression of a convolution operation commonly used in neural networks [33]. The equation above represents the value at time t in the l-th layer and frequency f. The formula includes two nested sums. In the first sum, the variables a and b take values from -p to p. In the second sum, the variables a and b also take values from -p to p. The convolution operation in speech processing, like the spatial dimensions in image processing, is performed by summing neighboring values in the time and frequency dimensions [34]. Weights represent the learnable parameters of the filters, and the bias term (b) is added to the weighted sum.

The formula calculates an output in the current layer by taking the surrounding sum of the inputs from the previous layer and appropriately weighing each output [35]. With this formula, in CNN architecture, the network learns to extract relevant features from the input spectrogram and performs the task of speech recognition [36], [37]. The Keras Sequential model facilitates the construction of linearly arranged neural architectures, enabling the seamless integration of layers, like fully connected or convolutional layers, each with singular input and output tensors [38]. Its efficacy is simplifying unidirectional data flow in various standard neural network applications. The usage of the Sequential API is convenient. The reason for using Sequential in our model is that it allows us to easily stack the desired layers in the desired order, enabling the straightforward creation of a model.

The model is architecturally composed of five layers intricately designed to work harmoniously. The first and second layers are similar, each consisting of a Convolutional 2D (Conv2D) layer, followed by a MaxPooling2D layer, and a Dropout layer. These layers are for feature extraction and reducing overfitting [39], [40]. The third layer is a Flatten layer, serving as a bridge between the convolutional and dense layers by converting the 2D feature maps into a 1D feature vector, crucial for subsequent processing [41]. In the fourth layer, we have a fully connected Dense layer coupled with a Dropout layer.

The dense layer plays a key role in combining the complex features learned by earlier layers in a flexible, non-linear way [42]. The fifth and final layer is another Dense layer, responsible for output and utilizing a SoftMax activation function, making it suitable for multi-class classification tasks. Overall, the model includes two Convolutional layers, two Max-Pooling layers, one Flatten layer, two Dense layers, and three Dropout layers, with one layer designated for output. It is meticulously compiled with the Categorical Cross Entropy loss function and optimized using the Adam optimizer, ensuring effective training and performance. For a clearer understanding of the model's structure and layers, refer to the following table:

Table 2. Details of the applied layers

Layers	
Number	**Types**
1	Conv2D, MaxPooling2D, Dropout
2	Conv2D, MaxPooling2D, Dropout
3	Flatten
4	Dense (Fully connected), Dropout
5	Dense (Output with SoftMax)

The first layer applies convolution operations to the audio signal, using 32 filters of size 3x3, each capturing distinct frequency components and their temporal changes. The command "model.add(Conv2D(32, kernel_size=(3, 3), activation='ReLU', input_shape=(X_train.shape[1], X_train.shape[2], 1)))" implements this convolution operation.

The second layer's MaxPooling operation reduces the feature map's size, decreasing the number of parameters and risk of overfitting while improving computational efficiency [32], [43]. The command "model.add(MaxPooling2D(pool_size=(2, 2)))" is used for this purpose, employing a 2x2 pooling region. The Dropout layer, applied in both the first and fourth layers with a rate of 25%, aims to mitigate overfitting by randomly deactivating input units during each update cycle [40]. Following this, the procedures of Conv2D, MaxPooling2D, and Dropout layers are repeated, with the second Conv2D layer employing 48 filters. This increase in filters allows for capturing a broader range of complex features.
The Flatten layer converts the multi-dimensional feature map into a one-dimensional vector, formatted for input into a fully connected dense layer. The command "model.add(Flatten())" facilitates this transformation.

The subsequent Dense layer, connected to every neuron in the preceding layer, enhances the model's ability to recognize intricate patterns. The command "model.add (Dense (128, activation='ReLU'))" sets the neuron count and incorporates the 'ReLU' activation function.

The final output layer, with neurons equal to the total class count, employs the "softmax" activation function to provide probabilistic class predictions. The model is trained using the categorical cross-entropy loss function, which is ideal for multi-class classification. The Adam optimizer dynamically adapts the learning rate for each parameter, facilitating faster convergence and achieving superior performance compared to traditional optimization algorithms. [44]. The model's weights are adjusted throughout the training process to minimize the loss function, enabling it to recognize speech commands effectively.

