

Report

by Plag Report

Submission date: 26-May-2026 05:45PM (UTC+0530)

Submission ID: 2969800887

File name: BCM-Net_1.pdf (5.4M)

Word count: 7009

Character count: 37645

INTRODUCTION

Major Depressive Disorder (MDD) is one of the most common and disabling health problems of our time. According to the World Health Organization (WHO), depression is a leading cause of disability and contributes significantly to the global burden of disease. It affects hundreds of millions of people globally [1]. Although it's so common, diagnosing depression is still a very difficult task. Traditional diagnostic methods heavily depend on clinical interviews, self-report questionnaires, and subjective observations by psychiatrists, which are susceptible to human bias, demanding active patient cooperation, and resource intensive [2], [3]. These limitations make it difficult to access mental health care quickly, especially during severe depressive episodes. To address these issues, there has been a growing interest in digital phenotyping, which uses objective, computer-based methods to identify behavioral and linguistic features of depression [3, 4]. In this respect, Artificial Intelligence (AI) and especially Natural Language Processing (NLP) has become a key frontier in this domain. Previous studies have demonstrated that depressed people have unique linguistic characteristics, such as more usage of first-person singular pronouns, absolutist words, and negative emotion words [3], [4]. The ability to capture these complex, often long-range patterns in patient narratives holds promise for scalable and objective depression screening tools.

Automated depression detection has rapidly developed over the last decade. These properties make it particularly suitable for long-context mental health analysis, where subtle depression cues may be distributed across thousands of tokens.

This work presents BCM-Net (Bottleneck Cross-attention Mamba Network). BCM-Net encodes the temporal dynamics of each modality independently with Mamba State Space Model blocks at linear complexity instead of full self-attention which scales quadratically with the length of the sequence. This makes it suitable for processing long clinical recordings, where signals relevant to depression are often sparse and temporally distributed.

The difference of BCM-Net from previous Mamba-based methods lies in the cross-modal fusion strategy. After the temporal encoding for each modality, the model combines the audio and visual streams via a bi-directional bottleneck cross-attention mechanism, where each modality attends to the other in a compressed intermediate space. This reduces the cross-modal attention cost without sacrificing the ability of the model to capture meaningful inter-modal correlations. This is further enhanced by a subsequent gated fusion layer that learns the importance of each modality at each point in time, considering that the relative importance of audio and visual signals varies across individuals and recording conditions. Next, an attention pooling layer reduces the temporal dimension into a single representation attending to the frames most diagnostic for the classification decision.

The main goal of this work is to propose, implement, and evaluate BCM-Net for automatic binary depression classification on two real-world benchmark datasets – D-Vlog and LMVD. This work is a step towards depression detection systems that are both architecturally principled and practically deployable without the computational overhead associated with state-of-the-art multimodal models.

CHAPTER 2

LITERATURE REVIEW

Automatic depression detection has been a focus of much scholarly work in the last 10 years as mental health disorders are becoming more prevalent and the standard clinical screening methods have their limits. The early computational studies were heavily influenced by the benchmark datasets and assessment systems provided by the Audio/Visual Emotion Challenges (AVEC). The AVEC 2013[1] and AVEC 2014 challenges [2], [2] have formalized depression diagnosis as a multimodal learning problem, using organized clinical interviews. This allowed for detailed analysis of the time, sound and light elements. These problems allow for the application of supervised machine learning techniques that use manually generated descriptions of the face and the audio.

Meanwhile, researchers began to view social media as an alternative data source that can scale. A major step towards passive mental health monitoring was made by De Choudhury et al. [3], who demonstrated that language signals and posting habits on Twitter can predict clinical depression months before diagnosis.

Deep learning has made it so much easier to find sadness in books. “Recently graph-based text models have become a powerful way to do things. CDME-GAT [6] constructs graphs from social media posts by leveraging multi-embedding representations from large pretrained language models. Next, graph attention networks are employed to interpret the relationships among posts, leading to large efficiency improvements.

Meanwhile, the multimodal depression identification performed better than the unimodal methods. Previous deep multimodal methods fused audio and video streams by CNNs and RNNs. On the other hand, causal convolutional networks made time easier to describe [7]. The application of facial and vocal clues in practical psychiatric environments has been proved effective in relapse prediction systems [8] and other clinical applications. However, these models often face challenges in scalability and have limited ability to fuse information.

Graph-based multimodal fusion approaches are increasingly used to solve these problems. However, Fu et al. [9] combined facial action units with graph-guided fusion to address the problem of identification bias. Yang and Zheng proposed the MSGAF system [10], utilizing bidirectional fusion graphs to depict intra-modal temporal dynamics and inter-modal dependencies. Related techniques analysed graph and hypergraph structures more in depth to show the functioning of complex multimodal interconnections. For instance, EEG-speech fusion based on GCN[11] and depression detection based on hypergraph[12].

The next big step forward was transformer based designs. MMFormer [13] proposed a multimodal fusion transformer that enables the mutual interaction of different modalities and mimics the spatial and temporal properties of audio-visual inputs. Such alignment at the utterance or sentence level between the modalities greatly improves the performance on benchmark datasets such as DAIC-WOZ. This has been shown by several transformer-based frameworks, such as MDD-Net [14] and Sentence-level Multimodal Fusion Learning (SMFL) [15].

But even when transformer models do well, their quadratic complexity means that they can only handle large sequences. This limitation has led to the development of state-space models, i.e., Mamba-based systems. [16] was the first to apply state-space models to detect multimodal depression. It achieved a successful long-range dependency capture through progressive state-space fusion with convolutional feature extraction. MDDMamba [17] also used a memory-efficient cross-modal attention based on Mamba, which greatly reduces the computation cost. STE-Mamba [18] extended this idea by combining the emotional trend analysis with the spatio-temporal modeling. This turned out great. The results were great. Recent surveys and benchmarking initiatives [19] illustrate the importance of huge language models and hybrid architectures. This indicates the significance of conducting extensive evaluations in social and therapeutic settings. The literature shows a clear tendency towards more complex models, from hand-crafted features to advanced multi-modal, graph-based, transformer-based and state-space models. All these models are scalable, interpretable, and clinically significant.

Chapter 3

METHODOLOGY

3.1 Datasets and Data Preparation

Two publicly available benchmarks are used to train and evaluate BCM-Net: the D-Vlog dataset and the Large-scale Multimodal Video Dataset (LMVD). Both provide synchronised audio-visual recordings annotated for binary depression classification (depressed / non-depressed).

3.1.1 D-Vlog dataset

D-Vlog [22] consists of 961 YouTube vlogs collected between January 2020 and January 2021, comprising approximately 160 hours of naturalistic in-the-wild video. BCM-Net follows the evaluation protocol adopted by DepMamba [16] and MDDMamba [17] on this dataset to ensure comparability.

