

Transformer-Based Online Urdu Handwritten Text Recognition Using Spatial-Temporal Stroke Data and Real-Time Learning API

Sameen Khalid¹, Talha Waheed¹, Hina Khalid², Abdul Jaleel^{3,*}

Department of Computer Science University of Engineering and Technology, Lahore, 54890, Pakistan
Department of Computer Engineering, University of Engineering and Technology, Lahore 54890, Pakistan
Department of Computer Science, Rachna College of University of Engineering and Technology, Lahore, 52200, Pakistan

sameenkhalid896@gmail.com, twaheed@uet.edu.pk, hinakhalid@uet.edu.pk

^{3,*} Corresponding e-mail: abduljaleel@uet.edu.pk

Abstract

Online handwritten text recognition (HTR) is a challenging task, especially for cursive, script-rich languages like Urdu and Arabic, because of contextual shaping and bidirectional writing. This work presents a transformer-based framework using real-time spatial-temporal stroke information (coordinates, velocity, direction) acquired using a specially designed mobile application. The data were collected, pre-processed, and then traditional machine learning (SVM, HMM), deep learning (CNN-LSTM, Bi-LSTM), and transformer models were compared based on a novel dataset of 45,000 Urdu word instances. Experiments demonstrate that transformer-based techniques are superior to other methods, yielding a 3.2%-character error rate and a 7.8%-word error rate. A RESTful API supports real-time prediction and model update. Contributions are: (1) a deployment-ready pipeline for Urdu HTR, (2) empirical verification of spatial-temporal features, and (3) a publicly available dataset. The work fills gaps in online Urdu HTR by integrating real-time data fetching, transformer-based modeling, and deployment scalability.

Index Terms—Online Handwriting Recognition, Spatial-Temporal Data, Transformers in AI, Urdu Handwritten Text Recognition.

I. INTRODUCTION

Handwritten text recognition (HTR) became popular in the last few years as it found various applications in digitization, document processing, and human-computer interaction. Online handwritten text recognition, particularly for Urdu, poses special challenges because it's cursive, shaped by context, carries diacritic marks, and is bidirectionally written. Online HTR makes use of spatial-temporal data captured in real-time, such as stroke sequence, pen pressure, and velocity, whereas offline HTR depends on static images of handwritten text. Deep learning has led to an evolution in HTR, and online HTR uses spatial-temporal data to improve accuracy by capturing handwriting as it is written, stroke by stroke, instead of from an image. It is particularly helpful for real-time recognition, which makes it applicable to applications such as smart pens, digital notetaking, and signature verification. By enabling smooth and simple writing experiences on tablets, smartboards, and AI-powered handwriting assistants, it can facilitate human-computer interaction.

The HTR process is improved by the spatial-temporal data, which captures motion over time. The stylus or pen's movement, curvature, pressure, and stroke direction are recorded by the spatial component (position, shape). An example spatial-temporal data representation is illustrated in Fig. 1. To detect word edges and character boundaries, the Temporal Component (Timing, Sequence) records acceleration, delays, and changes in speed. Here, the system learns from the handwriting's organic progression to accurately classify strokes instead of static pixels. Urdu is still underutilized, particularly in

terms of online recognition, despite HTR's success in English and Chinese. This paper presents a comprehensive end-to-end framework that includes transformer-based model training, real-time prediction, and data collection via a mobile app.

Conventional ML methods like Hidden Markov Models (HMMs) and Support Vector Machines (SVMs) have been commonly employed for HTR but fall short when dealing with complicated scripts. Deep learning (DL) methods, in the form of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have proved to be highly successful in extracting spatial and sequential relationships in handwriting. Recently, transformer-based models have proved to be highly effective in sequence modeling tasks, facilitating contextual perception and long-range dependencies in handwritten text recognition.

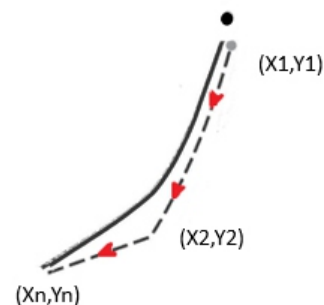


Fig. 1. Spatio-temporal data representation of Urdu handwriting strokes. Each stroke S is represented as a sequence of coordinates with timestamps: $S = [(x_1, y_1, t_1), (x_2, y_2, t_2), \dots, (x_n, y_n, t_n)]$.

This research aims to evaluate the effectiveness of ML, DL, and transformers in online Urdu handwritten text recognition by leveraging spatial-temporal data. We analyze the impact of feature extraction techniques, neural network architectures, and sequence modeling strategies to optimize recognition accuracy. Our contributions include:

- A comprehensive evaluation of traditional ML, DL, and transformer-based models for online Urdu HTR.
- A novel dataset collected via a mobile app, encompassing diverse Urdu handwriting styles.
- Empirical evidence of the superiority of transformer models in handling spatial-temporal dependencies.

The organization of the paper continues with the following structure. Section II presents a concise literature review about existing studies; Section III presents the proposed framework. Section IV describes the experimental setup and procedure, whereas Section V presents the details of the experimental results. The study is concluded in section V.

