International Journal of Computer Sciences and Engineering
Vol.**13**, Issue.**7**, pp.**51-57**, July **2025**
ISSN: 2347-2693 (Online)
Available online at: www.ijcseonline.org

ISSN: 2347-2693 (E)

Research Article

Evaluating the Impact of Audio Segment Duration on Transformer-Based Stuttering Detection Using Wav2Vec2

Rahul Singh[1*], **Deepti Gupta**[2]

[1,2]Dept. of Computer Science and Engineering, UIET, Panjab University, Chandigarh, India

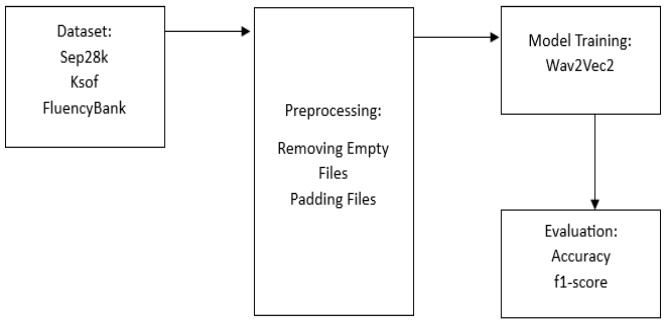
**Corresponding Author:* ✉

Received: 25/May/2025; **Accepted**: 27/Jun/2025; **Published**: 31/Jul/2025. **DOI**: https://doi.org/10.26438/ijcse/v13i7.5157

Abstract: Stuttering is a speech disorder that disrupts the fluency of verbal communication. Traditional assessment methods are subjective and labor-intensive, prompting the need for scalable, automated solutions. Recent advances in self-supervised learning and transformer-based models such as Wav2Vec2 offer promising capabilities for automated stuttering detection. This study investigates the effect of varying audio clip lengths on the classification accuracy of stuttering using Wav2Vec2 models. Experiments were conducted on three benchmark datasets—SEP-28k, FluencyBank, and KSoF—across clip durations ranging from 3 to 11 seconds. Results show that shorter audio segments (3–5 seconds) consistently achieve better classification accuracy, with a peak of 65.13% observed for 3-second segments using SEP-28k. Longer durations introduce performance variability, especially in cross-dataset evaluations. The findings support the design of efficient, real-time stuttering detection systems and inform optimal segment length for future speech analysis models.

Keywords: Stuttering detection, Speech processing, Wav2Vec2, Transformer models, Self-supervised learning, Audio segmentation, Deep learning

Graphical Abstract-



1. Introduction

Stuttering is a complex speech disorder characterized by involuntary disruptions such as repetitions, prolongations, and blocks. It affects approximately 1% of the global population and often leads to significant psychological and social burdens. Traditionally, speech-language pathologists (SLPs) rely on manual annotations and real-time listening to assess stuttering, which limits scalability and consistency. With the advancement of artificial intelligence and deep learning, automated speech recognition (ASR) has made it possible to detect disfluencies from raw audio. Transformer-based architectures, particularly Wav2Vec2[18], have shown exceptional performance in speech-related tasks due to their ability to learn rich feature representations using self-supervised learning. However, the effect of audio segment length on the performance of such models for stuttering detection remains underexplored. This study evaluates how varying the length of audio clips (3–11 seconds) impacts the performance of Wav2Vec2 in detecting stuttering across three datasets. Our contributions are: Systematic analysis of clip duration effects on model accuracy. Benchmarking across multiple real-world stuttering datasets. Insights into optimal configurations for real-time stuttering detection.

1.1 Objective of the Study

The Objective of this work is to find the impact of duration in stuttering detection using the transformer-based model Wav2Vec2.

1.2 Organization

This article is organized into the following sections which are as follows; Section 1 contains introduction, Section 2 contains related work in the field of stuttering, Section 3 contains methodology, Section 4 contains results and discussion and Section 5 concludes research work with future scope.