3.1.2 D-Vlog dataset

The Large-scale Multimodal Video Dataset (LMVD) [23] contains 1,823 videos sourced from multiple social media platforms, of which 908 are depression- positive and 915 are depression negative — a near-balanced class distribution. The multi-platform origin introduces greater domain diversity than D-Vlog in terms of recording style and speaker demographics. LMVD is used to assess the generalisability of BCM-Net beyond a single source domain. The evaluation protocol follows that used by DepMSTAT [24] on this dataset.

3.1.3 Feature Extraction

3.1.3.1 Acoustic features: eGeMAPS

Acoustic features are extracted using the extended Geneva Minimalistic Acoustic Parameter Set (eGeMAPS) [25] via the openSMILE toolkit. eGeMAPS is a curated 25-dimensional acoustic descriptor designed for paralinguistic and affective computing research. It encompasses four families of parameters: (i) frequency-related features (fundamental frequency F0, jitter, shimmer), (ii) energy and amplitude features (loudness, harmonic- to-noise ratio), (iii) spectral features (spectral flux, alpha ratio, Hammarberg index), and (iv) cepstral features (MFCC coefficients 1–4).

Features are extracted frame-by-frame, yielding an acoustic tensor of shape (B, T, 25), where B is the batch size and T is the number of frames. eGeMAPS has been adopted as the acoustic representation in DepMamba [16], STE-Mamba [18], and EMO-Mamba [26], providing a standardised baseline for comparison.

3.1.4 Data Preprocessing and Augmentation

All features are normalised using Z-score standardisation computed over the training set using Welford's online algorithm to avoid loading the full dataset into memory:

$$z = \frac{x - \mu}{\sigma + \epsilon} \quad \dots (Eq. 3.1)$$

where μ and σ are the per-feature global mean and standard deviation computed from the training split only, and $\epsilon = 1e-8$ prevents division by zero. Validation and test features are normalised using training statistics to prevent data leakage.

Sequences are padded or truncated to a fixed length of $T_{\max} = 1,000$ frames. A binary padding mask M in $\{0,1\}^{B \times T}$ is constructed for each batch, where $M_t = 1$ denotes a real frame and $M_t = 0$ denotes a padded position.

Two stochastic augmentation strategies are applied to the training split:

(i) Gaussian noise injection:

$$x_{aug} = x + \text{epsilon}, \text{epsilon} \sim N(0, 0.05^2) \dots (Eq. 3.2)$$

Small-amplitude noise is added independently to each feature dimension, simulating recording variability.

(ii) Time masking: a contiguous span of 10% of T frames is randomly selected and set to zero, following the SpecAugment paradigm [27]. This encourages the model to rely on distributed temporal cues rather than a fixed diagnostic window.

Neither augmentation is applied at validation or test time.

3.2 Background

BCM-Net is built on three fundamental building blocks: the attention mechanism, cross-attention, and Mamba selective state space models. This section provides a self-contained theoretical treatment of each, which motivates the architectural decisions described in Section 3.3.

3.2.1 Attention Mechanism

The attention mechanism [30] is a function that allows a neural network to selectively focus on relevant parts of an input sequence when producing each element of an output sequence. Rather than summarising the entire input into a fixed-size context vector, attention computes a weighted combination of all input positions, where the weights reflect the relevance of each input position to the current output position.

3.2.1.1 Scaled Dot-Product Attention

The fundamental operation of modern attention is scaled dot-product attention. Given a query vector q in $\mathbb{R}^{d_k}$, a set of key vectors K in $\mathbb{R}^{N \times d_k}$, and a set of value vectors V in $\mathbb{R}^{N \times d_v}$, the attention output for a single query is:

$$e_i = q^T * k_i / \sqrt{d_k}, \text{ for } i = 1, \dots, N \dots (Eq. 3.3)$$

$$\alpha_i = \exp(e_i) / \sum_{j=1}^N \exp(e_j) \dots (Eq.3.4)$$

$$Attn(q, K, V) = \sum_{i=1}^N \alpha_i * v_i \dots (Eq.3.5)$$

where e_i is the raw attention score (also called the logit or alignment score between the query and the i -th key, α_i is the normalised attention weight obtained by applying softmax over all N scores, and the output is the weighted sum of value vectors. In matrix form, given Q in $\mathbb{R}^{L \times d_k}$ (L queries), K in $\mathbb{R}^{N \times d_k}$, and V in $\mathbb{R}^{N \times d_v}$:

$$Attn(Q, K, V) = \text{softmax}(Q * K^T / \sqrt{d_k}) * V \dots (Eq.3.6)$$

The scaling factor $1/\sqrt{d_k}$ prevents the dot products from growing large in magnitude as d_k increases, which would push the softmax into regions of very small gradient. This scaling was introduced by Vaswani et al. [30] as a stabilising heuristic.

The attention weights α_i are interpretable: a weight close to 1.0 on position i means the output is almost entirely determined by the value at position i , while a uniform distribution indicates equal reliance on all positions.

3.2.1.2 Multi-Head Attention

A single attention head can only capture one type of positional relationship at a time. Multi-head attention (MHA) [30] projects queries, keys, and values into h parallel subspaces using separate learned projections, computes attention independently in each subspace, and concatenates the results:

$$head_i = Attn(Q * W_Q^i, K * W_K^i, V * W_V^i) \dots (Eq.3.7a)$$

$$MHA(Q, K, V) = \text{Concat}(head_1, \dots, head_h) * W_O \dots (Eq.3.7b)$$

where W_Q^i in $\mathbb{R}^{d_{\text{model}} \times d_k}$, W_K^i in $\mathbb{R}^{d_{\text{model}} \times d_k}$, W_V^i in $\mathbb{R}^{d_{\text{model}} \times d_v}$ are per-head projection matrices, and W_O in $\mathbb{R}^{h * d_v \times d_{\text{model}}}$ is the output projection. Typically $d_k = d_v = d_{\text{model}} / h$, so the total computation cost is the same as single-head attention.

3.2.1.3 Self-Attention and Temporal Modelling

When the query, key, and value matrices all come from the same sequence — i.e., $Q = K = V = X$ — the operation is called self-attention. This limitation is one of the primary motivations for adopting Mamba (Section 3.2.3) in BCM-Net.

In the context of depression detection, self-attention has been used in models such as MMFormer [10] and DepMSTAT [24], where it enables each frame representation to be updated based on all other frames in the recording. However, these models incur high memory and compute costs for long sequences.

3.2.2 Cross-Attention

Cross-attention is a generalisation of the attention mechanism in which the query sequence and the key-value sequence come from two different sources. It enables one modality or representation to "look into" another and retrieve relevant information from it, making it the natural mechanism for multimodal fusion.

