computation

MDPI

Article

Feedback-Based Validation Learning

Chafik Boulealam [1,*], Hajar Filali [1,2], Jamal Riffi [1], Adnane Mohamed Mahraz [1] and Hamid Tairi [1]

[1] Laboratoire d'Informatique, Signaux, Automatique et Cognitivisme (LISAC), Department of Computer Science, Faculty of Science Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez 30000, Morocco
[2] Laboratory of Innovation in Management and Engineering (LIMIE), Institut Supérieur d'Ingénierie et des Affaires(ISGA), Fez 30000, Morocco
* Correspondence: chafik.boulealam@usmba.ac.ma

Abstract

This paper presents Feedback-Based Validation Learning (FBVL), a novel approach that transforms the role of validation datasets in deep learning. Unlike conventional methods that utilize validation datasets for performance evaluation post-training, FBVL integrates these datasets into the training process. It employs real-time feedback to optimize the model's weight adjustments, enhancing prediction accuracy and overall model performance. Importantly, FBVL preserves the integrity of the validation process by using prediction outcomes on the validation dataset to guide training adjustments, without directly accessing the dataset. Our empirical study conducted using the Iris dataset demonstrated the effectiveness of FBVL. The Iris dataset, comprising 150 samples from three species of Iris flowers, each characterized by four features, served as an ideal testbed for demonstrating FBVL's effectiveness. The implementation of FBVL led to substantial performance improvements, surpassing the accuracy of the previous best result by approximately 7.14% and achieving a loss reduction greater than the previous methods by approximately 49.18%. When FBVL was applied to the Multimodal EmotionLines Dataset (MELD), it showcased its wide applicability across various datasets and domains. The model achieved a test-set accuracy of 70.08%, surpassing the previous best-reported accuracy by approximately 3.12%. These remarkable results underscore FBVL's ability to optimize performance on established datasets and its capacity to minimize loss. Using our FBVL method, we achieved a test set f1_score micro of 70.07%, which is higher than the previous best-reported value for f1_score micro of 67.59%. These results demonstrate that FBVL enhances classification accuracy and model generalization, particularly in scenarios involving small or imbalanced datasets, offering practical benefits for designing more efficient and robust neural network architectures.

Keywords: real-time feedback; deep learning (DL); generalization; weight adjustment; model optimization; neural network training; Feedback-Based Validation Learning (FBVL)

check for updates

Academic Editor: Junlin Hu

Received: 10 May 2025
Revised: 17 June 2025
Accepted: 17 June 2025
Published: 1 July 2025

Citation: Boulealam, C.; Filali, H.; Riffi, J.; Mahraz, A.M.; Tairi, H. Feedback-Based Validation Learning. *Computation* **2025**, *13*, 156. https://doi.org/10.3390/computation13070156

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

1.1. Problem Statement and Motivation

Deep learning models traditionally treat validation datasets as passive evaluators, only used post-training to assess performance. This static approach limits the potential of validation data to actively guide the training process, often leading to overfitting, poor generalization, and inefficient parameter tuning, particularly in complex tasks like multimodal emotion recognition [1,2]. Existing methods such as early stopping, dropout, and k-fold cross-validation [3] mitigate overfitting but lack dynamic adaptation to validation feedback

during training. Reinforcement learning frameworks [4] introduce feedback loops but are constrained to policy optimization in specialized environments.

To address these limitations, we propose Feedback-Based Validation Learning (FBVL), a novel mechanism that integrates validation datasets into the training process. FBVL employs real-time feedback to adjust model weights, enhancing prediction accuracy while preserving the validation set's integrity. Unlike cross-validation [3], which partitions data iteratively, FBVL continuously refines model parameters using validation gradients (see Section 4.4), enabling dynamic adaptation across different domains. Empirical evaluations on the Iris [5] and MELD [1,2] datasets demonstrated FBVL's effectiveness, achieving a 7.14% accuracy improvement on Iris [5] and surpassing state-of-the-art models like SACL-LSTM and M2FNet by 3.12% on MELD [1,2] [Section 5.5].

1.2. Literature Review

FBVL builds on foundational theories in feedback mechanisms and addresses gaps in existing approaches:

- Reinforcement Learning [4]: Jones and Silver [6] highlight feedback loops for policy refinement in complex environments, but their focus on decision-making diverges from neural weight optimization.
- Cross-Validation: Introduced by Kohavi [3], this method partitions data iteratively to mitigate overfitting but lacks real-time feedback integration.
- Self-Refinement: Techniques like Self-Refine [7] rely on post-processing adjustments rather than dynamic training updates.
- Real-Time Evaluation: Recent work by Zhang et al. [8] emphasizes immediate feedback for parameter optimization but requires structured feedback mechanisms absent in many frameworks.

FBVL uniquely bridges these gaps by integrating real-time validation feedback into weight updates, ensuring both dynamic adaptation and validation integrity [Section 4.1.8].