2. Related Work

The landscape of stuttering detection has undergone a profound transformation with the burgeoning influence of artificial intelligence, particularly deep learning, moving significantly beyond the foundational machine learning approaches that once characterized the field. Historically, methods for identifying stuttering relied on extensive feature engineering, employing acoustic characteristics such as Mel-frequency cepstral coefficients (MFCCs), linear prediction cepstral coefficients (LPCCs), fundamental pitch, shimmer, and voice onset time (VOT). These features were then fed into traditional classifiers like Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), Hidden Markov Models (HMMs), and Gaussian Mixture Models (GMMs), which demonstrated foundational effectiveness in detecting speech disfluencies but often required considerable manual effort in feature extraction and selection [1],[14].The paradigm shifted with the advent of deep learning, offering a powerful alternative by enabling models to learn intricate, hierarchical speech representations directly from raw audio data. Architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Bidirectional Long Short-Term Memory (BiLSTM) networks have emerged as pivotal tools, consistently demonstrating superior performance in various speech processing tasks, including stuttering classification [1],[11],[16]. This new wave of models has led to the development of hybrid architectures like FluentNet and StutterNet, which have further refined classification accuracy. FluentNet, for example, is designed as an end-to-end system, utilizing a Squeeze-and-Excitation ResNet to extract rich spectral features and BiLSTM layers to capture temporal relationships, further enhanced by an attention mechanism that focuses on critical speech features. It achieved state-of-the-art performance on the UCLASS corpus and introduced LibriStutter, a synthetic dataset, to combat the scarcity of real stuttered speech data [1],[13]. Similarly, StutterNet processes raw acoustic signals using a Time Delay Neural Network (TDNN) to effectively capture the temporal and contextual aspects of speech disfluencies, outperforming previous ResNet+BiLSTM models while significantly reducing computational costs [15]. Additionally, the application of BiLSTM, integrating MFCCs and phoneme probabilities, has shown strong generalization across multiple benchmark datasets, highlighting its potential for real-time speech therapy applications [16].A particularly impactful development has been the adoption of self-supervised learning, leveraging massive amounts of unlabeled audio data to pre-train highly effective models. Wav2Vec2.0 embeddings have proven instrumental, with studies reporting substantial improvements in stuttering detection (SD) accuracy. Models incorporating Wav2Vec2.0 embeddings have consistently outperformed baselines, benefiting from strategies like summing embeddings from multiple layers and concatenating them with MFCC features, which further enhances performance for traditional classifiers like SVMs [3],[5]. Fine-tuning Wav2Vec2.0 models on large stuttered speech datasets, such as SEP-28k and FluencyBank, has not only led to impressive classification improvements but also demonstrated remarkable cross-lingual transferability, showing that models trained on English stuttering data can perform effectively on German therapy speech [4]. The "Whister" approach, for instance, innovatively utilizes the hidden representations from Whisper's encoder layers for detecting and classifying stuttering events. This novel method, which trains classification heads on frozen Whisper embeddings while also incorporating MFCC features, has achieved state-of-the-art F1 scores on FluencyBank (0.70) and KSoF (0.66). A key finding from this research is that using longer audio segments (e.g., 5 seconds instead of 3) can notably improve classification accuracy, underscoring the importance of contextual window size [2]. Other advanced deep learning models, including transformer-based architectures like TranStutter, have also shown promising results by capturing complex temporal dependencies in speech signals [10].The availability and quality of datasets are paramount in training and validating these sophisticated models. Large-scale corpora like SEP-28k, which comprises over 28,000 speech clips (approximately 23 hours) curated from podcasts featuring individuals who stutter and annotated for various stuttering events (blocks, prolongations, sound repetitions, word/phrase repetitions, interjections), have become invaluable resources [8],[10],[9],[12],[15]. Other crucial datasets include FluencyBank, UCLASS, KSoF, and LibriStutter [2],[9],[12],[15]. Studies have consistently demonstrated that increasing dataset size directly correlates with substantial improvements in detection performance, with one study showing a 28% relative improvement and a 24% increase in F1-score by simply expanding the dataset [8]. Beyond data volume, researchers have explored advanced training strategies like Multi-Task Learning (MTL), where models jointly learn stuttering classification and auxiliary tasks such as speaker gender identification or metadata recognition (e.g., podcast type). MTL frameworks have shown improvements in classification for specific disfluency types like repetitions, blocks, and interjections, while also highlighting the need to address metadata entanglement [4],[6]. Adversarial Training (ADV) has also been employed to learn podcast-invariant speech representations, making models more robust to speaker variations and improving the detection of fluent speech [6].Despite these remarkable advancements, automated stuttering detection faces several persistent and complex challenges. Limitations in dataset size and quality, particularly concerning representative real-world speech, remain a significant hurdle. Data imbalance, where certain disfluency types are underrepresented, often leads to biased model performance [1],[2],[3],[6],[7],[8],[9],[12],[16]. High computational costs associated with training deep learning models, especially large transformer-based ones, pose a practical constraint for real-time deployment and broader accessibility [1],[11]. Furthermore, challenges in achieving robust domain generalization, managing real-world noise variations, and a lack of standardized evaluation protocols hinder direct comparisons and broader applicability of research findings [1],[2],[7],[8],[9], [14],[16]. Specific disfluency types, such as word repetitions and "garbage" disfluencies, continue to be particularly challenging for models to accurately detect [3],[9]. The discrepancies in performance across different stuttering subclasses and the