2.3. The Architecture of CRNN Model

The architecture of the proposed CRNN model combines the strengths of both CNN and LSTM layers using the sequential model framework. This approach ensures seamless data flow from one layer to the next.

The model has five essential layers, each serving a specific function. Initially, the model begins with a Conv2D layer, which applies 2D convolutions to the input data, extracting critical features from the audio signals. In this initial stage, the layer utilizes 32 filters of size 3x3 each, employing the 'ReLU' activation function to maintain positive values while setting negative values to zero. Following the Conv2D layer, the MaxPooling2D layer selects the highest value among neighboring pixels within a 2x2 area, effectively reducing the dimensions of the feature maps. This helps the model generalize better by reducing its sensitivity to minor variations in the data. It achieves this by introducing stochasticity using a dropout mechanism, which randomly deactivates a portion (e.g., 25%) of the units in each layer during training. This prevents overfitting and encourages the model to learn more robust representations [39]. As the model progresses, an additional Conv2D layer with 64 filters is introduced, mirroring the initial layer in its composition.

The data shape transforms within the Reshape layer, adapting it seamlessly for processing by the subsequent LSTM layer. This LSTM layer, equipped with 64 internal units, is designed to handle sequential data, like sentences or time series, expertly. The network culminates in two Dense layers. The first one, housing 128 neurons, utilizes the ReLU activation function to introduce non-linearity. This layer is followed by another Dropout mechanism, strategically removing 25% of its neurons during training to curb overfitting. Finally, the last Dense layer leverages the SoftMax activation function to produce probabilistic predictions for each class, ensuring they all add up to 100%. The configuration of 64 units in the LSTM layer and 128 units in the Dense layer is not arbitrarily but meticulously chosen through hyperparameter tuning. These specific values were carefully selected through experiments to strike a balance between model performance and complexity, yielding the most optimal outcome.

3. Evaluation Metrics and Hyperparameters

This section is meticulously designed to comprehensively understand the methodology and outcomes associated with the study in question. Initially, the evaluative metrics employed to gauge the efficacy of the models are elucidated. Subsequently, an exposition on selecting hyperparameters for the training phase is proffered, covering a spectrum of parameters and configurations meticulously adjusted to optimize the models' performance during the training regimen. In culmination, a comparative scrutiny of the empirical results is undertaken. The effectiveness of various machine learning methodologies and the model is assessed using criteria such as ROC curves, accuracy, recall, precision, F1-score, True Positive Rate (TPR), and False Positive Rate (FPR) scores.

$$\boldsymbol{Accuracy} = \frac{(\boldsymbol{TP}+\boldsymbol{TN})}{(\boldsymbol{TP}+\boldsymbol{TN}+\boldsymbol{FP}+\boldsymbol{FN})} \tag{2}$$

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{(\boldsymbol{TP}+\boldsymbol{FP})} \tag{3}$$

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{(\boldsymbol{TP}+\boldsymbol{FN})} \tag{4}$$

$$\boldsymbol{F1Score} = \boldsymbol{2} \cdot \frac{(\boldsymbol{Precision} \cdot \boldsymbol{Recall})}{\boldsymbol{Precision}+\boldsymbol{Recall}} \tag{5}$$

$$\text{TPR} = \frac{\boldsymbol{TP}}{\boldsymbol{TP}+\boldsymbol{FN}} \tag{6}$$

$$\text{FPR} = \frac{\boldsymbol{FP}}{\boldsymbol{FP}+\boldsymbol{TN}} \tag{7}$$

In the assessment of the efficacy of classification models, we depend upon four principal metrics: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP denotes the quantity of accurately recognized positive instances, whereas TN pertains to the correct identification of negative cases. In contrast, FP refers to the quantity of positive predictions made in error, and FN represents the true positive cases that were overlooked.

Accuracy is expressed as the quotient of the sum of true positives and true negatives over the total number of cases. Accuracy shines when dealing with balanced data and similar costs for incorrect predictions. However, its utility wanes in the presence of imbalanced class distributions, where it may offer a misleadingly optimistic view of the model's performance.

Precision is the ratio of correctly predicted positive observations to the total predicted positives. Recall is the ratio of correctly predicted positive observations to all observations in the actual class. F1-Score is the harmonic means of Precision and Recall and thus conjoins the properties of both metrics. ROC curve illustrates the relationship between the TPR and the FPR across various classification thresholds. AUC quantifies the total performance of the model by measuring the area beneath the ROC curve [45].