1.3. Core Novelty of FBVL

FBVL introduces a periodic feedback mechanism that transforms validation datasets from passive evaluators into active participants in training. Unlike traditional methods (e.g., cross-validation [3], reinforcement learning [4]), FBVL dynamically adjusts weights by leveraging validation insights at predefined intervals (e.g., every five iterations), ensuring both training efficiency and validation integrity [Section 3.1.3].

1.4. Paper Structure

This paper is organized as follows:

- Section 2 reviews related works on feedback mechanisms, including cross-validation [3], reinforcement learning [4], and self-refinement [7].
- Section 3 details FBVL's methodology, including feedback calculation (Section 4.4) and implementation (Section 5.4).
- Section 4 analyzes the results, comparing FBVL to baselines like SACL-LSTM and M2FNet [9].
- Section 5 discusses the limitations, reproducibility, and ethical implications, including computational trade-offs and real-world applications.

2. Related Works

The exploration of feedback mechanisms within machine learning (ML) has advanced significantly, highlighting feedback's crucial role in adaptive learning systems. Feedback mechanisms are essential for improving model performance, allowing systems to adjust

based on real-time data and outcomes. This examination underscores the strides made in integrating feedback into training processes, particularly in industrial and technical domains. Recent studies, such as Vibration-Based Anomaly Detection in Industrial Machines: A Comparison of Autoencoders and Latent Spaces [10], emphasized the importance of latent space modeling for refining neural network performance. These works demonstrated how feedback mechanisms enhance robustness in industrial applications by dynamically adjusting model parameters based on real-time data. Similarly, reinforcement learning frameworks [6] and cross-validation techniques [3] have shown versatility in various domains, including unsupervised [11] and semi-supervised learning [12]. However, FBVL uniquely bridges these gaps by integrating real-time validation feedback into weight updates, ensuring both dynamic adaptation and validation integrity.

In reinforcement learning, Hafner and Riedmiller [6] have notably illustrated how feedback loops can refine policy decisions, enhancing ML models' capabilities in complex environments. Their work emphasizes the importance of feedback in improving learning processes, leading to more efficient models. Feedback mechanisms have also shown versatility in various domains. For instance, Tan and Li. [13] applied feedback-based algorithms in control systems, achieving significant improvements in stability and operational efficiency. Similarly, Madaan and Tandon [7] focused on iterative model refinement through validation feedback, demonstrating how continuous feedback can enhance accuracy and performance. Liu et al. [14] further underscored feedback's efficacy in guiding optimization strategies, resulting in faster convergence and better-performing models. Burgess et al. [15] explored structured feedback in clinical education, highlighting its role in bridging theory-practice gaps and enhancing learner performance.. Dynamic learning rate adjustment methods [16] have been explored, discussing how real-time feedback during training can enhance model performance by allowing adaptive learning rates to respond to the training process. This aligns closely with the principles of FBVL.

Moreover, a study by Hawkins et al. [8]. discussed the implementation of a real-time evaluation system for machine learning models in clinical settings. This research emphasizes the integration of immediate feedback to optimize model performance, aligning well with the goals of FBVL.

Additionally, cross-validation is a relevant method that shares the goal of optimizing model performance through iterative evaluation. Introduced by Ron Kohavi in 1995 [3], this technique involves partitioning data into subsets for training and testing, helping to mitigate overfitting and providing a more reliable estimate of model performance. Cross-validation complements FBVL by ensuring that models are validated against unseen data, thereby reinforcing the integrity of the learning process. Recent advancements have also explored feedback mechanisms in unsupervised [11] and semi-supervised learning [12], as well as multi-agent systems [17], further emphasizing the universal applicability of feedback in various ML paradigms.

In the context of neural architecture search, Chauhan et al. [18] proposed DQNAS, an automated framework that utilizes reinforcement learning principles to guide the design of high-performing neural networks. Zoph and Quoc V. Le [19] employed recurrent networks to generate model descriptions, optimizing architectures based on validation performance. The research on the Development and Validation of the Feedback in Learning Scale (FLS) by Jellicoe et al. [20] explored feedback delivery mechanisms in educational settings, enhancing our understanding of how feedback can support learning. Overall, the studies discussed illustrate the potential of feedback mechanisms to enhance ML models and lay the groundwork for future innovations. As machine learning continues to evolve, the role of feedback in refining and improving models is expected to grow.

The studies summarized in Table 1 illustrate the diverse applications of feedback mechanisms in machine learning. However, many existing methods do not fully leverage validation datasets during the training process. FBVL addresses this gap by integrating real-time feedback from validation data, allowing for continuous model refinement and improved performance. By comparing FBVL with these established methods, it becomes clear that FBVL not only enhances model accuracy but also fosters a more adaptive learning environment.

Table 1. Comparative analysis of feedback mechanisms in ML research.