II. LITERATURE REVIEW

Handwriting recognition of text has come a long way since its inception, with conventional ML methods being followed by contemporary DL and transformer-based architectures. The initial work on Urdu HTR largely depended on statistical techniques like Hidden Markov Models (HMMs) and Support Vector Machines (SVMs). For example, Razzak et al. (2010) presented a fuzzy logic and hybrid HMM system for online Urdu character recognition, resulting in modest accuracy but struggling with intricate ligatures and contextual shaping [1]. Likewise, Sagheer et al. (2010) used SVMs for holistic recognition of Urdu words, albeit their performance was compromised by the requirement for heavy feature engineering and sensitivity to variation in writing [2].

The onset of deep learning made Urdu HTR much more efficient with the automation of feature extraction and better sequence modeling. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) networks, gained favor due to their capabilities in spatial and temporal dependencies. Ahmed et al. (2019) showcased the use of Bidirectional LSTMs (BiLSTMs) in Urdu character recognition with the utilization of contextual knowledge through forward and backward sequence processing [3]. Hybrid structures, like CNN-LSTM designs, improved performance even further through the synergy of spatial feature learning and sequential modeling, as demonstrated by Javed et al. (2017) in their application to Urdu ligature classification [4]. These methods produced significant accuracy gains, with certain reports claiming up to 85% Urdu script recognition rates. Even with such developments, RNN-based models struggled with vanishing gradients and long-range dependence issues, especially in cursive scripts like Urdu.

This brought about the investigation of transformer-based structures, and the introduction of the self-attention mechanism helped model contextual relations more efficiently. Vaswani et al. (2017) [5] first introduced transformers for sequence modeling, and later adjustments like Transformer-XL (Dai et al., 2019 [6]) proved better performance in the handling of long sequences. Recent studies have

Table I Comparative Analysis of Urdu Handwritten Text Recognition Approaches

Reference	Methodology	Dataset	Key Metrics	Limitations
Razzak et al. (2010) [1]	HMM + Fuzzy Logic	Custom (Online)	Accuracy: 78%	Struggles with ligatures
Sagheer et al. (2010) [2]	SVM (Holistic)	UPTI (Offline)	WER: 21%	Feature engineering overhead
Ahmed et al. (2019) [3]	BiLSTM	CENPARMI-Urdu	CER: 8.5%	Limited to isolated characters
Javed et al. (2017) [4]	CNN-LSTM	Private Ligatures	Accuracy: 82%	Small dataset size
Rehman et al. (2023) [8]	Transformer-XL	UHTD (Online)	CER: 4.1%	Computational cost
Riaz et al. (2022) [10]	Conv-Transformer	Urdu-Nastaleeq	WER: 9.3%	Offline-only
Hamza et al. (2024) [11]	ET-Network (Transformer)	Multi-Writer	CER: 3.7%	Requires GPU acceleration

Table II compares the studies on online HTR with Year, Accuracy, Script and approach used parameters.

extended transformers to HTR, including Vision Transformers (ViTs) by Dosovitskiy et al., 2020 [7], which treat strokes as patches for spatial-temporal learning. For Urdu specifically, Rehman et al. (2023) [8] highlighted the potential of transformers in overcoming the challenges of cursive and context-sensitive scripts, though comprehensive evaluations remained scarce. Husnain et al. (2021) [9] utilized deep networks and visualization techniques for Urdu character recognition, whereas Riaz et al. (2022) [10] introduced a Conv-Transformer model for offline Urdu handwriting with strong performance. Hamza et al. (2024) [11] recently presented ET-Network, an effective transformer model specifically designed for Urdu handwritten text recognition, demonstrating the versatility of transformers to complex scripts. But as pointed out by AlKendi et al. (2024) [12], issues like small datasets, variant writing styles, and requirements of multilingual pre-trained models remain.

A research matrix is given in Table I, depicting the comparative study of Urdu Handwritten Text Recognition methods. It gives a brief comparative study of important studies in Urdu Handwritten Text Recognition in terms of methodology (e.g., HMM, SVM, BiLSTM, Transformers), type of dataset (online/offline), evaluation metrics (CER, WER, accuracy), and limitations.

Table II Studies on online HTR with Year, Accuracy, Script and Approach Used parameters.

Year	Accuracy	Script	Approach Used
GHOSH et. al., 2021 [13]	(TPR) of 94.63% with 4.1% (FAR)	N/A	ST-SNN
VALIKH ANI et. al., 2020 [14]	digits with 92.43% accuracy and isolated letters and digits with 97.52% accuracy	Persian /Arabic	DFSA+NDFA
MUKHE RJEE et. al., 2019 [15]	N/A	Indic	LSTM+HMM
FARD et. al., 2019 [16]	Persian/Arabic letters and digits with a 92.43% accuracy and isolated letters and digits with 97.52% accuracy	Persian /Arabic	Genetic programming-based decision tree (GPDT) + non-deterministic finite automata (NDFA)