The cross-validation technique constitutes an esteemed metric to gauge the proficiency of a model's generalization capabilities concerning unseen data [46]. It plays a pivotal role in circumventing the predicament of overfitting, where the model demonstrates superior efficacy on the dataset utilized for training [47]. It is a mechanism for adjudicating amongst diverse model candidates, thereby evaluating their capacity for generalization. Various methodologies of cross-validation prevail, including but not limited to k-fold, leave-one-out, and stratified cross-validation, each presenting its unique set of benefits and limitations contingent upon the dataset and the task at hand [25]. Within the ambit of our model, k-fold cross-validation has been predominantly employed. This technique augments the duration of training in comparison to a singular train-test partition, attributing to multiple iterations of training. While it diminishes the propensity for overfitting, it may concurrently engender a modicum of bias contingent upon the selection of folds [48]. It is ubiquitously recognized as a robust method for the selection of models and the fine-tuning of hyperparameters.

The practice of executing k-fold cross-validation repetitively, for instance, tenfold, and calculating the mean of the outcomes, furnishes a more steadfast and dependable gauge of model efficacy [49]. In the case of our model, a criterion of twenty iterations has been adopted. A superior cross-validation indicates a model's enhanced generalization ability [50]. It is imperative, however, to eschew exclusive reliance on cross-validation as the sole criterion for decision-making [51]. Other aspects, such as the model's complexity, interpretability, and other evaluation metrics, should be considered in the final adjudication process.

3.1 Hyperparameters for Training

Hyperparameters constitute configurational parameters employed to architect the learning schema, significantly impacting the efficacy of the models [52]. The different approaches have been tested and considered industry best practices. For example, the Adam optimizer has been chosen based on findings in [53], dropout rates have been selected based on findings in [40] and the number of epochs has been selected with the hands-on experiments considering training duration. The scenario of overfitting was considered, and early stopping was employed.

A detailed overview of the hyperparameters used is presented in Table 3, which yielded outstanding outcomes across various ML methodologies.

4. Results

This section meticulously analyses the research results and initiates a dialogue on prospective future investigations. It is organized to initially present several graphs that illuminate insights into model performance and evaluation, followed by a discussion on the findings.

Figure 2 compares accuracy, precision, recall, F1-score, and cross-validation score across various models, including CNN, CRNN, LSTM, RF, SVM, GNB, KNN, GRU and DT.

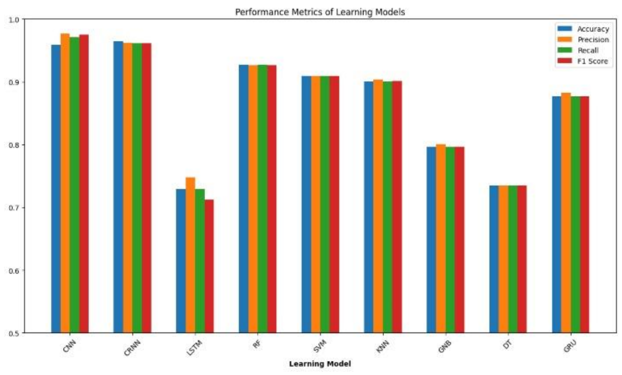

Figure 2. Comparative results for CoST task in Turkish.

Figure 3 illustrates the ROC curve for the CRNN model. Furthermore, additional graphs showcase the training and validation loss and accuracy rates over epochs for the models, highlighting their learning trajectory and capability to generalize over time. Subsequently, the discussion focuses on the notable outcomes and their contributions to the existing knowledge, alongside exploring potential avenues for further research.

Figure 4 shows the loss during training (in blue) and validation (in red) across epochs. If the validation loss increases, it could be a sign of overfitting. In the proposed model, the validation loss decreased over time.