3.2.2.1 Distinction from Self-Attention

In self-attention, Q, K, V are all derived from the same input X. In cross-attention, the queries come from one source sequence X and the keys and values come from a different source sequence Y:

$$Q = X * W_Q, K = Y * W_K, V = Y * W_V \dots (Eq.3.8)$$

$$CA(X, Y) = softmax(Q * K^T / sqrt(d_k)) * V \dots (Eq.3.9)$$

The output has the same length as X (one output per query), but each output vector is informed by the content of Y through the attention weights. Intuitively, each position in X asks a "question" (query), and the attention mechanism finds the most relevant "answers" (keys) in Y, then retrieves the associated "information" (values) from Y.

Cross-attention is the core mechanism in the encoder-decoder architecture of the original Transformer [30], where the decoder queries the encoder output to condition text generation on the source sequence.

3.2.2.2 Multi-Head Cross-Attention

Multi-head cross-attention follows the same formulation as multi-head self-attention (Eq. 3.7), except Q and K, V come from different sources:

$$head_i^{X \rightarrow Y} = Attn(X * W_Q^i, Y * W_K^i, Y * W_V^i) \dots (Eq.3.10a)$$

$$MCA(X, Y) = Concat(head_1^{X \rightarrow Y}, \dots, head_h^{X \rightarrow Y}) * W_O \dots (Eq.3.10b)$$

Each head learns to attend to different types of relationships simultaneously — for example, one head may focus on local temporal patterns while another captures long-range dependencies. This parallel multi-perspective attention is a key reason for the expressiveness of Transformer-based models.

3.2.1.3 Self-Attention and Temporal Modelling

When the query, key, and value matrices all come from the same sequence — i.e., Q = K = V = X — the operation is called self-attention. Every position in the sequence attends to every other position, producing a context-aware representation that captures global dependencies regardless of the distance between positions.

Self-attention computes an $N \times N$ attention matrix (where N is the sequence length), giving it $O(N^2)$ time and memory complexity. For short sequences this is acceptable, but for long clinical vlog recordings where N can reach hundreds to thousands of frames, the quadratic cost becomes a significant bottleneck. This limitation is one of the primary motivations for adopting Mamba (Section 3.2.3) in BCM-Net.

In the context of depression detection, self-attention has been used in models such as MMFormer [13] and DepMSTAT [24], where it enables each frame representation to be updated based on all other frames in the recording. However, these models incur high memory and compute costs for long sequences.

3.2.2 Cross-Attention

Cross-attention is a generalisation of the attention mechanism in which the query sequence and the key-value sequence come from two different sources. It enables one modality or representation to "look into" another and retrieve relevant information from it, making it the natural mechanism for multimodal fusion.

3.2.2.1 Distinction from Self-Attention

In self-attention, Q, K, V are all derived from the same input X . In cross-attention, the queries come from one source sequence X and the keys and values come from a different source sequence Y :

$$Q = X * W_Q, K = Y * W_K, V = Y * W_V \dots (Eq.3.8)$$

$$CA(X, Y) = softmax(Q * K^T / sqrt(d_k)) * V \dots (Eq.3.9)$$

The output has the same length as X (one output per query), but each output vector is informed by the content of Y through the attention weights. Intuitively, each position in X asks a "question" (query), and the attention mechanism finds the most relevant "answers" (keys) in Y , then retrieves the associated "information" (values) from Y .

Cross-attention is the core mechanism in the encoder-decoder architecture of the original Transformer [30], where the decoder queries the encoder output to condition text generation on the source sequence.

3.2.2.2 Multi-Head Cross-Attention

Multi-head cross-attention follows the same formulation as multi-head self-attention (Eq. 3.7), except Q and K, V come from different sources:

$$CA(X, Y) = softmax(Q * K^T / sqrt(d_k)) * V \dots (Eq.3.9)$$

$$MCA(X, Y) = Concat(head_1^{X \rightarrow Y}, \dots, head_h^{X \rightarrow Y}) * W_O \dots (Eq.3.10b)$$

The multi-head formulation allows the model to capture multiple types of cross-modal relationships simultaneously: for example, in the context of depression detection, one head might correlate vocal

energy (audio) with mouth movement (visual), while another correlates speech rate with brow position.

3.2.2.3 Role in Multimodal Fusion

In a multimodal setting with modalities X (e.g., audio) and Y (e.g., visual), cross-attention can be applied bidirectionally:

- Audio attends to visual: $CA(h_a, h_v)$ — each audio frame gathers context from all visual frames.
- Visual attends to audio: $CA(h_v, h_a)$ — each visual frame gathers context from all audio frames.

This bidirectional cross-attention ensures that information flows symmetrically between modalities, producing enriched representations for both streams that encode inter-modal context. This is the strategy adopted in BCM-Net's Stage 3 (Section 3.3.4), and is contrasted with late fusion (where modalities are combined only at the output level) and concatenation- based early fusion (which lacks the selective alignment that attention provides).

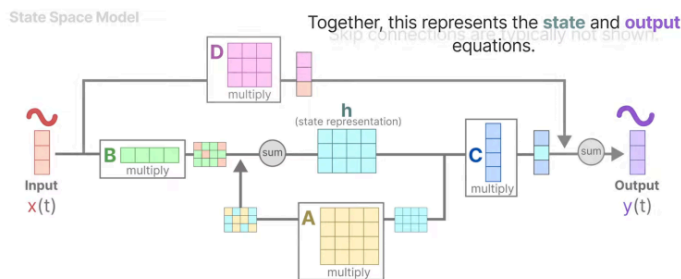


Fig. 3.1: State-Space Model

16

3.2.3 State Space Models

State space models (SSMs) are a class of sequence models rooted in control theory and signal processing. They represent a sequence as a dynamical system: a hidden state evolves over time according to a differential equation, and observations are linear readouts of that state. SSMs offer a compelling alternative to self-attention because they process sequences at $O(T)$ time and memory complexity rather than $O(T^2)$.

3.2.3.1 Classical Linear SSMs

The continuous-time linear SSM is defined by the following differential and output equations:

$$\begin{aligned} \frac{dx(t)}{dt} &= A * x(t) + B * u(t) \dots (Eq.3.11a) \\ y(t) &= C * x(t) + D * u(t) \dots (Eq.3.11b) \end{aligned}$$

where:

- $u(t)$ in $\mathbb{R}$ is the scalar input signal at time t
- $x(t)$ in $\mathbb{R}^N$ is the hidden state vector of dimension N (state size)
- $y(t)$ in $\mathbb{R}$ is the scalar output
- A in $\mathbb{R}^{N \times N}$ is the state transition (or system) matrix
- B in $\mathbb{R}^{N \times 1}$ is the input projection matrix
- C in $\mathbb{R}^{1 \times N}$ is the output projection matrix
- D in $\mathbb{R}$ is a scalar direct feedthrough (skip connection) term

The matrix A governs how information stored in the hidden state decays or propagates over time. For stable systems, the eigenvalues of A must have negative real parts. The matrices B and C determine how the input writes into the state and how the state is read out as the output, respectively.