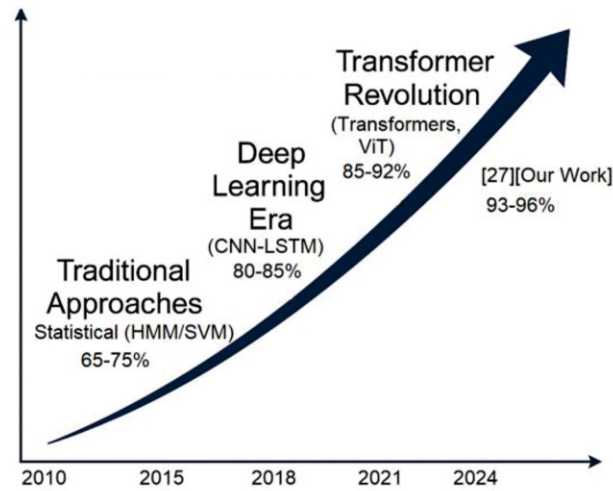


Fig. 2: Evolution of Urdu HTR Approaches

Existing studies focus on traditional approaches using hidden Markov models and support vector machines as early attempts, but face limitations with contextual ligatures and variability. In deep learning, CNNs and RNNs improved recognition rates. CNN-LSTM hybrids achieved up to 82% accuracy on Urdu datasets. Bi-LSTM further enhanced contextual understanding. Transformers introduced self-attention, enabling better stroke sequence modeling. However, few studies have focused on Urdu using transformers with spatial-temporal data. Fig. 2 presents the evolution of Urdu HTR approaches, highlighting the paradigm shift from handcrafted features to deep learning and transformer-based methods.

The existing research lacks real-time Urdu HTR datasets, complete end-to-end systems, and transformer-based evaluations. A summary of online datasets is tabularized in Table III.

III. PROPOSED FRAMEWORK FOR ONLINE URDU HANDWRITTEN TEXT RECOGNITION

To solve the problems of online Urdu handwritten text recognition (OUHTR), which is context-sensitive and has cursive scripts, we introduce a complete real-time recognition framework. The designed

Table III Summary of online datasets

Ref	Samples	Unit of Recognition	Online/Offline	Words Type
ALYAFEAI et.al. [17]	2500	Sentences	Online	Arabic
HESHMAT et.al. [18]	2520	characters	Online	Arabic
SHAMAILE H et.al. [19]	2000	N/A	both	Arabic
ABDELAZI Z et.al. [20]	21201, 42967, 15004	Segmented character, unique words	Online	Arabic
EL ABED et.al. [21]	20,575	words	Online	Arabic
ELANWAR et.al. [22]	3800, 19,400	Words, characters	Online	Arabic
OTT et.al.[23]	31275	characters	Online	English
Our dataset	4500	words	Online	Urdu

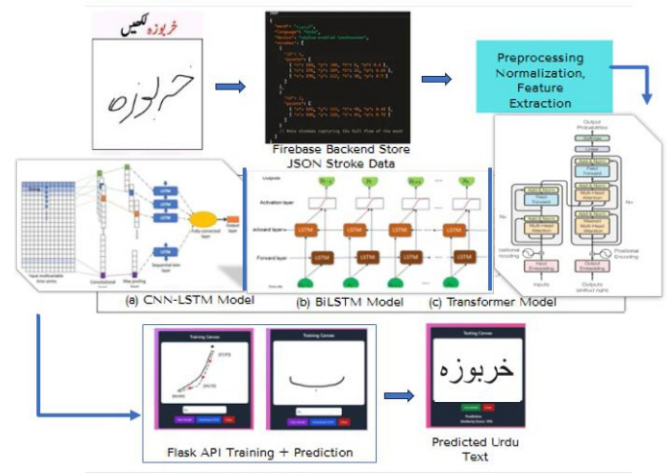


Fig. 3: End-to-end architecture of the proposed online Urdu handwritten text recognition system, showing the core modules

system combines mobile data collection, spatial-temporal feature processing, and sequence-based deep learning models, finally leading to a deployable API for live handwriting inferences, as shown in Fig. 3.

The framework is comprised of four interconnected modules:

1. Mobile Handwriting Data Collection App
2. Stroke Data Preprocessing and Feature Engineering
3. Model Training (ML, DL, Transformer Architectures)
4. Prediction and Deployment via RESTful API

Each of these modules has a specific task in constructing an end-to-end pipeline from raw handwriting input to output recognized Urdu text.

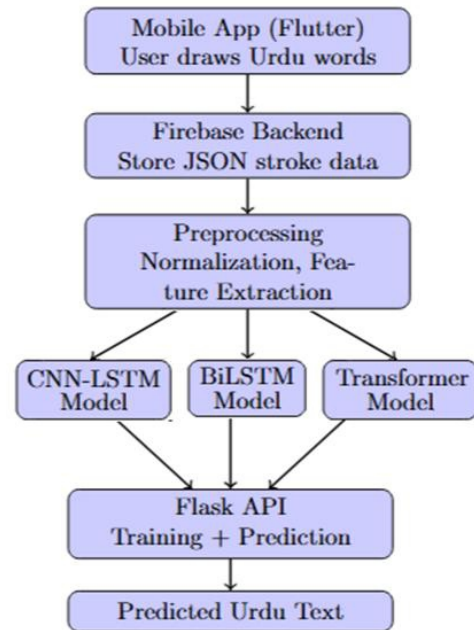


Fig. 4: Interactive workflow for the proposed framework for recognizing Urdu handwritten text from stroke sequences using deep learning models via a mobile-to-API pipeline.