Table 3. Hyperparameters for training process

Model Name	Hyperparameters	Model Name	Hyperparameters
RF	Number of Estimators: 100 Criterion: Gini Max Depth: None Min Samples Split: 2 Min Samples Leaf: 1 Max Features: Auto Random State: None	**LSTM**	Number of LSTM Units: 64 Number of Dense Units: 64 Dropout Rate: 0.3 Batch Size: 32 Optimizer: Adam Loss: Categorical Cross-Entropy Early_Stopping=True
CRNN	Conv2D Filters: 32, 64 Conv2D Kernel Size: (3, 3) Conv2D Activation: ReLU Conv2D Padding: Same MaxPooling Pool Size: (2, 2) MaxPooling Strides: None MaxPooling Padding: Valid LSTM Units: 64 First Dense Activation Function: ReLU Last Dense Activation Function: Softmax Number of Dense Units: 128 Optimizer: Adam Loss: Categorical Crossentropy Batch Size: 32 Early_Stopping=True	**CNN**	Conv2D Filters: 32, 64 Conv2D Kernel Size: (3, 3) Conv2D Activation: ReLU MaxPooling Pool Size: (2, 2) Dense Units: 128 Dense Activation: ReLU Dropout Rate: 0.25 Output Activation: Softmax Loss: Categorical Crossentropy Optimizer: Adam Batch Size: 32 Early_Stopping=True
DT	Criterion: Gini Splitter: Best Max Depth: None Min Samples Split: 2 Min Samples Leaf: 1	**GRU**	Number of GRU and Dense Units: 128 Early_Stopping=True Dropout Rate: 0.5 Activation of Dense Layer 1: ReLU Activation of Dense Layer 2: Softmax
KNN	Number of Neighbors: 3 Weight: Uniform Algorithm: Auto	**SVM**	C: 1 Kernel: RBF Gamma: Scale

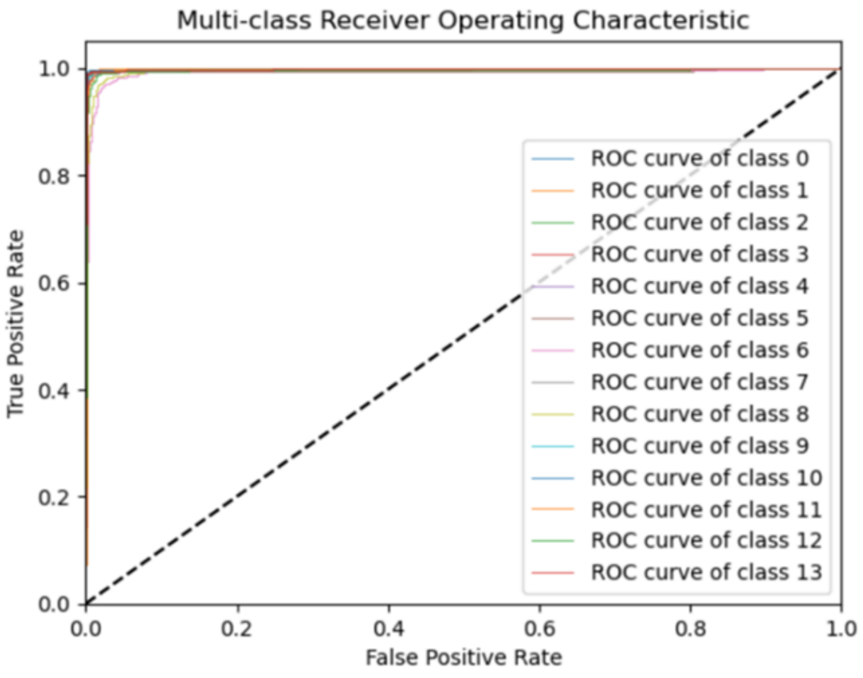

Figure 3. The ROC curve plots the performance of the implemented CRNN model.