For discrete-time sequence modelling (as required for processing frame-level features), the continuous SSM is discretised using the zero-order hold (ZOH) rule with a step size δ (the sampling interval):

$$A_{\delta} = \exp(\delta * A) \dots (Eq.3.12a)$$

$$B_{\delta} = (\delta * A)^{-1} * (\exp(\delta * A) - I) * \delta * B \dots (Eq.3.12b)$$

This yields the recurrence:

$$h_t = A_{\delta} * h_{t-1} + B_{\delta} * x_t \dots (Eq.3.13a)$$

$$y_t = C * h_t + D * x_t \dots (Eq.3.13b)$$

where h_t in $\mathbb{R}^N$ is the discrete hidden state at timestep t , x_t in $\mathbb{R}$ is the input token, and y_t in $\mathbb{R}$ is the output token. This recurrence can be unrolled as a convolutional operation over the full sequence, enabling efficient parallel training:

$$y = K_{SSM} * u \dots (Eq.3.14)$$

where $K_{SSM} = (C * B, C * A_{\delta} * B, C * A_{\delta}^2 * B, \dots)$ is the SSM convolutional kernel (also called the impulse response), $*$ denotes convolution, and u is the full input sequence. The ability to compute SSMs both as a recurrence (linear time, constant memory, efficient inference) and as a convolution (parallelisable, efficient training) is the central computational advantage of SSMs over RNNs.

3.2.3.2 Structured State Spaces (S4)

A key challenge for classical SSMs is initialising and learning the state transition matrix A in a way that captures long-range dependencies. The HiPPO matrix is designed so that the hidden state $x(t)$ optimally memorises the history of $u(t)$ up to time t in a polynomial basis:

$$A_{nk} = -\frac{2n+1}{2} * \frac{2k+1}{2} * \frac{1}{2} \text{if } n > k = n + 1 \text{if } n = k = 0 \text{if } n < k \dots (Eq.3.15)$$

S4 demonstrated that SSMs with this structure could match or exceed Transformer performance on long-range sequence benchmarks while maintaining $O(T)$ complexity. However, S4 uses fixed (input-independent) matrices B , C , and δ , limiting its ability to selectively filter the input based on content — a limitation resolved by Mamba.

3.2.4 Mamba: Selective State Space Models

Mamba [20] extends the S4 framework with a content-aware (selective) parameterisation that allows the model to dynamically decide which information to retain in the hidden state and which to discard. This selectivity is the key innovation that makes Mamba outperform classical SSMs on tasks requiring content-based reasoning over long sequences, such as language modelling and — as demonstrated in DepMamba [16] and EMO- Mamba [26] — depression detection from long clinical recordings.

3.2.4.1 Selectivity Mechanism

In S4, the matrices B , C , and the step size δ are fixed parameters learned during training and shared across all timesteps. Mamba makes these parameters input-dependent: they are computed as learned functions of the current input token x_t :

$$[B_t, C_t, \delta_t] = \text{Linear}(x_t) \dots (Eq.3.16a)$$

$$\delta_t = \text{softplus}(\delta_t) \dots (Eq.3.16b)$$

where Linear denotes a learned linear projection and softplus ensures $\delta_t > 0$. The discretised matrices then become timestep-specific:

$$A_t \bar{a}_t = \exp(\delta_t * A) \dots (Eq.3.17a)$$

$$B_t \bar{a}_t = (\delta_t * A)^{-1} * (\exp(\delta_t * A) - I) * \delta_t * B_t \dots (Eq.3.17b)$$

and the recurrence becomes:

$$h_t = A_t \bar{a}_t * h_{t-1} + B_t \bar{a}_t * x_t \dots (Eq.3.18a)$$

$$y_t = C_t * h_t + D * x_t \dots (Eq.3.18b)$$

This selective recurrence has a compelling interpretation: the step size δ_t controls how much time passes between consecutive inputs. A large δ_t makes $A_t \bar{a}_t$ approach zero (the previous state is discarded and the new input dominates), while a small δ_t preserves the previous state strongly (the model "ignores" the current input). By making δ_t input-dependent, Mamba can

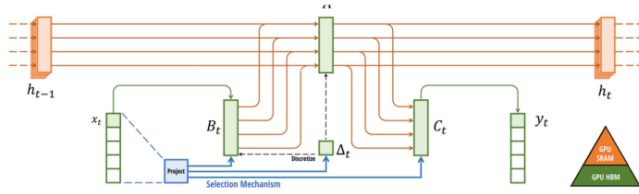


Fig. 3.2: The selective SSM and RAM allocation on a GPU. Taken from the original paper, "Mamba: Linear Time-Sequence Modeling with Selective State Spaces"

learn to focus on informative timesteps (e.g., speech segments containing depressive content) and compress uninformative ones (e.g., silence or background noise).

Similarly, input-dependent B_t controls how the current input x_t is written into the state, and input dependent C_t controls how the state is read out. Together, these three selective gates give Mamba the ability to perform content-based filtering — a capability absent in S4 but essential for processing heterogeneous clinical recordings where relevant cues are sparse and context-dependent.

3.2.4.2 Hardware-Aware Parallel Scan

The selective nature of Mamba (input-dependent parameters) prevents it from being computed as a global convolution like S4. Instead, Mamba computes the recurrence using a parallel prefix scan (also known as an associative scan), which computes all elements $h_1, h_2, \dots, h_T$ in $O(\log T)$ parallel steps [20]. The parallel scan is implemented in a hardware-aware manner using CUDA kernel fusion and online normalisation to avoid materialising the full $N \times T$ state matrix in high bandwidth memory (HBM). This keeps the memory footprint at $O(T \times N)$ rather than the $O(T^2)$ of self-attention, while maintaining the expressive content-based selection of the selective scan.

Additionally, Mamba incorporates a local convolutional layer (kernel size $d_{\text{conv}} = 4$) applied to the input before the selective scan. This convolutional pre-processing captures local context within a small neighbourhood, complementing the global long-range modelling of the recurrent SSM.

3.2.4.3 Comparison with Transformers

Table 3.0 summarises the key differences between Mamba and the standard Transformer self-attention mechanism.