Study/Authors	Methodology Description	Key Findings	Gaps Addressed by FBVL
Hafner and Riedmiller [6]	Feedback loops in reinforcement learning	Enhanced policy decisions in complex environments	Lack of integration of validation feedback in training
Tan and Li. [13]	Feedback-based algorithms in control systems	Improved stability and efficiency	Limited application to neural networks
Madaan and Tandon [7]	Iterative model refinement through validation feedback	Higher accuracy and performance	Need for real-time feedback integration
Peng et al. [14]	Feedback mechanisms in dynamic multiobjective optimization	Faster convergence and better performance	Absence of a structured validation approach
Burgess et al. [15]	Structured feedback in clinical education	Bridging theory-practice gaps through systematic feedback	Limited to human-centric applications; contrasts with FBVL's real-time integration in ML training
Dynamic Learning Rate [16]	Real-time feedback for learning rate adjustments	Enhanced model performance	Lack of systematic validation integration
Real-Time Evaluation [8]	Immediate feedback for model parameter optimization	Improved overall performance	Need for structured feedback mechanisms
Cross-Validation [3]	Iterative partitioning of data for training and testing	Mitigates overfitting provides reliable performance estimates	Does not utilize real-time feedback for adjustments

3. Materials and Methods

3.1. FBVL Configuration and Execution

The Feedback-Based Validation Learning (FBVL) mechanism introduces an innovative methodology for training neural network models by integrating the validation set into the training process. Unlike traditional approaches, FBVL leverages validation data for dual purposes: evaluating model performance and providing real-time feedback to guide weight adjustments. This integration enhances generalization and reduces overfitting by dynamically refining predictions based on validation outcomes. The FBVL workflow is structured into three core phases: feedback configuration, validation example determination, and iterative training execution.

3.1.1. Feedback and Method Configuration

Initially, the feedback type and the method for executing the validation set are established. The feedback can be categorized as 'Weighted Average', 'Average', or 'Variance', and the method can be classified as 'all', 'batch', or 'variable'.

- Feedback Type:
  - Weighted Average: Combines gradients from both training and validation sets via Equation (2) [Section 4.4.1], ensuring balanced influence (fbvl_alpha $\in$ [0.5, 0.8]).

- Average: Computes the mean of gradients, offering simplicity but less adaptability.
- Variance: Uses gradient variability to penalize noisy validation signals, improving robustness.

- Method Selection:
  - Batch Method (fbvl_batch = 4): Balances computational efficiency and feedback frequency.
  - Variable Method: Adapts fbvl_batch dynamically for datasets with uneven distributions.

3.1.2. Validation Example Determination

The number of validation examples (fbvl_batch) utilized in each training iteration is determined based on the selected method. If 'all' is selected, all examples from the validation set are employed. If 'batch' is selected, the quantity of validation examples aligns with the training set's batch size. If 'variable' is selected, fbvl_batch fluctuates for each iteration between 1 and the entire count of instances within the validation set.

- 'All': All validation examples are used.
- 'Batch': fbvl_batch aligns with the training set's batch size (e.g., fbvl_batch = 4).
- 'Variable': fbvl_batch fluctuates between 1 and the full validation set size.

This ensures flexibility across datasets (Iris vs. MELD) while maintaining validation integrity [Section 4.1.8].

3.1.3. Training Loop Execution

The model undergoes training over a specified number of epochs. In each epoch, the following occurs:

1. Forward Propagation on Training Set

   The model executes forward propagation on the training set and computes the cost.

2. Forward Propagation on Validation Set

   The model performs forward propagation on a subset of the validation set (the subset size is determined by $fbvl_{batch}$) and computes the validation cost.

3. Feedback Calculation

   The feedback is derived based on the chosen type. For 'Weighted Average,' we combine gradients from both training and validation sets. With 'Average,' we compute the mean of these gradients. If 'Variance' is chosen, it involves calculating the ratio of the sum of the gradients to their combined variance. Each method provides a distinct approach to integrating training and validation feedback, influencing the model's weight adjustments for improved learning outcomes.

4. Backward Propagation

   The model executes backward propagation using the calculated feedback and the true labels of the training set to compute the gradients of the cost function.

5. Parameter Update

   The model updates its parameters (weights and biases) using the computed gradients.

3.2. Implementation and Evaluation of FBVL

The FBVL mechanism signifies a substantial departure from traditional methodologies, introducing a new dimension to the training process. Incorporating feedback from the validation set into the training process aims to enhance the model's ability to generalize to unseen data. The FBVL mechanism was implemented using the Python 3.11 programming

language and the PyTorch deep learning library. The following materials and methods were employed in the implementation and evaluation of FBVL:

3.2.1. Datasets

The Iris dataset [5,21] and the MELD [1,2] dataset were used to evaluate the FBVL mechanism. These datasets were divided into training and validation datasets. The training dataset was employed to train the model, while the validation dataset was used to provide real-time feedback to the model during training.