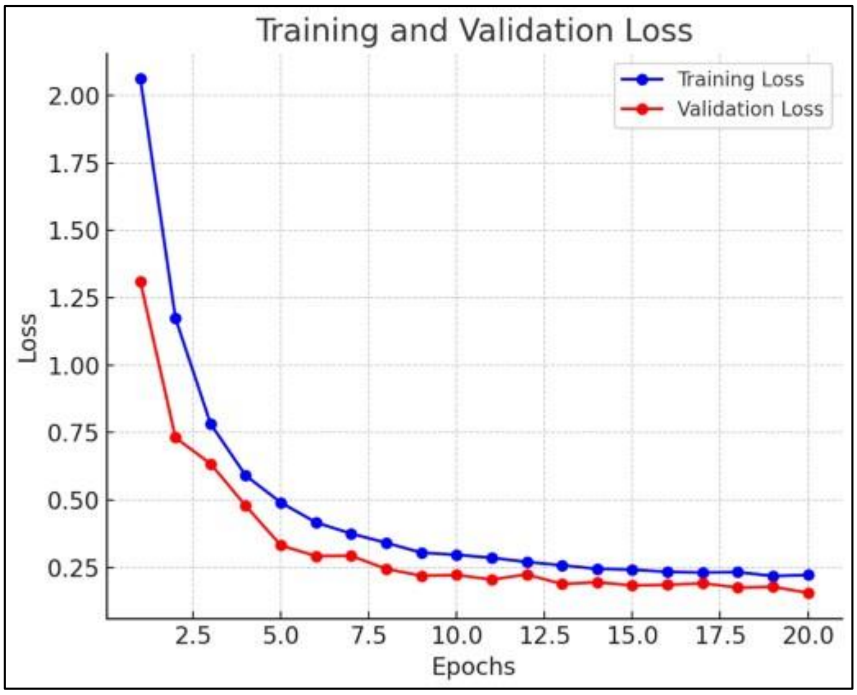

Figure 4. Visualizations depicting the training and validation loss progress of the model over time.

Figure 5 displays the accuracy during training (in blue) and validation (in red) across epochs. Increasing accuracy over time is a good sign, showing that the model is effectively learning to classify the data. Consistently higher validation accuracy than training accuracy indicates that the proposed model is generalizing well.

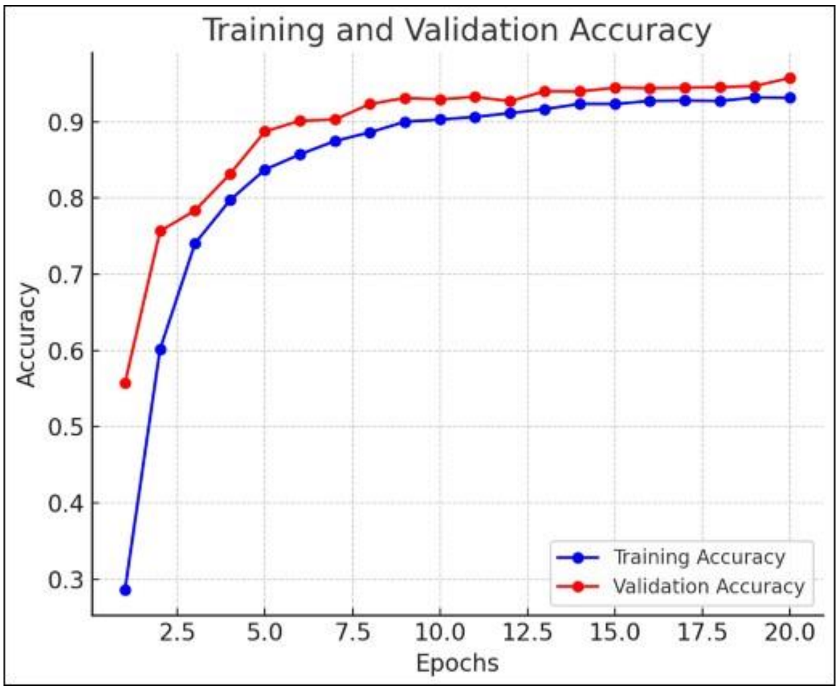

Figure 5. Visualizations illustrating the training and validation accuracy progress of the model over time.

Table 4 compares eight different ML models, evaluating their strengths and limitations across five key performance indicators: accuracy, precision, recall, F1 score, and cross-validation score. The CRNN model is pre-eminent, boasting the apex of accuracy (0.96468), while the CNN model eclipses its counterparts in both precision (0.97677) and F1 score (0.97552).

Moreover, the table furnishes statistical insights by summarizing each metric's maximum, minimum, mean, and standard deviation (Std) values. This multifaceted metric evaluation not only identifies the models that excel with the dataset in question but also accentuates the imperative of meticulously aligning ML methodologies with the stipulated performance objectives and the inherent characteristics of the dataset.

Table 4. Comparison of results obtained from various ML techniques in a table.