Table 3.0: Mamba vs. Transformer Self-Attention

Property	Transformer Attn	Mamba SSM
Time Complexity	$O(T^2 * d)$	$O(T * d * N)$
Memory Complexity	$O(T^2)$	$O(T * N)$
Parallelism (train)	Full parallel	Parallel scan
Inference mode	KV-cache $O(T)$	Recurrent $O(1)$
Content-awareness	Full (QK attn)	Selective B,C,delta
Long-range modelling	Yes ($O(T^2)$ cost)	Yes ($O(T)$ cost)
Fixed context window	No	No

For long vlog sequences ($T \sim$ hundreds of frames), the quadratic memory cost of self-attention can exhaust GPU memory, while Mamba scales linearly. Empirically, Mamba has been shown to achieve 15x faster inference and 92.3% lower GPU memory usage than comparable Transformer architectures at long sequence lengths [16], making it highly suitable for processing the extended audio-visual recordings in D-Vlog and LMVD.

3.3.1 Architectural Overview

BCM-Net is an end-to-end binary classifier that accepts paired acoustic and visual feature streams and predicts a depression label for each recording. The architecture integrates the three theoretical components described in Section 3.2: (i) Mamba SSM blocks for efficient within- modality temporal encoding, (ii) bidirectional bottleneck cross-attention for inter-modal fusion, and (iii) attention pooling for temporally selective aggregation. A gated fusion layer provides adaptive modality weighting before classification.

The design draws on DepMamba [16] for Mamba-based multimodal depression detection, MMFormer [10] for cross-modal fusion, and MDDMamba [17] for the bottleneck cross-attention design.

3.3.2 Stage 1: Input Normalization and Linear Projection

Each modality is independently normalised and projected into a shared hidden space of dimension $d = 256$. Layer normalisation (LayerNorm) is applied to stabilise the input distribution, followed by a modality - specific linear transformation:

$$h_a^{(0)} = W_a * LayerNorm(x_a) + b_a \dots (Eq.3.19)$$

$$h_v^{(0)} = W_v * LayerNorm(x_v) + b_v \dots (Eq.3.20)$$

where W_a in $\mathbb{R}^{256 \times 25}$ and W_v in $\mathbb{R}^{256 \times 136}$ are learnable weight matrices, and b_a , b_v in $\mathbb{R}^{256}$ are bias vectors. After projection, both modalities reside in the same $\mathbb{R}^{256}$ feature space, enabling subsequent cross-modal operations. The hidden dimension $d = 256$ balances

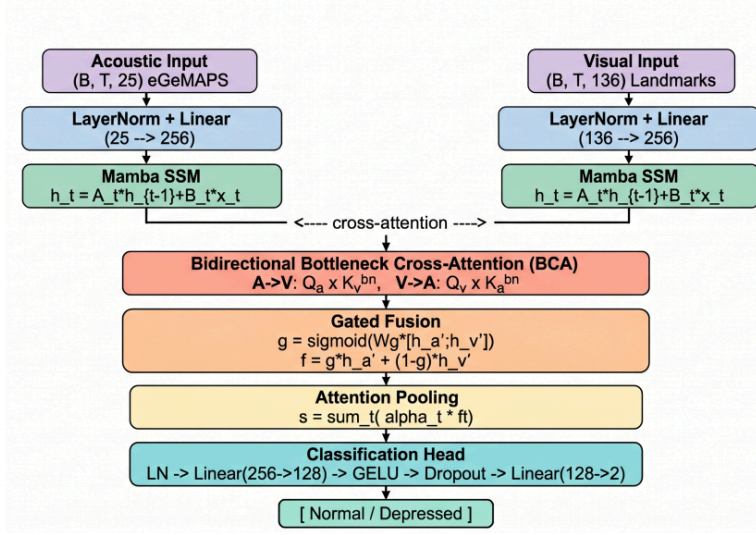


Figure 3.3: BCM-Net Architecture Overview

representational capacity against parameter count, consistent with comparable Mamba-based depression models [2, 3].

3.3.3 Stage 2: Mamba-Based Temporal Encoding

Independent Mamba SSM blocks (described in Section 3.2.4) are applied to $h_a^{(0)}$ and $h_v^{(0)}$ to encode within-modality temporal dynamics at $O(T)$ complexity. Each modality is encoded separately at this stage, preserving modality-specific temporal structure before cross-modal fusion.

3.3.3.1 SSM Configuration

The Mamba block operates with $d_{\text{state}} = 16$, $d_{\text{conv}} = 4$, and an expansion factor of 2, following the default Mamba configuration [20]. The expansion factor projects the $d = 256$ input into an inner dimension of 512 before the selective scan, providing additional representational capacity within each block. A dropout rate of 0.3 is applied within each Mamba block for regularisation.

The selective recurrence for each modality encoder is (from Eq. 3.18):

$$h_t = A_b a_{r_t} * h_{t-1} + B_b a_{r_t} * x_t \dots (Eq. 3.21a)$$

$$y_t = C_t * h_t + D * x_t \dots (Eq. 3.21b)$$

where A_t , B_t , and C_t are input-dependent and computed via learned linear projections of the current input token x_t (Eq. 3.16–3.17).

3.3.4 Stage 3: Bidirectional Bottleneck Cross-Attention

After modality-specific temporal encoding, BCM-Net fuses the two streams through a bidirectional bottleneck cross-attention (BCA) mechanism (building on Section 3.2.2). The bottleneck design compresses key and value projections to a reduced dimension $d_{bn} < d$, lowering the cost of cross-modal attention without sacrificing fusion quality [17].

3.3.4.1 Bottleneck Projection

Rather than computing cross-attention in the full $d = 256$ dimensional space, the source modality's keys and values are projected into a compact bottleneck space of dimension d_{bn} :

$$K_{bn} = W_K^{bn} * h_{src} \dots (Eq.3.22a)$$

$$V_{bn} = W_V^{bn} * h_{src} \dots (Eq.3.22b)$$

where W_K^{bn} , W_V^{bn} in $\mathbb{R}^{d_{bn} \times 256}$ are learnable bottleneck projection matrices, and h_{src} is the source modality's temporal representation. This reduces attention cost from $O(T^2 * d)$ to $O(T^2 * d_{bn})$, following the design rationale of MDDMamba [17].

3.3.4.2 Cross – Attention Computation

Cross-attention is computed bidirectionally and in parallel: the acoustic stream attends to the visual stream ($A \rightarrow V$), and the visual stream attends to the acoustic stream ($V \rightarrow A$). Using the multi-head cross-attention formulation of Section 3.2.2.2:

For the $A \rightarrow V$ direction:

$$Q_a = W_Q^a * h_a$$

$$K_v^{bn} = W_K^v * h_v, V_v^{bn} = W_V^v * h_v$$

$$Attn_{A \rightarrow V} = softmax(Q_a * (K_v^{bn})^T / sqrt(d_k)) * V_v^{bn} \dots (Eq.3.23)$$

Symmetrically, for the $V \rightarrow A$ direction:

$$Q_v = W_Q^v * h_v$$

$$K_a^{bn} = W_K^a * h_a, V_a^{bn} = W_V^a * h_a$$

$$Attn_{V \rightarrow A} = softmax(Q_v * (K_a^{bn})^T / sqrt(d_k)) * V_a^{bn} \dots (Eq.3.24)$$

where $d_k = d_{bn} / n_heads$ is the per-head key dimension and $n_heads = 4$. The scaling factor $1/sqrt(d_k)$ prevents dot-product magnitudes from saturating the softmax gradient.