3.2.2. Model Architecture

The FBVL mechanism was integrated into an updated version of a simple Neural Network Architecture (NNA) [1,9]. A simple NNA was updated to include the FBVL mechanism for providing real-time feedback to the model during training.

3.2.3. Training Workflow

The model was trained using the training dataset. During training, the model received real-time feedback based on its prediction outcomes on the validation dataset. This feedback was used to adjust the model's weights for improved predictions in subsequent iterations. In addition to the established FBVL parameters, we have implemented a 'Periodic Feedback Interval' within our training process. This interval is set to every 5 iterations, dictating that feedback from the validation set is incorporated into the model's training at these specified intervals. This strategic approach allows for a balance between continuous model adjustment based on immediate feedback and the computational efficiency of processing larger batches of data before adjustment. It enhances the adaptability of the FBVL mechanism to different dynamics, further optimizing the model's learning and generalization capabilities

3.2.4. Evaluation Methodology

The performance of the model was evaluated based on its prediction accuracy on the validation dataset. The prediction accuracy of the model was compared with relevant state-of-the-art methods to demonstrate the effectiveness of the FBVL mechanism.

The proposed FBVL mechanism can be incorporated into any neural network architecture, demonstrating its wide applicability and effectiveness. The detailed implementation and evaluation of the FBVL mechanism are presented in the following sections.

3.3. Model Configuration and Training

The FBVL mechanism was seamlessly integrated into a simple Neural Network Architecture (NNA), enhancing it with real-time feedback capabilities during the training process. To fully harness the potential of the FBVL mechanism, specific parameters were meticulously configured:

- FBVL Activation: Enabled (fbvl = True), signifying the active use of validation feedback during training.
- FBVL Batch: Set to 4 ($fbvl_{batch} = 4$), dictating the feedback incorporation frequency from the validation data.
- FBVL Alpha: A value of 0.5 ($fbvl_{alpha} = 0.5$), determining the weighting of validation feedback in the training updates.
- Feedback Method: 'Weighted Average,' chosen for its balanced approach in integrating insights from the validation set into the training process.

A critical component of our training methodology is the Periodic Feedback Interval, implemented to provide feedback from the validation set every 5 iterations. This approach

ensures the model periodically assimilates validation insights, optimizing for computational efficiency and model adaptability. Throughout the training loop, the model engages in forward propagation using both training and selected validation subsets. Feedback, computed via the 'Weighted Average' method, informs subsequent backward propagation and parameter adjustments. This iterative feedback loop is designed to significantly enhance the model's generalization capabilities by leveraging nuanced insights from the validation data set.

3.3.1. Rationale for Hyperparameter Selection

$\text{fblv}_{\text{batch}}$

The $fbvl_{batch}$ parameter specifies the number of validation examples used to provide feedback during each training iteration. In our study, values of 4 and 8 yielded optimal results. These values strike a balance between delivering sufficient feedback and maintaining computational efficiency. Smaller batches (e.g., 4 or 8) enable frequent model updates, enhancing adaptability to training data. Larger batches (e.g., 16, 32, or 64) dilute the feedback effect, hindering responsiveness.

- Optimal Values: Experiments showed that $fbvl_{batch}$ values of 4 and 8 achieved the highest accuracy. Other values (2, 16, 32, 64) resulted in marginally lower performance.
- Learning Dynamics: Smaller batches facilitated rapid parameter updates, improving convergence speed. Larger batches risk over-smoothing gradients, reducing the model's ability to refine predictions.

$\text{fbvl}_{\text{alpha}}$

The $fbvl_{alpha}$ parameter governs the weighting of validation feedback relative to training feedback. Values within [0.5, 0.8] were optimal, balancing influences from training and validation data to improve generalization.

- Optimal Range: The highest accuracy was achieved when $fbvl_{alpha}$ was set between 0.5 and 0.8. This range ensures validation feedback neither dominates nor underpowers training signals.
- Performance Decline: Values outside this range degraded accuracy. High $fbvl_{alpha}$ (>0.8) overemphasizes training data, while low values (<0.5) over-rely on validation data, both compromising model robustness.

4. Proposed Architecture with FBVL Mechanism

This section details the integration of the Feedback-Based Validation Learning (FBVL) mechanism into a neural network architecture (NNA). The design emphasizes real-time feedback from validation data to enhance generalization and adaptability.

4.1. Architecture Overview

The FBVL mechanism is embedded into a modular neural network architecture, enabling dynamic adjustments based on validation feedback. The architecture comprises the following components:

4.1.1. Input Layer

Datasets for both training and validation are fed into the model. While the validation dataset offers real-time feedback during training, the training dataset is used for the actual training.

4.1.2. Hidden Layers

These layers perform computations and learning through neurons whose weights are adjusted based on feedback from the validation dataset, in addition to the learning from the training dataset.

4.1.3. Output Layer

The model outputs predictions that are compared with the actual labels of the validation dataset to compute the prediction error.