Proposed Approaches	Learning Model	Accuracy	Precision	Recall	F1-Score
1st Approach	CNN	0.95922	0.97677	0.97135	0.97552
2nd Approach	**CRNN**	**0.96468**	**0.96188**	**0.96168**	**0.96164**
3rd Approach	LSTM	0.72965	0.74829	0.72921	0.71210
4th Approach	RF	0.92694	0.9268	0.92691	0.92672
5th Approach	SVM	0.90976	0.90966	0.90985	0.90963
6th Approach	KNN	0.90070	0.9036	0.90075	0.90143
7th Approach	GNB	0.79687	0.80101	0.79657	0.79662
8th Approach	DT	0.73532	0.73513	0.73516	0.73486
9th Approach	GRU	0.87710	0.88287	0.87710	0.876824
	Max.:	0.96468	0.97677	0.97135	0.97552
	Min.:	0.72965	0.73513	0.72921	0.71210
	Mean:	0.86539	0.87039	0.86643	0.86481
	Std.:	0.09688	0.09526	0.09837	0.10259

5. Conclusion and Discussion

This section thoroughly reviews the outcomes of our research and initiates a discussion on possible future directions. It begins by exploring the significant findings and their impact on the current state of knowledge. Further areas for research exploration are identified. Different ML models were evaluated, utilizing metrics including accuracy, precision, recall, F1 score, and cross-validation score. Notably, the CNN model exhibited exceptional effectiveness, especially CoST in Turkish, achieving an accuracy rate of 95.34% after 10 epochs, which improved to 95.92% upon reaching 20 epochs.

Moreover, it exhibited high precision, recall, and F1 scores. Surpassing the CNN, the CRNN model showed even higher accuracy, with 96.35% at 10 epochs and 96.46% at 20 epochs, alongside exceptional precision, recall, and F1 scores, indicating superior performance for this specific task. As a side note, the epoch count has been increased to 30 in our methodology. However, due to the activation of the early stopping parameter set to true, the system could identify the optimum number of epochs without succumbing to overfitting or underfitting, thereby ensuring the robustness and efficiency of the training process. On the contrary, the LSTM model could have been more effective, achieving only 73% accuracy, 75% precision, 73% recall, and a 71% F1 score, suggesting it is a less optimal choice for CoST in Turkish. Other models, including Random Forest, SVM, KNN, Gaussian Naive Bayes, GRU and Decision Tree, demonstrated mixed results, with Random Forest achieving the highest accuracy of 93%. There remains potential for improvement in their performance compared to CNN and CRNN. While direct comparison with existing research is challenged by differences in methodologies, datasets, and lack of enough literature about CoST in Turkish, our work distinguishes itself by offering a different solution, achieving outstanding accuracy. Our CRNN model outperforms previous research, achieving outstanding results with accuracy, F1 score, and cross-validation score of 0.96. It exceeds the capabilities of the DNN HM model presented in the article 'Turkish Speech Recognition

Based on Deep Neural Networks' [54], which reports a WER of 14.82 and an accuracy of 0.85. 'A Novel End-to-End Turkish Text-to-Speech (TTS) System via Deep Learning' [55] while prioritizing perceptual quality with 4.49 MOS, our research achieves the highest possible transcription accuracy.

This study marks a significant advancement in speech recognition and NLP within the Turkish language context as we look ahead to future developments. Yet, several areas warrant further investigation. Firstly, expanding the dataset size would be crucial for future studies. A more extensive and diverse dataset would enable a more comprehensive evaluation of the models' generalization capabilities and performance in real-world scenarios. Secondly, the development of model optimization techniques would be beneficial. Investigating strategies such as hyperparameter tuning, diversifying model architectures, and optimizing data pre-processing steps could significantly enhance the effectiveness of CNN, CRNN, and other models. Thirdly, expanding the scope of research to encompass complex tasks like emotion recognition and speech variation analysis is essential. Such advancements allow a more holistic understanding of speech by considering tone, emphasis, and emotional content. Finally, exploring targeted applications and industrial implementations of these models is imperative. Evaluating the performance of CoST techniques in domains like automatic captioning, virtual assistants, and call center speech analysis would provide valuable insights into the usability and effectiveness of these technologies in practical settings. These directions pave the way for future research in Turkish speech recognition and contribute significantly to the broader field of NLP, fostering the development of more sophisticated speech recognition systems.

Contributions of the authors

This study has significantly profited from the diverse expertise of its authors. Karaoğlan played a crucial role in conceptualization, research design, editing, supervision, project management, critical review, and final approval. Gençyılmaz focused on the literature review, data collection, data analysis, model development, and manuscript writing.

Conflict of Interest Statement

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The study complies with research and publication ethics.

References

[1] S. McRoy, *Principles of natural language processing*. Susan McRoy, 2021.