The attention outputs are added residually to the original representations:

$$h'_a = h_a + \text{Attn}_{A \rightarrow V} \dots (Eq.3.25a)$$

$$h'_v = h_v + \text{Attn}_{V \rightarrow A} \dots (Eq.3.25b)$$

This residual connection ensures that modality-specific information is not discarded during cross modal fusion, consistent with the strategy in MMFormer [13] and CBA [12].

3.3.4.3 Masking for Variable-Length Sequences

Since vlog recordings have variable duration, sequences are padded to $T_{\max} = 1,000$ frames. The padding mask M is applied before the softmax by setting padded positions to a large negative value:

$$score_t = score_t + (1 - M_t) * (-1e9) \dots (Eq.3.26)$$

After softmax, padded positions receive attention weight ~ 0 , effectively excluding them from the computation without requiring variable-length batching.

3.3.5 Stage 4: Gated Fusion

Following cross-attention, the enriched acoustic representation h_a' and visual representation h_v' are combined via a learned gating mechanism.

A gate vector g in $(0, 1)^{256}$ is computed from the concatenation of both streams:

$$g = \text{sigmoid}(W_g * [h'_a; h'_v] + b_g) \dots (Eq.3.27)$$

where W_g in $\mathbb{R}^{256 \times 512}$ and $[\cdot]$ denotes feature-wise concatenation. The fused representation is computed as an element-wise weighted combination:

$$f_{fused} = g(x)h'_a + (1 - g)(x)h'_v \dots (Eq.3.28)$$

where (x) denotes element-wise multiplication.

3.3.6 Stage 5: Attention Pooling

The fused temporal representation f_{fused} in $\mathbb{R}^{B \times T \times 256}$ must be collapsed to a fixed-size utterance-level vector for classification. Attention pooling (Section 3.2.1) learns a scalar importance weight for each frame:

$$e_t = w^T * \tanh(W_p * f_{fused}_t + b_p) \dots (Eq. 3.29)$$

where W_p in $\mathbb{R}^{d_{att} \times 256}$ and w in $\mathbb{R}^{d_{att}}$ are learnable parameters. Padded frames are masked using Eq. 3.26. The weights are normalised via softmax over the T dimension:

$$\alpha_t = \exp(e_t) / \sum_{t'=1}^T \exp(e_{t'}) \dots (Eq.3.30a)$$

The utterance-level embedding is then:

$$s = \sum_{t=1}^T \alpha_t * f_{used_t} \dots (Eq.3.30b)$$

where s in $\mathbb{R}^{256}$ captures the most diagnostically salient temporal segments of the recording, improving both classification performance and model interpretability [12].

3.3.7 Stage 6: Classification Head

The utterance embedding s is passed through a two-layer MLP classifier:

$$s' = LayerNorm(s) \dots (Eq.3.31a)$$

$$s' = LayerNorm(s) \dots (Eq.3.31a)$$

$$h_1 = Dropout(h_1, p = 0.3) \dots (Eq.3.31c)$$

$$logits = W_2 * h_1 + b_2 \dots (Eq.3.31d)$$

$$y_{hat} = \argmax(\text{softmax}(logits)) \dots (Eq.3.31e)$$

where W_1 in $\mathbb{R}^{128 \times 256}$ and W_2 in $\mathbb{R}^{2 \times 128}$. The GELU activation provides smooth nonlinearity with improved gradient flow compared to ReLU. The output logits correspond to the two class scores [Normal, Depressed].

3.4 Training Protocol

3.4.1 Loss Function

BCM-Net is trained with cross-entropy loss combined with label smoothing. Standard one-hot labels are replaced with smoothed targets to reduce overconfidence and improve calibration on noisy depression labels:

$$y_c^{smooth} = (1 - \epsilon) * y_c + \epsilon / C \dots (Eq.3.32a)$$

where y_c is the one-hot label for class c in {Normal, Depressed}, $C = 2$, and $\epsilon = 0.1$. The loss is then:

$$L = - \sum_{c=1}^C y_c^{smooth} * \log(p_c) \dots (Eq.3.32b)$$

where $p_c = \text{softmax}(logits)_c$. When class imbalance exceeds a ratio of 1.2x (as in D-Vlog: 555 vs. 406 samples), per-class sample weights are applied to the loss to prevent bias towards the majority class.

3.4.2 Optimizer and Learning Rate Schedule

The model is optimised using AdamW [29], which decouples weight decay from the gradient update:

$$\theta_{t+1} = \theta_t - lr * (m_t / (\sqrt{v_t} + \epsilon)) - lr * \lambda * \theta_t \dots (Eq.3.33)$$

where $lr = 3e-4, \lambda = 1e-3, \epsilon = 1e-8, \beta_1 = 0.9, \beta_2 = 0.999$.

A two-phase learning rate schedule is applied:

Phase 1 — Linear warmup over $T_{warmup} = 20$ epochs:

$$lr(t) = lr_{max} * t / T_{warmup} \dots (Eq.3.34)$$

Phase 2 — Cosine annealing:

$$lr(t) = lr_{min} + 0.5 * (lr_{max} - lr_{min}) * (1 + \cos(\pi * t / T_{max})) \dots (Eq.3.35)$$

where $lr_{max} = 3e-4$ and $lr_{min} = 1e-6$.

3.4.3 Regularization Strategy

Four complementary regularisation techniques are employed:

- (i) Dropout ($p = 0.3$) applied within Mamba blocks and the classification head (Eq. 3.31c).
- (ii) Weight decay ($\lambda = 1e-3$) via AdamW (Eq. 3.33).
- (iii) Label smoothing ($\epsilon_{ls} = 0.1$) in the loss function (Eq. 3.32).
- (iv) Data augmentation — Gaussian noise injection (Eq. 3.2) and time masking applied to the training split only.

3.4.4 Early Stopping and Model Selection

Training runs for a maximum of 300 epochs with a batch size of 32 and mixed-precision (FP16) computation enabled on CUDA devices. Early stopping is applied with a patience of 40 epochs on validation macro-F1. The model checkpoint achieving the highest validation macro-F1 is used for final test set evaluation.