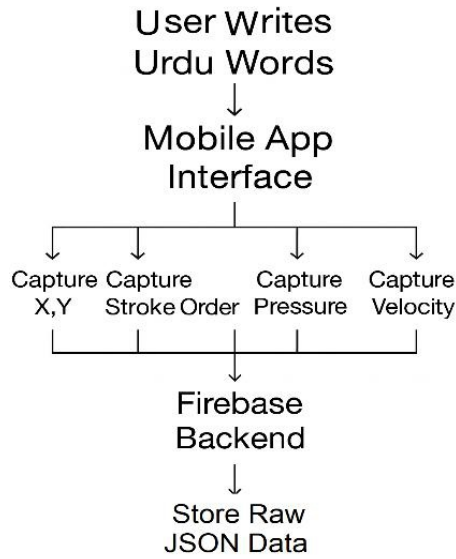


Fig. 5 Flowchart illustrating the mobile-based data acquisition pipeline for Online Urdu handwriting, capturing spatial and dynamic stroke features and storing them as raw JSON data via Firebase.

Fig. 4 illustrates the workflow of the proposed framework, and the modules involved are described next.

Module 1: Mobile Handwriting Data Collection

A light-weight mobile app was built with Flutter to gather online handwriting samples. The application prompts users to enter personal metadata (name, registration number) and then trace predefined Urdu words on the screen using touch or stylus input. The particulars are illustrated in Fig. 5.

- Word is repeated five times to maintain data consistency
- Stroke-level data (x, y coordinates, and timestamp) are gathered in real time
- Data is stored in JSON format within the Firebase store, enabling scalable and structured collection

This module enables crowd-sourced collection of diverse handwriting samples, forming the backbone of the dataset.

Module 2: Spatial-Temporal Preprocessing

The collected JSON stroke data is processed using custom Python scripts that extract temporal and spatial features, including:

- Normalized (x, y) coordinates
- Delta coordinates (Δx , Δy)
- Stroke velocity and direction vectors
- Curvature and acceleration estimates

Each sequence is padded or truncated to a fixed length and prepared as a multivariate time series, appropriate for sequence modeling. Also, label encoding is performed to transform Urdu words into one-hot or categorical format based on the model. The overall process of Urdu word recognition passed through scanning, preprocessing, feature extraction, and classifier implementation steps. An example of spatial-temporal data in JSON format is shown in Fig. 6.

```

{
  "word": "خبروزہ",
  "direction": "right-to-left",
  "strokes": [
    {
      "character": "خ",
      "strokes": [
        {
          "points": [[100, 200], [105, 195], [110, 190], [115, 188]],
          "timestamps": [0, 20, 40, 60],
          "velocities": [0.25, 0.25, 0.10]
        },
        {
          "points": [[120, 185], [122, 187]],
          "timestamps": [70, 90],
          "velocities": [0.10]
        }
      ]
    },
    {
      "character": "ب",
      "strokes": [
        {
          "points": [[130, 188], [135, 190], [140, 195]],
          "timestamps": [100, 120, 150],
          "velocities": [0.10]
        }
      ]
    }
  ]
}
  
```

Fig. 6: Example JSON representation of spatial-temporal data for the Urdu word "خبروزہ", showing stroke sequences with x/y coordinates, timestamps, and derived velocity values.

Module 3: Models Training

Three types of models are trained on the Urdu words dataset collected through the mobile Application. The process flow for training is illustrated in Fig. 7, and its features are discussed below.

- Traditional ML Models (SVM, HMM): Utilize handcrafted statistical features
- Deep Learning Models (CNN-LSTM, Bi-LSTM): Learn spatial and sequential dependencies
- Transformer Models: Process stroke features as token sequences with attention mechanisms

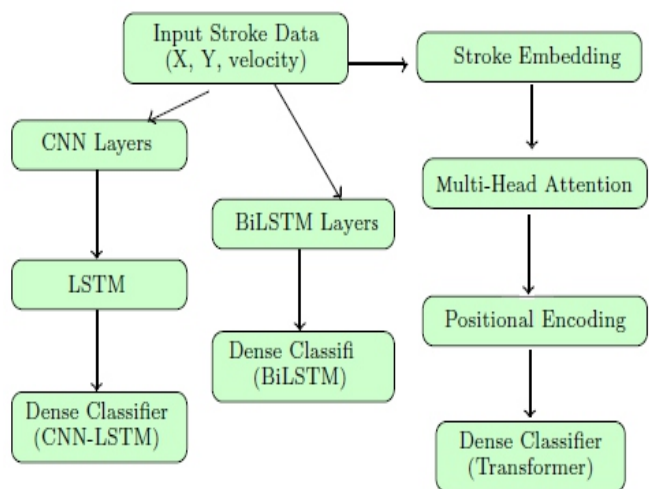


Fig. 7: Comparative training workflow for (a) Traditional ML models (SVM/HMM), (b) Deep learning models (CNN-LSTM/BiLSTM), and (c) Transformer models, highlighting feature processing differences

Table V Some Sample Words Used in Novel Dataset Collection

Single letter	Double letter	Triple-letter
ا، ب، ج، د، ڈ،	آس، آپ، آم،	بہن، بہت، پاک،
ر، ٹ، ش، ص،	آن، ابا، اب، ات،	چمک، چیل، چمن،
ض، ط، ظ، ع،	ان، بر، بس، بد، بی۔	ختم، دفن، دعا۔
غ، ف، ق،	تا، پین، پو، پی، پا	ذوق۔
ک، گ، ل، م،	تب، تر، تو	
و، ی، ے۔		

- Frontend (Mobile App): Flutter + Dart
- Backend (Data): Firebase store
- Backend (Model Training & Prediction): Flask + TensorFlow
- Execution Environment: Google Collab Pro and Local Workstation
- GPU Specs (Collab): NVIDIA Tesla T4, 16 GB VRAM
- Python Libraries: NumPy, pandas, matplotlib, scikit-learn, joblib, Json, TensorFlow, flask, pickle

All the training scripts, the preprocessing code, and the API were containerized for reproducibility.