[2] A. Akmajian, A. K. Farmer, L. Bickmore, R. A. Demers, and R. M. Harnish, Linguistics: *An introduction to language and communication.* MIT press, 2017.

[3] M. Gales, S. Young, and Others, “The application of hidden Markov models in speech recognition,” *Foundations and Trends in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2008.

[4] M. L. P. Bueno, A. Hommersom, P. J. F. Lucas, and A. Linard, “Asymmetric hidden Markov models,” *International Journal of Approximate Reasoning*, vol. 88, pp. 169–191, 2017.

[5] M. S. Barakat, M. E. Gadallah, T. Nazmy, and T. El Arif, “Investigating the effect of speech features and the number of HMM mixtures in the quality HMM-based synthesizers,” *in The International Conference on Electrical Engineering*, 2008, vol. 6, pp. 1–12.

[6] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, “Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM,” arXiv preprint arXiv:1706. 02737, 2017.

[7] H. Sak, A. W. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” 2014.

[8] G. Van Houdt, C. Mosquera, and G. Nápoles, “A review on the long short-term memory model,” *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5929–5955, 2020.

[9] S. Yang, X. Yu, and Y. Zhou, “Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example,” *in 2020 International workshop on electronic communication and artificial intelligence (IWECAI)*, 2020, pp. 98–101.

[10] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” Advances in neural information processing systems, vol. 27, 2014.

[11] M. Li et al., “The deep learning compiler: A comprehensive survey,” IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 3, pp. 708–727, 2020.

[12] K. Mohamad and K. M. Karaoglan, “Enhancing Deep Learning-Based Sentiment Analysis Using Static and Contextual Language Models,” *Bitlis Eren Universitesi Fen Bilimleri Dergisi*, vol. 12, no. 3, pp. 712–724, 2023.

[13] A. Mehrish, N. Majumder, R. Bharadwaj, R. Mihalcea, and S. Poria, “A review of deep learning techniques for speech processing,” Information Fusion, p. 101869, 2023.

[14] Kurtkaya M, “Turkish Speech Command Dataset,” https://www.kaggle.com/, 2021. [Online]. Available: https://www.kaggle.com/datasets/muratkurtkaya/turkish-speech-command-dataset/data. [Accessed: 18-Apr-2024].

[15] M. Tohyama, *Sound and signals*. Springer Science & Business Media, 2011.

[16] F. Riehle, *Frequency standards: basics and applications*. John Wiley & Sons, 2006.

[17] J. O. Smith, “Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications. 2007.” W3K Publishing, 2023.

[18] J. L. Flanagan, Speech analysis synthesis and perception, vol. 3. Springer Science & Business Media, 2013.

[19] S. A. Majeed, H. Husain, S. A. Samad, and T. F. Idbeaa, “Mel Frequency Cepstral Coefficients (MFCC) Feature Extraction Enhancement in the Application of Speech Recognition: A Comparison Study,” *Journal of Theoretical & Applied Information Technology*, vol. 79, no. 1, 2015.

[20] A. Sithara, A. Thomas, and D. Mathew, “Study of MFCC and IHC feature extraction methods with probabilistic acoustic models for speaker biometric applications,” *Procedia computer science*, vol. 143, pp. 267–276, 2018.

[21] P. Zinemanas, M. Rocamora, M. Miron, F. Font, and X. Serra, “An interpretable deep learning model for automatic sound classification,” *Electronics*, vol. 10, no. 7, p. 850, 2021.

[22] Y. Jia et al., “Speaker recognition based on characteristic spectrograms and an improved self-organizing feature map neural network,” *Complex & Intelligent Systems*, vol. 7, pp. 1749–1757, 2021.

[23] K.-H. N. Bui, H. Oh, and H. Yi, “Traffic density classification using sound datasets: an empirical study on traffic flow at asymmetric roads,” *IEEE Access*, vol. 8, pp. 125671–125679, 2020.

[24] J. Salamon and J. P. Bello, “Deep convolutional neural networks and data augmentation for environmental sound classification,” *IEEE Signal processing letters*, vol. 24, no. 3, pp. 279–283, 2017.

[25] G. James, D. Witten, T. Hastie, R. Tibshirani, and Others, An introduction to statistical learning, vol. 112. Springer, 2013.

[26] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights into Imaging*, vol. 9, pp. 611–629, 2018.

[27] K. O’shea and R. Nash, “An introduction to convolutional neural networks,” arXiv preprint arXiv:1511. 08458, 2015.