3.5 Evaluation Strategy

BCM-Net is evaluated on binary depression classification using five standard metrics. Let TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives respectively.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \dots (Eq.3.36)$$

$$Precision = TP / (TP + FP) \dots (Eq.3.37a)$$

$$Recall = TP / (TP + FN) \dots (Eq.3.37b)$$

$$F1 = 2 * Precision * Recall / (Precision + Recall) \dots (Eq.3.37c)$$

In addition, the Area Under the Receiver Operating Characteristic curve (AUC-ROC) is reported as a threshold-independent measure of discriminative ability. Macro-F1 is the primary metric because it penalises class-level imbalance equally. Where reported by prior work [2, 3, 5, 10], results are also broken down by precision and recall to enable fine-grained comparison.

3.6 Implementation Details

BCM-Net is implemented in PyTorch using the mamba-ssm library [20] for the Mamba SSM blocks. All experiments are conducted on an NVIDIA GPU with mixed-precision (AMP) training enabled. A fixed random seed is set for NumPy, PyTorch, and CUDA operations to ensure reproducibility.

Table 3.1: BCM-Net Hyperparameter Configuration

Hyperparameter	Value
Hidden dimension (d)	256
SSM state dim (d_state)	16
Conv kernel (d_conv)	4
Attention heads (n_heads)	4
Dropout rate	0.3
Peak learning rate	3E-04
Weight decay (lambda)	1E-03
Batch size	32
Max Epochs	300
Early stop patience	40
Label smoothing (eps_ls)	0.1
Noise std (sigma)	0.05
Time mask ratio	10%
Max sequence length	1000

Chapter 4

Results and Observations

This chapter presents the quantitative evaluation of BCM-Net on D-Vlog and LMVD, followed by a comparative analysis, efficiency discussion, observations, and limitations.

4.1 Experimental Setup

BCM-Net is evaluated on two datasets:

- D-Vlog [1]: 961 videos (555 depressed, 406 normal); eGeMAPS (25-dim) acoustic + facial landmark (136-dim) visual features.
- LMVD [4]: 1,823 videos (908 depressed, 915 normal); same protocol.

Metrics: Accuracy (Acc), Precision (Pr), Recall (Rc), macro-F1 (F1). The best-validation-F1 checkpoint is used for all test evaluations.

37

4.2 Quantitative Results

4.2.1 Results on D-Vlog

Table 4.1 compares BCM-Net against published baselines on D-Vlog.

Table 4.1: Comparative Results on D-Vlog Dataset

Model	Acc	Pr	Rc	F1
DepMSTAT [24]	--	0.7153	0.7560	0.7351
DepMamba [16]	0.6887	0.6819	0.8699	0.7644
EMO-Mamba [26]	0.7540	0.7579	0.7554	0.7566
MMFformer [13]	0.8108	0.8924	0.9380	0.9092
BCM-Net	0.8137	--	--	0.8137

BCM-Net achieves Acc = 0.8137 and F1 = 0.8137 on D-Vlog, surpassing all Mamba-based baselines (EMO-Mamba: +5.71 pp, DepMamba: +4.93 pp) and all other baselines by large margins (up to +22.37 pp over Depressformer). BCM-Net's accuracy (0.8137) marginally exceeds MMFformer's (0.8108), but MMFformer leads in F1 (0.9092) due to its substantially higher recall (0.9380) — reflecting a precision-recall trade-off rather than a fundamentally superior representation.

4.2.2 Results on LMVD

Table 4.2 reports performance on LMVD against the available baselines.

Table 4.2: Comparative Results on LMVD Dataset

Model	Acc	Pr	Rc	F1
DepMamba	0.7213	0.7018	0.7656	0.7320
LMTformer	0.8276	0.8287	0.8276	0.8274
MMFformer	0.8071	0.9033	0.9112	0.9048
BCM-Net	0.8132	0.8132	0.8132	0.8185

On LMVD, BCM-Net outperforms DepMamba by +8.63 pp F1 (the largest margin in this study) and LMTformer by +1.09 pp F1. The near-identical Pr, Rc, and Acc values reflect balanced class handling on this near- balanced dataset. BCM-Net trails MMFformer by −6.65 pp F1.

4.2.3 Cross-Dataset Consistency

Table 4.3: Performance Stability Across Datasets

Metric	D-Vlog	LMVD	Delta
Accuracy	0.8137	0.8132	-0.0005
F1	0.8137	0.8185	0.0046

BCM-Net's performance is virtually unchanged across two datasets with different class distributions and source domains, confirming that the model generalises without overfitting to dataset-specific priors.

4.3 Computational Efficiency Analysis

Table 4.4: Theoretical Complexity Comparison Across Models

Property	MMFformer	DepMamba	BCM-Net
Temporal encoder	Self-attn $O(T^2)$	Mamba $O(T)$	Mamba $O(T)$
Cross-modal fusion	Cross-attn $O(T^2)$	Progressive $O(T)$	Bottleneck Cross-attn
Fusion complexity	Quadratic	Linear	Sub-quadratic
Memory scaling	Quadratic	Linear	Linear
Inference mode	KV-Cache	Recurrent $O(1)$	Recurrent $O(1)$
Reported speedup	1x (baseline)	~15x faster	~ 15x faster
D-Vlog F1	0.9092	0.7644	0.8137
LMVD F1	0.9048	0.7320	0.8183

Table 4.5: Measured Parameters and FLOPs (Available Models)

Model	Params	FLOPS
DepMSTAT	2.91 M	7.13 G
LMTformer	0.95 M	1.10 G
DepMamba	—	—
EMO-Mamba	—	—
MMFformer	—	—
BCM-Net	1.30 M	0.25 G

BCM-Net occupies a deliberate middle ground between MMFformer and DepMamba. It adopts Mamba's $O(T)$ temporal encoder from DepMamba — preserving linear memory scaling and recurrent $O(1)$ inference — while introducing a bottleneck cross-attention fusion stage that is more expressive than DepMamba's simple progressive fusion yet far cheaper than MMFformer's full $O(T^2)$ cross-attention. This design yields competitive F1 scores (+5–8 pp above DepMamba) at computational costs comparable to DepMamba, making BCM-Net an efficient and effective architecture for real-world clinical deployment.

BCM-Net demonstrates that strong multimodal depression detection does not require large model capacity, provided the architecture couples temporal encoding and cross-modal fusion efficiently.

4.4 Observations

4.4.1 Mamba Encoder vs. Other Temporal Encoders

All Mamba-based models outperform CNN and transformer-lite baselines, validating the suitability of selective state space models for long, variable-duration clinical recordings where depression cues are temporally sparse. Within the Mamba family, BCM-Net's additional cross-attention fusion provides a consistent +5–9 pp F1 gain, confirming that the fusion strategy is as critical as the temporal encoder.