impact of dataset partitioning (where overlapping speakers between training and test sets can lead to over-optimistic results) also require careful consideration [4],[9].Looking ahead, future research aims to address these critical issues and push the boundaries of AI-driven stuttering detection. Key directions include focusing on multimodal learning, which integrates information from various modalities beyond just audio; further optimizing self-supervised transformer-based models for enhanced accuracy and interpretability; incorporating multilingual support to broaden the applicability of detection systems; and rigorously developing larger, publicly available, and more diverse datasets to improve generalization and real-time deployment capabilities [1],[2],[7],[9], [10],[12],[13],[15],[16]. Refining multi-class learning strategies to handle overlapping stuttering patterns, enhancing model adaptability across multiple datasets, and fine-tuning models for detecting stuttering locations in speech frames are also crucial areas of focus [4],[5],[7],[12],[15]. This ongoing research underscores the transformative potential of AI in providing better speech therapy and assessment tools, ultimately enhancing the quality of life for individuals who stutter and improving ASR systems tailored to their unique speech patterns.

3. Methodology

3.1Datasets

The study utilizes three datasets for stuttering detection:Sep28k: A large-scale dataset containing stuttered speech samples from podcasts.

The Sep28k dataset is a large-scale, publicly available corpus specifically designed for the detection of stuttering events in spontaneous speech. Developed from English-language podcasts featuring people who stutter, it contains over 28,000 labeled audio clips annotated across various disfluency types, including blocks, prolongations, repetitions, and interjections. Each clip is carefully segmented and labeled by multiple human annotators to ensure accuracy and reliability. Sep28k, a critical resource for training and evaluating ML models aimed at automatic stuttering detection, providing diverse and naturally occurring speech patterns essential for robust model development [8].

FluencyBank: A dataset specifically curated for studying speech fluency, used for testing and validation in some experimental setups. The FluencyBank dataset is a comprehensive resource developed as part of the TalkBank project, aimed at studying fluency and disfluency patterns across a wide range of speakers. It includes recordings of both people who stutter and fluent speakers, covering various ages, backgrounds, and speaking contexts. The dataset contains detailed transcriptions and annotations for different types of speech disruption, such as repetitions, prolongations, interjections and blocks. FluencyBank is valuable not only for stuttering research but also for broader investigations into language development, speech disorders, and fluency assessment. Its rich linguistic and acoustic information makes it an important tool for training and evaluating speech recognition and stuttering detection models [8].Ksof: The Kassel State of Fluency (KSoF) dataset is a specialized speech corpus designed for stuttering research, particularly focused on therapy-related speech patterns. It contains over 5,500 audio clips recorded during different stages of therapy at the Kasseler Stottertherapie institute. The dataset is labeled across six key disfluency types—such as blocks, repetitions, prolongations and modified speech techniques taught during therapy. KSoF stands out by offering recordings from people who underwent intensive speech therapy, providing a unique resource for studying both natural stuttering behaviors and therapy-induced fluency changes. It supports advancements in automated stuttering detection and complements broader datasets like Sep28k by emphasizing therapy-centered speech dynamics [17].

3.2.1 Data Splitting Strategies

The corpus were split into training, validation, and test sets using different approaches: Custom Test and Validation Sets: In some experiments, 3,000 samples were allocated for both validation and test sets, with the remaining 23,000 samples used for training.10% Test and 9% Validation Split: A standard random split ensuring a balanced division of training, validation, and test data. FluencyBank as Test Set: In specific configurations, FluencyBank was used exclusively as the test set, while Sep28k served as the training dataset. The KSoF dataset is a specialized speech corpus designed for stuttering research, particularly focused on therapy-related speech patterns. It contains over 5,500 audio clips recorded during different stages of therapy at the Kasseler Stottertherapie institute. The dataset is labeled across six key disfluency types—such as blocks, repetitions, prolongations and modified speech techniques taught during therapy. It is also used as a test set.