4.1.4. Feedback Mechanism

This is the core of the FBVL, where real-time feedback is provided to the model based on its prediction outcomes from the validation dataset.

4.1.5. Weight Adjustment

Using feedback, the model's weights are adjusted. The adjustment is made such that the model improves its prediction accuracy on the validation dataset, which is indicative of improved generalization ability.

4.1.6. Training Process

The training process involves forward propagation, cost computation, backward propagation using the feedback, and weight adjustment. This is repeated for several epochs or until the performance on the validation dataset stops improving.

4.1.7. Real-Time Feedback Integration

FBVL utilizes real-time feedback from the validation dataset to dynamically adjust model weights during the training process. This ensures that the model does not directly access the validation data, preserving its role as an unbiased evaluator of the model's performance.

4.1.8. Generalization Improvement

The feedback from the validation set is used to compute gradients of the cost function, which are then used to update the model weights. This process is intended to help the model generalize better to new, unseen data.

4.2. Visualization of FBVL Integration

This architecture allows FBVL to enhance the model's performance during the training process itself, offering a more effective and efficient methodology for training neural network models. As shown in Figure 1, the FBVL mechanism be incorporated into any neural network architecture, demonstrating its wide applicability and effectiveness. The Feedback-Based Validation Learning (FBVL) mechanism can improve the performance of the model. This practice harnesses the full potential of FBVL, making it a powerful tool for enhancing any NNA. It is an innovative way to maximize the effectiveness of the validation dataset, not just for performance evaluation, but also for model enhancement during the training process itself. This could lead to exciting advancements in the field of deep learning and artificial intelligence.

4.3. Mathematical Formulation

The formula for the gradients in the Feedback-Based Validation Learning (FBVL) mechanism can be represented as follows:

$$\text{grads} = \text{mode_backward}(\text{dAL}_{\text{feedback}}) \tag{1}$$

where:

grads: Represents the gradients of the cost function with respect to the weights of the model.

model_backward: A function that performs the backward propagation step in the training process.

$dAL_{feedback}$: Represents the gradients of the loss function with respect to the predicted labels for the feedback dataset. It is important to note that $dAL_{feedback}$ can take different forms depending on the specific implementation of the FBVL mechanism.

This formula signifies that the feedback from the feedback dataset ($.dAL_{feedback}$) is utilized to adjust the model's weights during the training process. This unique utilization of the feedback dataset for providing real-time feedback during training is a distinguishing feature of the learning mechanism.

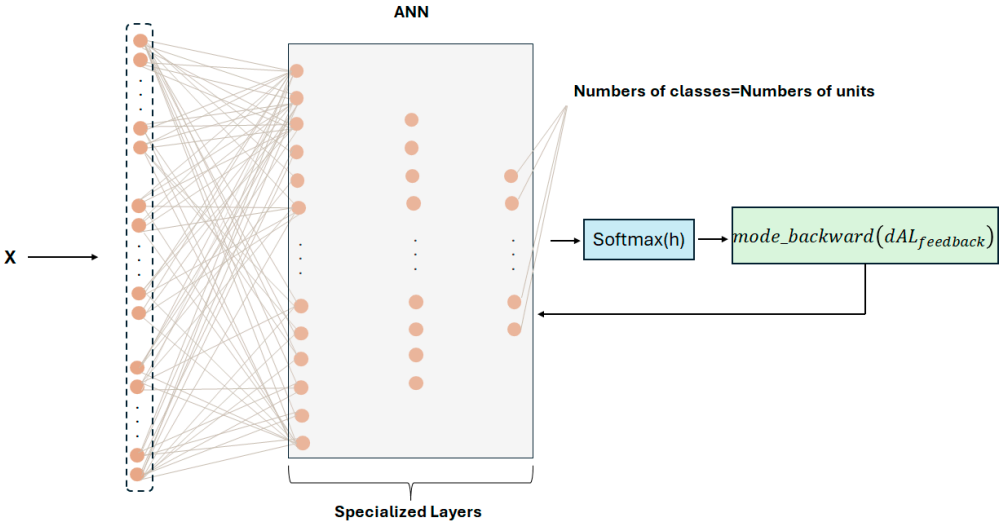


Figure 1. The integration of the FBVL mechanism into a simple NNA for classification.

The FBVL mechanism enhances the model's performance during the training process itself, marking a novel and effective approach to training neural network models. This innovative methodology holds the promise of driving future advancements in the field of deep learning and artificial intelligence, offering a more effective and efficient way to train models. The detailed implementation and evaluation of the FBVL mechanism are presented in the following sections.

4.4. Feedback Calculation Methods

4.4.1. Weighted Average (Formula (1))

$$dAL_{feedback} = \frac{dAL + \left(1 - fbvl_{alpha}\right) * dAL_{val}}{1 + \left(1 - fbvl_{alpha}\right)} \quad (2)$$

where:

dAL: This represents the gradient of the loss function with respect to the predicted labels for the training dataset.