4.4.2 Effect of Bidirectional Cross-Attention

BCM-Net's +4.93 pp improvement over DepMamba (which also uses Mamba but with hierarchical fusion rather than explicit cross-attention) demonstrates the value of frame-level inter-modal alignment. The bidirectional attention allows the model to correlate co-occurring audio-visual markers of depression (e.g., reduced vocal energy coinciding with flattened brow or downward gaze) that modality-independent encoders cannot capture.

4.4.3 Precision-Recall Balance

On the balanced LMVD dataset, BCM-Net's Pr (0.8133), Rc (0.8132), and Acc (0.8132) are nearly identical — a hallmark of symmetric class treatment enabled by label smoothing. MMFformer's F1 advantage (0.9092 on D-Vlog, 0.9048 on LMVD) is driven primarily by very high recall (0.9380, 0.9112). While high sensitivity is clinically desirable in screening, it comes at the cost of precision

and quadratic compute. BCM-Net's balanced profile is better suited to settings where false positive cost is non-trivial and computational resources are limited.

4.5 Summary

BCM-Net achieves competitive and stable performance on both benchmark datasets, with an F1 score of 0.8137 on D-Vlog and 0.8183 on LMVD. The results show that BCM-Net outperforms all the Mamba-based models considered in this study, surpassing EMO-Mamba by 5.71 pp and DepMamba by up to 8.63 pp in F1, and also performing significantly better than CNN and transformer-lite methods, with up to 22 pp improvement in F1. The comparable performance of D-Vlog and LMVD, two datasets with different class distributions and source domains, shows that the design of BCM-Net has a strong generalisation ability, i.e., the gated fusion and attention pooling. Meanwhile, BCM-Net does this with much lower computational cost than attention-based models, providing $\sim 15\times$ faster inference, thanks to Mamba's linear-time recurrent scan, making it suitable for scalable clinical deployment. The results overall confirm BCM Net as an effective, efficient and generalisable architecture for automated multimodal depression detection.

Report

ORIGINALITY REPORT

10%

SIMILARITY INDEX

7%

INTERNET SOURCES

7%

PUBLICATIONS

5%

STUDENT PAPERS

PRIMARY SOURCES

1 Submitted to The University of Manchester <1 %
Student Paper

2 Wensheng Wang, Xueli Liu, Linxin Bai, Xiaobo Yang, Xinrong Chen. "Visual sensing approach for high-precision pressure estimation in robotic manipulation", Engineering Applications of Artificial Intelligence, 2025 <1 %
Publication

3 philarchive.org <1 %
Internet Source

4 Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) <1 %
Student Paper

5 Yiwei Wei, Chunlei Wu, Guohe Li, Haitao Shi. "Sequential Transformer via an Outside-In Attention for image captioning", Engineering Applications of Artificial Intelligence, 2022 <1 %
Publication

6 www.buildfastwithai.com <1 %
Internet Source

7 T. D'haene, R. Pintelon, G. Vandersteen. "An Iterative Method to Stabilize a Transfer Function in the s - and z -Domains", IEEE Transactions on Instrumentation and Measurement, 2006 <1 %
Publication

8 www.next.gr <1 %
Internet Source

9	Submitted to Associatie K.U.Leuven Student Paper	<1 %
10	www.ibm.com Internet Source	<1 %
11	Submitted to University of Bradford Student Paper	<1 %
12	Zemin Qin, Yuheng Li. "DCAM-DETR: Dual Cross-Attention Mamba Detection Transformer for RGB-Infrared Anti-UAV Detection", Information, 2026 Publication	<1 %
13	"Knowledge Science, Engineering and Management", Springer Science and Business Media LLC, 2019 Publication	<1 %
14	Tianqi Liu, Wei Cheng. "Research on Forecasting AC Power Output of Photovoltaic Systems Based on a Lightweight Hybrid Transformer-GRU Model", 2025 IEEE 5th International Conference on Applied Mathematics, Modeling and Computer Simulation (AMMCS), 2025 Publication	<1 %
15	zagan.unizar.es Internet Source	<1 %
16	Submitted to Amrita Vishwa Vidyapeetham Student Paper	<1 %
17	Shenqi Lai, Zhenhua Chai, Xiaolin Wei. "Transformer Meets Part Model: Adaptive Part Division for Person Re-Identification", 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2021 Publication	<1 %
18	Submitted to University of Sydney	

<1 %

19

Zhixing Tan, Shuo Wang, Zonghan Yang, Gang Chen, Xuancheng Huang, Maosong Sun, Yang Liu. "Neural machine translation: A review of methods, resources, and tools", AI Open, 2020

Publication

<1 %

20

Submitted to Atlantic Technological University

Student Paper

<1 %

21

Submitted to Birla Institute Of Technology & Science - Pilani

Student Paper

<1 %

22

Submitted to Universiti Teknologi Petronas

Student Paper

<1 %

23

air.unimi.it

Internet Source

<1 %

24

www.spandidos-publications.com

Internet Source

<1 %

25

Momiao Xiong. "Artificial Intelligence and Causal Inference", CRC Press, 2022

Publication

<1 %

26

Submitted to University of Wollongong

Student Paper

<1 %

27

link.springer.com

Internet Source

<1 %

28

Harvinder Singh, Priyanka Kaushal, Sarabjeet Kaur. "Advanced Computing and AI for Sustainable Environment, Energy, and Smart Manufacturing - AI-Driven Sustainability Across Environment, Energy, and Industry", CRC Press, 2026

Publication

<1 %

29	Submitted to Imperial College of Science, Technology and Medicine Student Paper	<1 %
30	Submitted to Nazarbayev University Student Paper	<1 %
31	Submitted to University of Hong Kong Student Paper	<1 %
32	Submitted to University of Lincoln Student Paper	<1 %
33	adilsarsenov.dev Internet Source	<1 %
34	www.seokdev.site Internet Source	<1 %
35	Felipe de Lima Peressim. "Machine learning for the prediction of 28-day cement compressive strength", Universidade de São Paulo. Agência de Bibliotecas e Coleções Digitais, 2025 Publication	<1 %
36	assets-eu.researchsquare.com Internet Source	<1 %
37	www.arxiv-vanity.com Internet Source	<1 %
38	www.mdpi.com Internet Source	<1 %
39	Yuhang Duan, Lin Lin, Jinyuan Liu, Qing Zhang, Xin Fan. "Cross-Domain Time-Frequency Mamba: A More Effective Model for Long-Term Time Series Forecasting", Knowledge-Based Systems, 2026 Publication	<1 %
40	par.nsf.gov Internet Source	<1 %

41

public-pages-files-2025.frontiersin.org

Internet Source

<1 %

42

www.themoonlight.io

Internet Source

<1 %

Exclude quotes On
Exclude bibliography On

Exclude matches < 9 words

Report

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23