3.3 Preprocessing

3.3.1 Audio Processing

Resampling: All audio clips were resampled to 16,000 Hz. Blanks**:** Removing audio files which don't have any data. Trimming & Silence Removal: Ensured uniform clip lengths and removed excessive silent regions. Augmentation Techniques (if applied): Adding background noise, Time-stretching (speed variations), Pitch shifting.

3.3.2 Label Processing

Encoding categorical labels using LabelEncoder. Ensuring consistent class distribution across training, validation, and test splits.

3.4 Model Selection: Wav2Vec2 for Stuttering Detection

The study utilizes Wav2Vec2, a transformer-based speech recognition model, for feature extraction and classification. The model architecture consists of: Audio Input (16kHz, 3 sec) → Wav2Vec2 Feature Extractor → Transformer Layers → Fully Connected Layer → Softmax (Classification)

3.5 Model Training Process

3.5.1 Training Pipeline

Model: Wav2Vec2ForSequenceClassification from the Hugging Face Transformers library, Optimizer: AdamW,

Loss Function: CrossEntropyLoss. Training Strategy: Mixed Precision Training using torch.cuda.amp to reduce memory usage. Gradient Accumulation for large batch training. Dynamic Learning Rate Adjustment with "ReduceLROnPlateau".

3.5.2 Hyperparameter Settings

The model was trained using a batch size of 16 or 32, with a learning rate of 1e-5. AdamW was chosen as the optimizer, along with the ReduceLROnPlateau scheduler that adjusts the learning rate based on the validation loss. Training was carried out for 15 to 20 epochs to ensure stable convergence.

3.5.3 Checkpointing and Early Stopping

The best model was saved based on validation accuracy. Training was stopped early if validation loss did not improve for a fixed number of iterations. Lowering the learning rate from 1e-5 did not bring any improvements at all.

3.6 Model Evaluation

To assess model performance, the best-trained model was assessed on the test set. The following parameters were computed: Test Loss (CrossEntropyLoss) In order to evaluate as well as optimize model's training and testing data loss, CE loss function was used. CE loss function considers differences between probabilities of the actual and predicted data values and if they closely match with each other then the value of loss function would be less else value would be very high. Mathematically, CE Loss function could be written as in equation 3.1

$$L = -\sum_{i=1}^{N} \sum_{j=1}^{C} y_{i,j} \log(\hat{y}_{i,j} \quad (3.1)$$

Accuracy is the ratio of sum of all the TP as well as TN values determined values over sum of all the TP, TN, FN and FP values. Mathematically, accuracy could be written as in equation 3.2

$$Accuracy = (TP + TN)/(TP + TN + FP + FN \quad (3.2)$$

Precision is the ratio of sum of all the TP over sum of all the TP and FP values. Mathematically, precision could be written as in equation 3.3

$$Precision = TP/(TP + FP) \quad (3.3)$$

Recall is the ratio of sum of all the TP values over sum of all the TP and FN values. Mathematically, recall could be written as in equation 3.4

$$Recall = TP/(TP + FN) \quad (3.4)$$

F1-score is the ratio of product of precision and recall over sum of precision and recall values. Mathematically, recall could be written as in equation 3.5

$$F1 = 2 * (Precision * Recall)/(Precision + Recall \quad (3.5)$$

3.7 Hardware and Computational Resources

3.7.1 Hardware Setup

GPU: NVIDIA Tesla T4 (15 GB RAM) on Kaggle. Training Time: 5-10 hours per experiment. Libraries Used: PyTorch, Hugging Face Transformers. Dell Inspiron 15 5518 Intel Core i5 11320H 16GB 512GB SSD

3.7.2 Handling Memory Constraints

To prevent memory overflow issues: Mixed Precision Training (torch.cuda.amp) was utilized. Gradient Accumulation was used to simulate larger batch sizes. GPU Memory Clearing (torch.cuda.empty_cache()) was performed between epochs.