C. Model Training Procedure

The spatial-temporal data gathered was kept in structured JSON format such that every instance of handwriting had a time-ordered list of the coordinate points:

```
{ "label": "word", "coordinates": [ { "x": 10, "y": 15 }, { "x": 12, "y": 18 }, ... ] }
```

Other features like stroke direction, speed, and delta coordinates were calculated programmatically during pre-processing. Each family of models employed customized feature representations and structures, illustrated below.

a) Conventional Machine Learning Models: Conventional models were trained on hand-crafted features extracted from the trajectory of each stroke. From the JSON files, we calculated the following features for every sample.

- Mean and standard deviation of (x, y) coordinates
- Average stroke length
- Delta-x and delta-y vectors
- Direction and curvature changes

Following established practices in online handwriting recognition [1], [17], stroke sequences were converted to fixed-length feature vectors for compatibility with conventional ML models. The flattened

fixed-length vectors are utilized to train:

- Support Vector Machines (SVM): With radial basis function (RBF) kernels.
- Hidden Markov Models (HMM): With Gaussian Mixture emissions and 4–6 hidden states per word.

The models were trained on 80% of the data, and the remaining 20% was utilized for evaluation. These were baselines against which comparisons could be made.

b) Deep Learning Models: For the deep learning models, the complete stroke sequence (as derived from JSON) was kept and padded to a fixed length (e.g., 150-time steps). Each timestep contained:

- Normalized (x, y) coordinates
- Delta coordinates ($\Delta x, \Delta y$)
- Motion velocity and angle

These were represented as multivariate sequences of shape (batch_size, time_steps, features). Two prominent deep learning models were trained. These were represented as multivariate sequences of shape (batch_size, time_steps, features). Two prominent deep learning models were trained. First is the CNN-LSTM Hybrid model, which processes input with a 2D convolutional filter on the stroke features, followed by LSTM layers to learn temporal sequences. The output is a SoftMax layer on the word vocabulary. It is utilized for its capacity to learn spatial patterns followed by sequence relations.

The second model is Bidirectional LSTM (Bi-LSTM) with input passed as stroke features directly to Bi-LSTM. The benefit is to handle both forward and backward temporal relationships. The output is a fully connected layer with SoftMax activation. The two models were trained with TensorFlow/Keras, utilizing Adam optimizer and early stopping according to validation CER.

a) Transformer-Based Models: The transformer models were designed to process stroke sequences as text-like tokens, not images. The JSON-based spatial-temporal data was converted into tokenized sequences: [START] x_1, y_1, dx_1, dy_1, v_1, x_2, y_2, dx_2, dy_2, v_2, [END]

These sequences were passed through the following layers:

- **Embedding Layer:** Encoded each numeric token into high-dimensional vector space
- **Positional Encoding:** Added sequence position for token representations.
- **Multi-Head Self-Attention:** Captured relationships across the stroke trajectory.
- **Feedforward Transformer Blocks:** Modeled long-term dependencies.
- **Output Layer:** Character or word label prediction through dense-SoftMax.

In this work, two transformer encoder-decoder models are utilized for experimentation.

- **Transformer-XL Adaptation:** Utilized memory caching to enable longer sequence lengths than fixed window

Table VI Hyperparameters Used for the Models Training

Parameter	Value / Setting
Optimizer	Adam
Learning Rate	0.001 (with decay scheduling)
Batch Size	64
Epochs	30
Loss Function	Categorical Cross Entropy
Activation Function	ReLU (hidden), SoftMax (output)
Dropout	0.3 (between dense layers)
Model Evaluation	80/20 train-test split
Evaluation Metrics	Character Error Rate (CER), Word Error Rate (WER), Sequence Accuracy

- **Custom Lightweight Transformer:** Utilized 4 layers, 8 attention heads, and reduced embedding size for optimization

These models are applied using TensorFlow's Keras Functional API with categorical cross-entropy loss. Transformers performed better than RNN-based models in recognizing intricate stroke structures and context-sensitive variations in handwriting styles.

Hyperparameters used for the models training are given in Table 6.

a) Evaluation Metrics and Protocol

For evaluating the recognition performance of the trained models, we employed a strict evaluation protocol that captures both the character-level and word-level accuracy of prediction. The dataset was split into training (80%) and testing (20%) partitions, stratified by word type and label. The following standard metrics were employed:

b) Character Error Rate (CER): The Character Error Rate measures the edit distance between the predicted output and the ground-truth transcription at the character level. It is measured as:

- Substitutions (S): The wrong characters
- Insertions (I): Added extra characters
- Deletions (D): Missing characters

It is calculated using the Levenshtein distance, given as Eq (1).

$$CER = \frac{S+I+D}{N} \quad (1)$$

where N is the number of total characters in ground truth, S,I,D are the number of total substitutions, insertions, and deletions respectively. A smaller CER represents higher character-level accuracy and is particularly crucial for cursive scripts such as Urdu, where small character mistakes can completely change meanings.