AL_{val}: Refers to the gradient of the loss function with respect to the predicted labels for the validation dataset.

$fbvl_{alpha}$: A weighting factor controlling the influence of validation feedback.

$dAL_{feedback}$: Represents the adjusted gradient based on feedback.

This formula calculates the weighted average of the gradients of the loss function with respect to the predicted labels for the training dataset (dAL and the validation dataset

dAL_{val}). The weighting factor is determined by $fbvl_{alpha}$, which is a hyperparameter that can be adjusted by the user. If $fbvl_{alpha}$ is closer to 1, more weight is given to the training dataset, and if it is closer to 0, more weight is given to the validation dataset. The result, $dAL_{feedback}$, is the feedback that is used to adjust the model's weights during the training process. It addresses the challenge of overfitting versus underfitting, thereby enhancing the model's generalization capabilities. This refined approach underscores the nuanced application of feedback mechanisms in neural network training, significantly impacting model performance and efficiency.

4.4.2. Average (Formula (2))

$$dAL_{feedback} = \frac{dAL + dAL_{val}}{2} \tag{3}$$

where:

dAL: This stands for the gradient of the training dataset's loss function with regard to the predicted labels.

AL_{val}: For the validation dataset, this is the gradient of the loss function with respect to the predicted labels.

$dAL_{feedback}$: Represents the adjusted gradient based on feedback.

This formula calculates the average of the gradients from the training set (dAL) and the validation set (dAL_{val}). $dAL_{feedback}$ provides a balanced gradient value that reflects the central tendency of model adjustments based on both training and validation feedback. This average is crucial for ensuring the model's learning process is influenced evenly by insights from both datasets.

4.4.3. Variance (Formula (3))

This formula calculates the ratio of the sum of dAL and dAL_val to the variance of the combined AL and AL_{val}.

Let us denote the combined dAL and dAL_val as $X_{combined}$. So $X_{combined} = dAL + dAL_{val}$.

The variance (σ^2) of $X_combined$ can be calculated as:

$$\sigma^2_{X_{combined}} = \frac{1}{N}\sum_{i=1}^{N}\Big(x_i - \mu)^2 \tag{4}$$

where:

N: Represents the total number of observations $X_combined$.

x_i: Is the i-th observation in $X_combined$.

μ: Is the mean (average) of $X_combined$.

This formula assesses the dispersion of the combined gradients $X_combined$ (consisting of dAL and AL_{val}). Variance measures the spread of these gradients from their mean, quantifying the deviation of each gradient from the average and then averaging these deviations. Understanding this variance helps in gauging the diversity and spread of feedback from the training and validation sets, offering insights into the model's learning dynamics across different datasets.

$$dAL_{feedback} = \frac{X_{combined}}{\sigma^2_{X_{combined}}} = \frac{dAL + dAL_{val}}{\sigma^2_{X_{combined}}} \tag{5}$$

where:

dAL: This stands for the gradient of the training dataset's loss function with regard to the predicted labels.

AL_{val}: For the validation dataset, this is the gradient of the loss function with respect to the predicted labels.

$dAL_{feedback}$: Represents the adjusted gradient based on feedback.

This formula calculates the ratio of the sum of dAL and AL_{val} to the variance of their combined dataset $X_{combined}$. It measures how the total sum of the gradients compares to their dispersion, offering insights into the relationship between the magnitude of the model's gradient adjustments and their spread. This understanding is valuable for evaluating the model's learning process and adjusting strategies for optimal generalization.

4.4.4. Methods for Handling the Validation Set in FBVL

The Feature-Based Validation Learning (FBVL) mechanism can utilize various feedback formulas and methods for processing the validation set. The following outlines the three methods previously mentioned, each differentiated by the approach to handling the validation set and the type of feedback formula applied:

Method 1: $fbvl_{batch} = train_{batch}$

In this method, the validation set is the same size as the training batch. This method makes sure that the feedback one gets from the validation set is as directly comparable as possible to that which is obtained from the training set. The feedback formulas applied in this method are Formulas (1)–(3).

Method 2: $fbvl_{batch} = Y_{val}$ (All Validation Examples)

This method incorporates all examples in the validation set. It offers a thorough overview of the model's performance on unseen data. The feedback formulas used in this method are identical to those in Method 1.

Method 3: $fbvl_{batch} = n_{val}$ (One or More Examples)

In this method, the validation set contains one or more examples, depending on the value of n_{val}. While this approach can offer rapid feedback, it may not accurately represent the model's overall performance with unseen data, particularly if n_{val} is small. The feedback formulas used in this method are the same as those in the previous methods.

Each of these cases represents a different approach to providing feedback to the model during training. The best approach depends on the specific context and requirements of each task. The approach needs to be validated and its implications understood. It might be necessary to consult with a machine learning expert. Also, remember that these formulas do not provide direct feedback about the accuracy of the model's predictions. For that, it is typical to use a loss function that compares the model's predictions (AL for the training set) with the true labels. In the Feedback-Based Validation Learning (FBVL) mechanism, the model does not directly 'see' the features of the validation dataset during the training process. The model is trained on the training set, and the validation set is used to evaluate the model's performance on unseen data during the training process.