4. Results and Discussion

Here we discuss and analyze the outcomes of stuttering detection experiments conducted using transformer-based models on Sep28k, FluencyBank, and KSoF datasets. It compares performance across different input durations, dataset combinations, and validation strategies. Key findings are discussed to highlight trends, strengths, and limitations, offering insights into model behavior and the impact of data configurations on classification accuracy.

Table 4.1 Accuracy Obtained for Training set- Sep28k Test set- Sep28k for different audio duration

Dataset	Audio Duration (sec)	Accuracy (%)
Sep28k	3	63.91
Sep28k	5	64.26
Sep28k	7	64.49
Sep28k	9	62.49
Sep28k	11	62.72

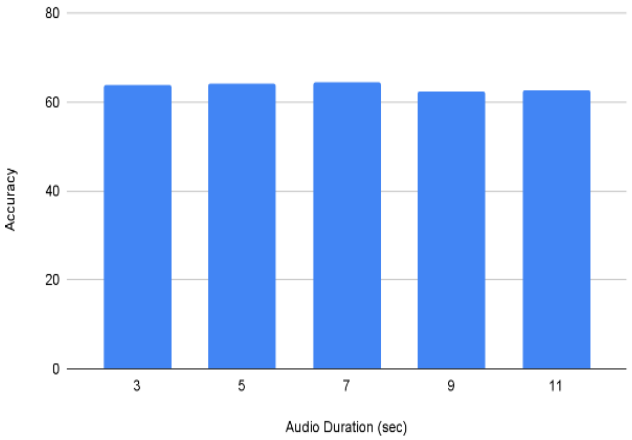


Figure 4.1 Training, Validation- sep28k, Test Set- sep28k

Observation: The table presents a comparative evaluation of model performance across varying audio durations, using the Sep28k dataset. The training set remains constant in each configuration, while 10% of the data is reserved for testing and 9% for validation. Audio durations were varied from 3 seconds to 11 seconds in steps of 2 seconds to observe the effect on classification accuracy. The highest accuracy (64.49%) was achieved when using 7-second audio segments, suggesting this duration may provide an optimal balance between contextual richness and noise. A slight performance

dip was observed for longer segments (9 and 11 seconds), with accuracies dropping to 62.49% and 62.72%, respectively. Using only 3-second clips achieved a competitive accuracy of 63.91%, indicating that even short segments contain sufficient stuttering cues for reliable detection. These results highlight that moderate-length audio segments (5–7 seconds) are most effective for stuttering classification tasks using the Sep28k dataset (Table 4.1)(Figure 4.1).

Table 4.2 Accuracy Obtained for Training set- Sep28k Test set- Ksof for different audio duration

Dataset	Audio Duration (sec)	Accuracy (%)
Sep28k + Ksof	3	45.16
Sep28k + Ksof	5	47.74
Sep28k + Ksof	7	48.07
Sep28k + Ksof	9	47.43
Sep28k + Ksof	11	45.47

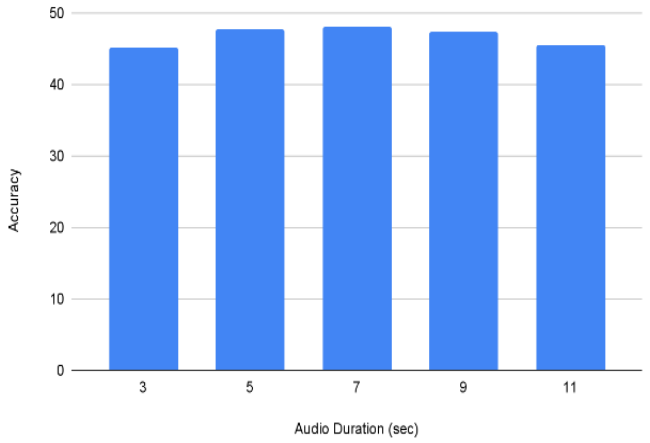


Figure 4.2 Training, Validation- sep28k, Test Set- ksof

Observation: The model achieved the highest accuracy of 48.07% with 7-second audio segments. A steady improvement in accuracy is observed from 3 seconds (45.16%) to 7 seconds. After 7 seconds, accuracy slightly drops at 9 seconds (47.43%) and 11 seconds (45.47%). Moderate-length audio (5–7 seconds) provides better context for the model, enhancing prediction capability. Longer segments (9–11 seconds) may introduce background noise or unrelated information, slightly degrading performance. Incorporating KSoF introduces valuable speech variability (e.g., modified speech due to therapy). However, it also brings additional complexity that may require refined preprocessing or model adaptation to prevent performance drop (Table 4.2)(Figure 4.2).