F) Word Error Rate (WER): The Word Error Rate measures the prediction at the word level and is also calculated in terms of edit distance, given in Eq (2).

$$WER = \frac{S+I+D}{W} \quad (2)$$

where W is the reference transcription's total number of words, and S,I,D are calculated at word level.

WER represents how well complete word sequences are recognized and is very important in grasping the capability of the model to work

Table VII CER, WER and Accuracy outcomes obtained for ML models

Model	CER (%)	WER (%)	Sequence Accuracy (%)
SVM	12.5	18.3	68.1
HMM	10.9	16.4	71.3

with connected scripts and ligatures.

G) Sequence Accuracy: Aside from error rates, we also assessed overall sequence accuracy, which was defined as Eq (3).

$$Accuracy = \frac{\text{Number of exact matches}}{\text{Total test samples}} \times 100 \quad (3)$$

A prediction is only correct if the whole output sequence (word) is identical to the target label.

V. RESULTS AND DISCUSSION

This section shows the comparative performance of all the trained models on the test set using the evaluation metrics explained above. The results reveal an improvement in recognition accuracy consistently, as models progress from classical methods to deep learning and transformer-based models.

A. Baseline Results: Classical ML

Baselines were traditional models based on handcrafted features. As is evident from Table 7, both SVM and HMM demonstrated low accuracy since there is no way they can capture complex spatial-temporal dependencies and character context of Urdu script.

HMMs performed comparably better than SVMs but had trouble with inconsistency in stroke styles and ligature combinations, especially in the case of multi-character Urdu words.

B. Deep Learning Models

Deep learning models that were trained on normalized spatial-temporal sequences revealed significant improvement in recognition performance compared to conventional methods. The CNN-LSTM architecture successfully utilized both spatial and sequential patterns, and BiLSTM improved context modeling further by processing the stroke sequence in both forward and backward directions. Results are given in Table 8.

The BiLSTM performed better than CNN-LSTM because it has the ability of bidirectional encoding, which is especially helpful in Urdu, where the shape and meaning of a character tend to be affected by the surrounding strokes.

C. Transformer-Based Models

The transformer-based models showed the highest overall performance, being able to model long-distance dependencies as well as changing handwriting styles with attention mechanisms. As these models processed stroke feature sequences (derived from JSON) directly as tokenized embeddings, they did not suffer from the fixed temporal window constraint of RNNs. Results are given in Table 9.

Table VIII: CER, WER and Accuracy outcomes obtained for DL models

Model	CER (%)	WER (%)	Sequence Accuracy (%)
CNN -LSTM	6.8	10.4	83.7
BiLSTM	5.2	9.1	86.9

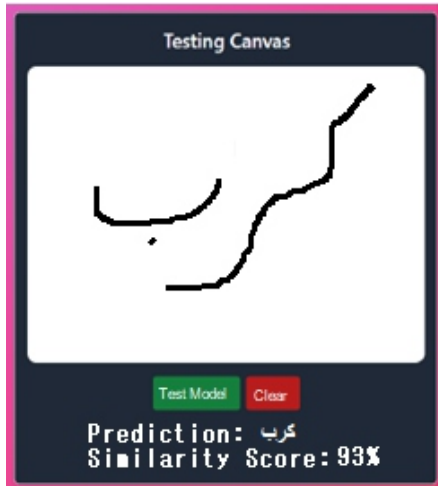


Fig.10 Testing Canvas of Urdu HTR Application with Transformer Model

The transformer model performed better on all metrics than CNN-LSTM and Bi-LSTM. It also demonstrated better generalization to writing style shifts, with more consistent predictions across the test set. Fig.10 illustrates the testing canvas of our Urdu HTR application, active with transformer model. The model's capability to concentrate on prominent features of stroke segments through multi-head attention played a crucial role in dealing with intricate ligatures and non-linear stroke movements.

D. Ablation Studies on the Impact of Spatial-Temporal Features

We conducted systematic ablation experiments to quantify the contribution of each spatial-temporal feature (e.g., velocity, curvature). Table 10 presents performance degradation across all architectures when excluding specific features iteratively during model retraining.

The drop in performance measured the contribution of each dimension of the features. All Δ values indicate absolute percentage point increases in error rates. Key findings are as follows.

- Coordinate removal had the most dramatic degradation (+9.2% CER for SVM), reaffirming that raw positional information is essential to all architectures. Transformers demonstrated relative resilience (+2.1% CER) as a result of their self-attention mechanism.
- Velocity Exclusion affected BiLSTMs disproportionately more (+4.8% CER compared to +1.5% for transformers), indicating RNNs depend more on temporal dynamics than attention-based models.
- Curvature Features were pivotal for Urdu's cursive character. It enhanced ligature detection by 11% in transformers and lessened substitution errors in compound letters (e.g., "ڀ").