[28] A. Adeel, M. Gogate, and A. Hussain, “Contextual deep learning-based audio-visual switching for speech enhancement in real-world environments,” *Information Fusion*, vol. 59, pp. 163–170, 2020.

[29] H. Tian, S.-C. Chen, and M.-L. Shyu, “Evolutionary programming based deep learning feature selection and network construction for visual data classification,” *Information Systems Frontiers*, vol. 22, no. 5, pp. 1053–1066, 2020.

[30] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[31] G. Koppe, A. Meyer-Lindenberg, and D. Durstewitz, “Deep learning for small and big data in psychiatry,” *Neuropsychopharmacology*, vol. 46, no. 1, pp. 176–190, 2021.

[32] Y. Chen, L. Li, W. Li, Q. Guo, Z. Du, and Z. Xu, “Chapter 6 - Deep learning processors,” in AI Computing Systems, Y. Chen, L. Li, W. Li, Q. Guo, Z. Du, and Z. Xu, Eds. Morgan Kaufmann, 2024, pp. 207–245.

[33] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” *in Computer Vision--ECCV 2014: 13th European Conference*, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13, 2014, pp. 818–833.

[34] K. Santosh, N. Das, and S. Ghosh, “Deep learning: a review,” Deep Learning Models for Medical Imaging, pp. 29–63, 2022.

[35] M. M. Taye, “Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions,” *Computation*, vol. 11, no. 3, p. 52, 2023.

[36] Y. Wu, H. Mao, and Z. Yi, “Audio classification using attention-augmented convolutional neural network,” *Knowledge-Based Systems*, vol. 161, pp. 90–100, 2018.

[37] K. M. Karaoglan and O. Findik, "Enhancing Aspect Category Detection Through Hybridised Contextualised Neural Language Models: A Case Study in Multi-Label Text Classification," The Computer Journal, p. bxae004, 01 2024.
[38] F. Chollet, *Deep learning with Python*. Simon and Schuster, 2021.
[39] P. Galeone, *Hands-on neural networks with TensorFlow 2.0: understand TensorFlow, from static graph to eager execution, and design neural networks*. Packt Publishing Ltd, 2019.
[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[41] J. Jin, A. Dundar, and E. Culurciello, "Flattened convolutional neural networks for feedforward acceleration," arXiv preprint arXiv:1412. 5474, 2014.
[42] S. Ruder, "Neural transfer learning for natural language processing," NUI Galway, 2019.
[43] C. Ozdemir, "Avg-topk: A new pooling method for convolutional neural networks," *Expert Systems with Applications*, vol. 223, p. 119892, 2023.
[44] A. G. Ganie and S. Dadvandipour, "From big data to smart data: a sample gradient descent approach for machine learning," *Journal of Big Data*, vol. 10, no. 1, p. 162, 2023.
[45] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.
[46] H. Wang and H. Zheng, "Model Cross-Validation," InEncyclopedia of Systems Biology, 2013.
[47] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.
[48] G. C. Cawley and N. L. C. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *The Journal of Machine Learning Research*, vol. 11, pp. 2079–2107, 2010.
[49] Y. Jung and J. Hu, "AK-fold averaging cross-validation procedure," *Journal of nonparametric statistics*, vol. 27, no. 2, pp. 167–179, 2015.
[50] G. Jiang and W. Wang, "Error estimation based on variance analysis of k-fold cross-validation," *Pattern Recognition*, vol. 69, pp. 94–106, 2017.
[51] R. B. Rao, G. Fung, and R. Rosales, "On the dangers of cross-validation. An experimental evaluation," *in Proceedings of the 2008 SIAM international conference on data mining*, 2008, pp. 588–596.
[52] E. Bartz, T. Bartz-Beielstein, M. Zaefferer, and O. Mersmann, *Hyperparameter Tuning for Machine and Deep Learning with R: A Practical Guide*. Springer Nature, 2023.
[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412. 6980, 2014.
[54] U. A. Kimanuka and O. Buyuk, "Turkish speech recognition based on deep neural networks," *Suleyman Demirel Universitesi Fen Bilimleri Enstitusu Dergisi*, vol. 22, pp. 319–329, 2018.
[55] S. Oyucu, "A Novel End-to-End Turkish Text-to-Speech (TTS) System via Deep Learning," *Electronics*, vol. 12, no. 8, p. 1900, 2023.