Table 4.3 Accuracy Obtained for Training set- Sep28k Test set- FluencyBank for different audio duration

Dataset	Audio Duration (sec)	Accuracy (%)
Sep28k + FluencyBank	3	63.61
Sep28k + FluencyBank	5	62.26
Sep28k + FluencyBank	7	62.26
Sep28k + FluencyBank	9	61.54
Sep28k + FluencyBank	11	61.59

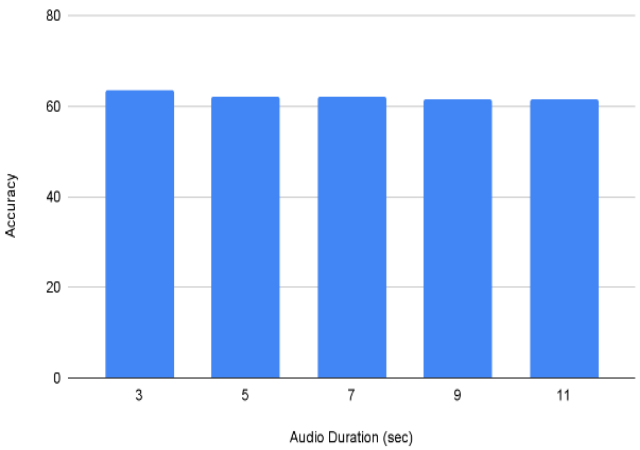


Figure 4.3 Training, Validation- sep28k, Test Set- fluencybank

Observation: The highest accuracy of 63.61% was achieved using 3-second audio segments. A slight decline in performance was observed with increasing audio durations. Shorter audio segments (3 seconds) appear optimal for capturing dysfluency-specific features. Longer segments may introduce irrelevant or redundant information, which can dilute the model's ability to focus on stuttering cues. Segment duration is a crucial hyperparameter when combining multiple datasets with varying speech characteristics. The results suggest that when using Sep28k + FluencyBank, shorter inputs are more effective for stuttering detection tasks. Optimal accuracy is achieved with 3-second clips, reinforcing prior observations that concise segments enhance detection performance when using diverse, naturalistic datasets(Table 4.3)(Figure 4.3).

Sep28k consistently achieves the highest accuracy across time durations (peaking at 0.6449 for 7 seconds).Sep28k + FluencyBank performs slightly lower but remains close to Sep28k (best at 0.6493 for 3 seconds).Sep28k + Ksof performs significantly lower than the other two combinations, with accuracy ranging between 0.45–0.48.3-second clips provide the highest accuracy, with 7-second clips performing well in some cases. Using FluencyBank for testing results in slightly lower accuracy, likely due to dataset variability. Shorter clips (3s-5s) consistently outperform longer clips (9s-11s), suggesting clearer speech segmentation. A 10% test and 9% validation split with Sep28k alone yields better performance compared to setups using FluencyBank for testing.Sep28k (alone) gives the most consistent and high-performing results across durations.Sep28k + FluencyBank is competitive, especially at shorter durations (3s).Sep28k + Ksof underperforms, likely due to data quality or compatibility issues. Short to medium clip lengths (3–7s) appear optimal for accuracy, with longer durations offering no added benefit and possibly introducing noise. This chapter presented the results and discussion based on various experimental setups. The analysis highlights the impact of dataset variations, clip duration, and batch size on the model's performance.

5. Conclusion and Future Work

This study explores the application of ML and DL techniques for automatic stuttering detection. Various datasets, including Sep28k and FluencyBank, were utilized to evaluate the impact of different test-validation splits, clip durations, and dataset combinations on classification accuracy. The results indicate several key findings: Shorter clips (3s-5s) consistently outperform longer clips (9s-11s), suggesting that shorter segments provide clearer speech patterns for classification. Using FluencyBank for testing resulted in slightly lower accuracy, likely due to dataset variability and differences in speech recording environments. A 10% test and 9% validation split with Sep28k alone yielded better performance compared to setups using FluencyBank for testing. The highest accuracy (65.13%) was achieved using Sep28k with 3-second clips, while longer clips tended to reduce classification accuracy. Hybrid dataset setups (Sep28k + FluencyBank) showed competitive results, but their performance varied based on the test-validation approach. Overall, the study demonstrates that dataset selection, clip duration, and test-validation strategies significantly impact the accuracy of stuttering detection models. These insights are valuable for optimizing ML models for real-world speech disorder applications.