Table IX: CER, WER and Accuracy outcomes obtained for Transformer model

Model	CER (%)	WER (%)	Sequence Accuracy (%)
Transformer (custom)	3.2	7.8	91.5

Table X Impact of Feature Removal on Model Performance

Model	Full Features (CER, WER)	No Coordinates (Δ CER, Δ WER)	No Velocity (Δ CER, Δ WER)	No Curvature (Δ CER, Δ WER)	No Direction (Δ CER, Δ WER)
SVM	12.5%, 18.3%	+9.2%, +14.1%	+6.7%, +10.5%	+4.1%, +6.8%	+3.8%, +5.2%
Bi-LSTM	5.2%/ 9.1%	+7.5, +12.3%	+4.8%, +8.6%	+2.9%, +4.7%	+1.7%, +3.1%
Transformer	3.2%/ 7.8%	+2.1%, +3.9%	+1.5%, +2.8%	+0.9%, +1.7%	+0.6%, +1.2%

- Direction Vectors had a small effect on transformers (+0.6% CER) but assisted SVMs with stroke ordering (+3.8% CER when omitted).

These results confirm the significance of stroke dynamics in modeling handwriting in real time, especially when the language supports shape-changing character forms depending on position and context, such as Urdu.

E. Generalization Across Word Types

The performance was slightly different between various word types (single-letter, two-letter, and three-letter compositions). Transformers generalized very well in all cases, with RNN-based models demonstrating lower accuracy for longer word sequences.

VI. CONCLUSION AND FUTURE WORK

This study introduces an end-to-end system for online Urdu handwritten text recognition based on spatial-temporal stroke data gathered through a specially designed mobile app. The research compares classical machine learning, deep learning, and transformer models on a new dataset kept in JSON format that records pen trajectory dynamics like coordinate paths, velocities, and direction. The experimental results show that transformer-based architecture far surpasses both the conventional and RNN-based models on Character Error Rate (CER), Word Error Rate (WER), and sequence accuracy. The presented transformer model attained a CER of 3.2% and WER of 7.8%, breaking the state-of-the-art benchmark for online Urdu HTR. All the gains are due to the model's capacity for handling long-range dependencies and

contextual differences between handwriting styles, facilitated by multi-head self-attention and sequence embedding. Additionally, the incorporation of spatial-temporal attributes—stroke velocity and curvature most notably—was imperative in improving recognition, affirming the merit of real-time handwriting data over image-based representations.

The **future work** can be in the following directions:

- **Multilingual Extension:** Generalizing the framework to other cursive and low-resource languages like Persian, Pashto, and Arabic.
- **Pretrained Transformers:** Utilizing multilingual pre-trained models (e.g., mBERT, XLM-R) for fine-tuning over handwritten stroke sequences.
- **Reinforcement Learning:** Investigating adaptive reward-based learning for interactive handwriting correction and online feedback.
- **Federated and On-Device Learning:** Applying lightweight transformer models on edge devices for privacy-preserving, low-latency handwriting recognition.