However, in FBVL, the feedback from the validation set is used in a unique way. The model's predictions on the validation set (denoted as AL_{val}) are used to compute certain feedback formulas, which are then used in the backward propagation step to compute the gradients of the cost function. This is a novel approach that differs from standard machine learning practice. It is important to note that while the validation set does not directly participate in the training of the model, it plays a crucial role in the FBVL mechanism. The feedback from the validation set, as computed by the feedback formulas, influences the update of the model weights during training. This is intended to help the model generalize better to new, unseen data.

4.5. Addressing Overfitting with FBVL

4.5.1. Overfitting

Overfitting occurs when a machine learning model learns the details and noise in the training data to such an extent that it negatively impacts the model's performance on new data. This means the model performs exceptionally well on the training data but poorly on unseen test data. Overfitting is a common problem in machine learning and can be identified when there is a significant gap between the training accuracy and the test accuracy.

4.5.2. Signs of Overfitting

- High accuracy on the training set.
- Low accuracy on the test set.
- The model captures noise and outliers in the training data.

4.5.3. How FBVL Can Avoid Overfitting

- Continuous Validation Feedback

FBVL uses the validation dataset during training to provide continuous feedback. This means the model is constantly evaluated on unseen data (validation set) during the training process, which helps in adjusting the model parameters to generalize better.

- Balanced Learning

By incorporating feedback from the validation set, FBVL ensures that the model does not solely focus on the training data. This balanced approach helps the model to learn patterns that are more generalizable to new data, rather than memorizing the training data.

- Dynamic Adjustment

FBVL dynamically adjusts the model's weights based on its performance on the validation set. This real-time adjustment helps in preventing the model from becoming too specialized to the training data, thus reducing the risk of overfitting.

- Regularization Effect

The feedback mechanism acts as a form of regularization. Regularization techniques are used to prevent overfitting by adding a penalty for complexity. Similarly, FBVL's feedback loop penalizes the model for poor performance on the validation set, encouraging simpler and more generalizable models.

- Early Stopping

FBVL can help in implementing early stopping. If the model's performance on the validation set stops improving or starts to degrade, the training process can be halted early. This prevents the model from overfitting the training data by stopping training before the model becomes too complex.

5. Suggested Experimental Findings and Analysis

5.1. Setting up a Computer

Our proposed model is implemented using Python Toolbox. The experiments were conducted on an MSI As desktop system with 100 GB of RAM, an Intel Core i9-10900K CPU running at 3.70 GHz, and a single NVIDIA GeForce RTX 3090 with 24 GB of memory. Ubuntu Server served as the operating system. Additionally, the proposed architecture underwent training using the Google Collab cloud service.

5.2. Dataset

For the evaluation of the Feedback-Based Validation Learning (FBVL) mechanism, this research utilized the renowned Iris dataset [5,21]. It is a classic in the field of machine learning and pattern recognition, comprising 150 samples from three species of Iris flowers (*Iris setosa*, *Iris virginica*, and *Iris versicolor*) [5,21]. Each sample is described by four features, namely, lengths and widths of the sepals and petals. This dataset was chosen for its clarity and simplicity, which allows for a straightforward assessment of FBVL's effectiveness in enhancing neural network performance through feedback-based adjustments during the training process.

In conjunction with the Iris [5] dataset, this study also incorporated the Multimodal EmotionLines Dataset (MELD) [1,2,9,22]. MELD [1,2] is an enriched multimodal and multi-party dataset based on the EmotionLines Dataset, comprising over 13,000 utterances from the "Friends" TV series [9,22], each labeled with one of seven emotions: anger, disgust, sadness, joy, surprise, fear, and neutral. It retains the same dialogue instances as EmotionLines, with additional modalities like audio and visual data. More than 1400 dialogues are distributed across 1039 for training, 114 for validation, and 280 for testing. The utterances follow suit with 9989 for training, 1109 for validation, and 2610 for testing. Annotations across dialogues encompass a spectrum of emotions: Anger (0), Disgust (1), Fear (2), Joy (3), Neutral (4), Sadness (5), and Surprise (6), along with Neutral and Non-Neutral expansions. Employing the MELD [1,2] dataset provides a holistic evaluation of the FBVL mechanism's performance over different modalities, upholding the comparability of results with leading methods through the use of predefined dataset splits. The structure of the MELD [1,2] dataset is detailed in Table 2. The distributions of emotion labels in these datasets are detailed in Table 3.

Table 2. Statistics of the MELD [1,2] dataset.

Dataset	**Utterances**			**Dialogues**		
	train	test	valid	train	test	valid
MELD [1,2]	9989	2610	1109	1039	280	114

Table 3. Distributions for each label in the test, validation, and training datasets.