Future research should focus on: Developing Bigger and More Varies Datasets**:** To enhance generalization across various speakers and languages. Improving Domain Adaptation**:** Utilizing transfer learning and semi-supervised learning to leverage existing ASR models. Integrating Multimodal Approaches**:** Combining speech, facial expressions, and textual cues for enhanced classification. Optimizing Real-Time Deployment**:** Developing lightweight models for mobile applications and speech therapy tools. While the study achieved promising results, several challenges remain, paving the way for future research directions: Data Augmentation**:** Implementing advanced augmentation techniques such as time-stretching, pitch shifting, and noise injection could enhance model robustness and improve accuracy. Multimodal Approaches**:** Future work could integrate audio, video, and text-based features to develop a more comprehensive stuttering detection system. Real-Time Deployment**:** Optimizing models for low-latency inference could enable real-time applications in speech therapy and assistive technology. Transfer Learning and Pretrained Models**:** Leveraging large-scale self-supervised speech models such as Whisper and Wav2Vec2 could improve generalization across diverse datasets. Handling Dataset Variability**:** More research is needed to improve cross-dataset generalization and reduce the performance gap when testing on datasets like FluencyBank. Ethical Considerations and Bias Reduction**:** Ensuring that models are trained on diverse datasets to minimize bias and improve inclusivity for different speech patterns and demographics. By addressing these challenges, future studies can develop more robust, efficient, and scalable solutions for automatic stuttering detection, contributing to the advancement of AI-driven speech disorder research and applications.

Author's Contribution

Rahul Singh: carried out the research based on the different inputs and tried out various experiments under the supervision

Deepti Gupta: offered theoretical insights, carefully examined and rewrote the work for key intellectual elements and gave her approval before submitting the finished version.

Acknowledgements

This Research has received no external funding.

References

[1] Shakeel A. Sheikh, Md Sahidullah, F. Hirsch, and S. Ouni, "Machine learning for stuttering identification: Review, challenges and future directions", *Neurocomputing*, Vol.**514**, pp.**385-402, 2022**. doi: 10.1016/j.neucom.2022.10.015.

[2] V. Changawala and F. Rudzicz, "Whister: Using Whisper's representations for Stuttering detection", in *Proc. Interspeech* **2024**.

[3] S. A. Sheikh, M. Sahidullah, F. Hirsch, and S. Ouni, "End-to-End and Self-Supervised Learning for ComParE 2022 Stuttering Sub-Challenge", *arXiv preprint*, arXiv:2207.10817, **2022**. doi: 10.48550/arXiv.2207.10817

[4] S. P. Bayerl, D. Wagner, E. Nöth, and K. Riedhammer, "Self-supervised learning for stuttering detection: Challenges and opportunities", *arXiv preprint*, arXiv:2204.03417, **2022**. doi: 10.48550/arXiv.2204.03417.

[5] S. A. Sheikh, M. Sahidullah, F. Hirsch, and S. Ouni, "Advances in Stuttering Detection: Exploring Self-Supervised and End-to-End Learning Approaches", *arXiv preprint*, arXiv:2204.01564, **2022**. doi: 10.48550/arXiv.2204.01564

[6] S. A. Sheikh, M. Sahidullah, F. Hirsch, and S. Ouni, "Robust Stuttering Detection via Multi-task and Adversarial Learning", *arXiv preprint*, arXiv:2204.01735, **2022**. doi: 10.48550/arXiv.2204.01735.

[7] R. Alnashwan, N. Alhakbani, A. Al-Nafjan, A. Almudhi, and W. Al-Nuwaiser, "Computational Intelligence-Based Stuttering Detection: A Systematic Review", *Diagnostics*, Vol.**13**, No.**23**, pp.**35-37, 2023.** doi: 10.3390/diagnostics13233537.