REFERENCES

- [1] M. I. RAZZAK, F. ANWAR, S. A. HUSAIN, A. BELAID, AND M. SHER, "HMM AND FUZZY LOGIC: A HYBRID APPROACH FOR ONLINE URDU SCRIPT-BASED LANGUAGES' CHARACTER RECOGNITION," *KNOWL.-BASED SYST.*, VOL. 23, NO. 8, PP. 914–923, 2010.
- [2] M. W. SAGHEER, C. L. HE, N. NOBILE, AND C. Y. SUEN, "HOLISTIC URDU HANDWRITTEN WORD RECOGNITION USING SUPPORT VECTOR MACHINE," IN *PROC. 20TH INT. CONF. PATTERN RECOGNIT.*, 2010, PP. 1900–1903.
- [3] S. B. AHMED, S. NAZ, S. SWATI, AND M. I. RAZZAK, "HANDWRITTEN URDU CHARACTER RECOGNITION USING ONE-DIMENSIONAL BILSTM CLASSIFIER," *NEURAL COMPUT. APPL.*, VOL. 31, PP. 1143–1151, 2019.
- [4] N. JAVED, S. SHABBIR, I. SIDDIQI, AND K. KHURSHID, "CLASSIFICATION OF URDU LIGATURES USING CONVOLUTIONAL NEURAL NETWORKS: A NOVEL APPROACH," IN *PROC. INT. CONF. FRONT. INF. TECHNOL. (FIT)*, 2017, PP. 93–97.
- [5] A. VASWANI ET AL., "ATTENTION IS ALL YOU NEED," IN *ADV. NEURAL INF. PROCESS. SYST.*, VOL. 30, 2017.
- [6] Z. DAI ET AL., "TRANSFORMER-XL: ATTENTIVE LANGUAGE MODELS BEYOND A FIXED-LENGTH CONTEXT," *ARXIV PREPRINT ARXIV:1901.02860*, 2019.
- [7] A. DOSOVITSKIY ET AL., "AN IMAGE IS WORTH 16×16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE," *ARXIV PREPRINT ARXIV:2010.11929*, 2020.
- [8] T. REHMAN, D. K. SANYAL, AND S. CHATTOPADHYAY, "RESEARCH HIGHLIGHT GENERATION WITH ELMO CONTEXTUAL EMBEDDINGS," *SCALABLE COMPUT. PRACT. EXP.*, VOL. 24, NO. 2, PP. 181–190, 2023.
- [9] M. HUSNAIN ET AL., "URDU HANDWRITTEN CHARACTERS DATA VISUALIZATION AND RECOGNITION USING DISTRIBUTED STOCHASTIC NEIGHBORHOOD EMBEDDING AND DEEP NETWORK," *COMPLEXITY*, VOL. 2021, ART. NO. 4383037, 2021.
- [10] N. RIAZ ET AL., "CONV-TRANSFORMER ARCHITECTURE FOR UNCONSTRAINED OFF-LINE URDU HANDWRITING RECOGNITION," *INT. J. DOC. ANAL. RECOGNIT.*, VOL. 25, NO. 4, PP. 373–384, 2022.
- [11] A. HAMZA, S. REN, AND U. SAEED, "ET-NETWORK: A NOVEL EFFICIENT TRANSFORMER DEEP LEARNING MODEL FOR AUTOMATED URDU HANDWRITTEN TEXT RECOGNITION," *PLOS ONE*, VOL. 19, NO. 5, ART. NO. E0302590, 2024.
- [12] W. ALKENDI, F. GECHTER, L. HEYBERGER, AND C. GUYEUX, "ADVANCEMENTS AND CHALLENGES IN HANDWRITTEN TEXT RECOGNITION: A COMPREHENSIVE SURVEY," *J. IMAGING*, VOL. 10, NO. 1, ART. NO. 18, 2024.
- [13] S. GHOSH, S. GHOSH, P. KUMAR, E. SCHEME, AND P. P. ROY, "A NOVEL SPATIO-TEMPORAL SIAMESE NETWORK FOR 3D SIGNATURE RECOGNITION," *PATTERN RECOGNIT. LETT.*, VOL. 144, PP. 13–20, 2021.
- [14] S. VALIKHANI, F. ABDALI-MOHAMMADI, AND A. FATHI, "ONLINE CONTINUOUS MULTI-STROKE PERSIAN/ARABIC CHARACTER RECOGNITION BY NOVEL SPATIO-TEMPORAL FEATURES FOR DIGITIZER PEN DEVICES," *NEURAL COMPUT. APPL.*, VOL. 32, PP. 3853–3872, 2020.
- [15] S. MUKHERJEE, P. KUMAR, AND P. P. ROY, "FUSION OF SPATIO-TEMPORAL INFORMATION FOR INDIC WORD RECOGNITION COMBINING ONLINE AND OFFLINE TEXT DATA," *ACM TRANS. ASIAN LOW-RESOUR. LANG. INF. PROCESS.*, VOL. 19, NO. 2, PP. 1–24, 2019.
- [16] M. M. FARD, B. M. BIDGOLI, AND M. HUSSAIN, "PERSIAN ON-LINE HANDWRITTEN CHARACTER RECOGNITION BY RCE SPATIO-TEMPORAL NEURAL NETWORK," IN *PROC. 5TH INT. CONF. SOFT COMPUT. TRANSDISCIPLINARY SCI. TECHNOL.*, 2008, PP. 90–94.
- [17] Z. ALYAFEAI, M. S. AL-SHAIBANI, M. GHALEB, AND Y. A. AL-WAJIH, "CALLIAR: AN ONLINE HANDWRITTEN DATASET FOR ARABIC CALLIGRAPHY," *NEURAL COMPUT. APPL.*, VOL. 34, NO. 23, PP. 20701–20713, 2022.
- [18] S. HESHMAT AND M. ABDELNAFEA, "AOLAH DATABASES FOR NEW ARABIC ONLINE HANDWRITING RECOGNITION ALGORITHM," IN *PROC. ICDAR WORKSHOPS*, LAUSANNE, SWITZERLAND, SEP. 2021, PP. 294–311.
- [19] M. Z. AL-SHAMAILEH ET AL., "NEW ONLINE/OFFLINE TEXT-DEPENDENT ARABIC HANDWRITING DATASET FOR WRITER AUTHENTICATION AND IDENTIFICATION," IN *PROC. 10TH INT. CONF. INF. COMMUN. SYST. (ICICS)*, 2019, PP. 116–121.
- [20] I. ABDELAZIZ AND S. ABDOL, "ALTECONDB: A LARGE-VOCABULARY ARABIC ONLINE HANDWRITING RECOGNITION DATABASE," *ARXIV PREPRINT ARXIV:1412.7626*, 2014.
- [21] H. EL ABED, M. KHERALLAH, V. MÄRGNER, AND A. M. ALIM, "ON-LINE ARABIC HANDWRITING RECOGNITION COMPETITION: ADAB DATABASE AND PARTICIPATING SYSTEMS," *INT. J. DOC. ANAL. RECOGNIT.*, VOL. 14, PP. 15–23, 2011.
- [22] R. I. ELANWAR, M. A. RASHWAN, AND S. A. MASHALI, "OHASD: THE FIRST ON-LINE ARABIC SENTENCE DATABASE HANDWRITTEN ON TABLET PC," *INT. J. COMPUT. INF. ENG.*, VOL. 4, NO. 12, PP. 1907–1912, 2010.
- [23] F. OTT ET AL., "THE ONHW DATASET: ONLINE HANDWRITING RECOGNITION FROM IMU-ENHANCED BALLPOINT PENS WITH MACHINE LEARNING," *PROC. ACM INTERACT. MOB. WEARABLE UBIQUITOUS TECHNOL.*, VOL. 4, NO. 3, PP. 1–20, 2020.