[8] C. Lea, V. Mitra, A. Joshi, S. Kajarekar and J. P. Bigham, "SEP-28k: A Dataset for Stuttering Event Detection from Podcasts with People Who Stutter", ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, pp.**6798-6802, 2021.** doi: 10.1109/ICASSP39728.2021.9413520

[9] P. Filipowicz and B. Kostek, "Rediscovering Automatic Detection of Stuttering and Its Subclasses through Machine Learning—The Impact of Changing Deep Model Architecture and Amount of Data in the Training Set", *Applied Sciences*, Vol.**13**, No.**10**, pp.**6192**, **2023**. doi: 10.3390/app13106192.

[10] Basak, K.; Mishra, N.; Chang, H.-T. TranStutter: A Convolution-Free Transformer-Based Deep Learning Method to Classify Stuttered Speech Using 2D Mel-Spectrogram Visualization and Attention-Based Feature Representation. *Sensors*, *23*, 8033, **2023**. doi.org/10.3390/s23198033

[11] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh and K. Shaalan, "Speech Recognition Using Deep Neural Networks: A Systematic Review", in IEEE Access, Vol.**7**, pp.**19143-19165, 2019**. doi: 10.1109/ACCESS.2019.2896880.

[12] T. Kourkounakis, A. Hajavi, and A. Etemad, "Detecting Multiple Speech Disfluencies Using a Deep Residual Network with Bidirectional Long Short-Term Memory", *arXiv preprint*, arXiv:1910.12590, **2019**. doi: 10.48550/arXiv.1910.12590

[13] T. Kourkounakis, A. Hajavi, and A. Etemad, "FluentNet: End-to-End Detection of Stuttered Speech Disfluencies With Deep Learning", *arXiv preprint*, arXiv:2009.11394, **2020**. doi: 10.48550/arXiv.2009.11394

[14] S. Khara, S. Singh and D. Vir, "A Comparative Study of the Techniques for Feature Extraction and Classification in Stuttering", *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, India, pp.**887-893, 2018.** doi: 10.1109/ICICCT.2018.8473099

[15] S. A. Sheikh, M. Sahidullah, F. Hirsch, and S. Ouni, "StutterNet: Stuttering Detection Using Time Delay Neural Network", *arXiv preprint*, arXiv:2105.05599, **2021**. doi: 10.48550/arXiv.2105.05599

[16] Jouaiti, Melanie & Dautenhahn, Kerstin. Dysfluency Classification in Stuttered Speech Using Deep Learning for Real-Time Applications, **2022**. 10.1109/ICASSP43922.2022.9746638.

[17] S. P. Bayerl, A. Wolff von Gudenberg, F. Hönig, E. Noeth, and K. Riedhammer",KSoF: The Kassel State of Fluency Dataset – A Therapy Centered Dataset of Stuttering", in *Proceedings of the Language Resources and Evaluation Conference*, Marseille, France: European Language Resources Association, pp.**1780–1787, 2022.**

[18] K. Floridi and M. Chiriatti, "GPT-3: Its Nature, Scope, Limits, and Consequences", *arXiv preprint arXiv:2006.11477*, Sep. **2020**. doi.org/10.48550/arXiv.2006.11477

AUTHORS PROFILE

Rahul Singh earned his B.E. in Computer Science and Engineering from Chandigarh University Panjab India 2022. M.E. in Computer Science and Engineering from University Institute of Engineering and Technology from Panjab University Chandigarh India 2025.

Deepti Gupta received her BE in Computer Science and Engineering from University of Jammu, Jammu and Kashmir, India in 2006; MTECH in Computer Science and Engineering from National Institute of Technology, Jalandhar, Punjab, India in the year 2009 and PhD in Computer Science and Engineering from National Institute of Technology, Jalandhar, Punjab, India in the year 2015. She worked as Assistant Professor in the department of Computer Science and Engineering, National Institute of Technology, Delhi in the year 2014. She is currently working as Assistant Professor in the department of Computer Science and Engineering, University Institute of Engineering and Technology, Panjab University, Chandigarh, India. Her professional research activity lies in the field of wireless sensor networks and data mining. She has published 20 research papers in the International Journals/Conferences. She has supervised 06 M.E theses and is currently supervising 03 PhDs. She is Life Member of Advanced Computing & Communications Society, Indian Institute of Science, Bangalore, India, (L6233A1523472) and Indian Society for Technical Education (I.S.T.E.), New Delhi, India, (LM 110527).