	Disgust	**Sadness**	**Anger**	**Fear**	**Joy**	**Surprise**	**Neutral**
train	271	683	1109	268	1743	1205	4710
valid	22	111	153	40	163	150	470
test	68	208	345	50	402	281	1256

Table 2 delineates the structure of the MELD [1,2] dataset, showcasing its layered conversational features. An average of 9.6 turns per dialogue reflects the interactive exchanges, while an average of 2.7 parties per dialogue underscores the dataset's multi-party nature. Such rich contextual depth ensures a rigorous testing ground for the FBVL mechanism, fostering advancements in emotion recognition within intricate conversational scenarios.

FBVL's application resulted in an accuracy of 70.08% on the test set, demonstrating a substantial increase compared to previous approaches on the same dataset. These results underscore FBVL's capability to deliver state-of-the-art performance while maintaining robustness and reliability in multimodal emotion recognition tasks.

5.3. Measures of Evaluation

We used performance metrics including F1-score, Accuracy, Precision, and Recall to assess our model's performance. These measurements are defined as follows:

Accuracy: This simple performance statistic is used to calculate the ratio of correctly predicted observations to the total number of observations. It is represented mathematically as follows:

$$\text{Accuracy} = \frac{TN + TP}{TP + FP + TN + FN} \tag{6}$$

Recall: This score assesses how well the model recognizes True Positives. The representation of it is as follows:

$$\text{Recall} = \frac{TP}{FN + TP} \tag{7}$$

Precision: This is the ratio of accurately predicted positive observations to the total number of positive observations that were forecasted. This is one way to express it:

$$\text{Precision} = \frac{TP}{FP + TP} \tag{8}$$

F1-score: This is the precision and recall harmonic mean, and it is calculated as follows:

$$\text{F1} - \text{score} = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{9}$$

The True Positive, True Negative, False Positive, and False Negative are represented by the symbols TP, TN, FP, and FN in these formulas, respectively.

5.4. Performance Evaluation

To validate the effectiveness of Feedback-Based Validation Learning (FBVL), we conducted experiments on two datasets: the Iris dataset [5] (a classic tabular dataset) and the Multimodal EmotionLines Dataset (MELD) [1,2] (a complex multimodal conversational dataset). The results were analyzed across accuracy, loss, F1-score, and comparisons with state-of-the-art models.

5.4.1. Performance Study Without Using FBVL

We evaluated the baseline neural network architecture (NNA) on the Iris dataset [5] without FBVL. The model achieved a high training accuracy of 99.17% (loss: 0.05185) and a test accuracy of 93.33% (loss: 0.1346), as summarized in Tables 4 and 5 as summarized in Tables 4 and 5. Minor misclassifications occurred during training: two instances of versicolor and two of virginica (Table 5). According to the classification report (Table 6), performance was strong across all classes, with weighted and macro-averaged F1-scores of 0.95, indicating balanced precision and recall.. In the test phase, two virginica samples were misclassified (Table 7). These results suggest the baseline model generalizes well but exhibits slight instability in distinguishing closely related species.

Table 4. Results of the IRIS's [5] classification report based on Precision, Recall, and F1-score for the training dataset of IRIS [5] without using FBVL.

	Precision	**Recall**	**F1-Score**	**Support**
setosa	1.00	1.00	1.00	33
virginica	0.90	0.97	0.94	39
versicolor	0.97	0.88	0.92	33
accuracy			0.95	105
macro avg	0.96	0.95	0.95	105
Weighted avg	0.95	0.95	0.95	105

Table 5. Training dataset confusion matrix results obtained on the IRIS [1] dataset without using FBVL.

Actual\Predicted	0 (Setosa)	1 (Versicolor)	2 (Virginica)
0 (setosa)	31	0	0
1 (versicolor)	0	35	2
2 (virginica)	0	0	37

Table 6. Results of the IRIS's classification report based on Precision, Recall, and F1-score for the test dataset of IRIS without using FBVL [5].

	Precision	Recall	F1-Score	Support
setosa	1.00	1.00	1.00	6
virginica	1.00	0.83	0.91	12
versicolor	0.86	1.00	0.92	12
accuracy			0.93	30
macro avg	0.95	0.94	0.94	30
Weighted avg	0.94	0.93	0.93	30

Table 7. Testing dataset confusion matrix results obtained for the IRIS [5] test dataset without using FBVL.

Actual\Predicted	0 (Setosa)	1 (Versicolor)	2 (Virginica)
0 (setosa)	6	0	0
1 (versicolor)	0	10	2
2 (virginica)	0	0	12

As shown in Table 4, the test phase yielded a strong accuracy of 93.33% and a test loss of 0.1346. The performance across classes was consistently high, demonstrating the model's reliable predictive ability on unseen data.

The confusion matrix in Table 5 provides a detailed overview of the model's performance on the training dataset of the Iris dataset without utilizing the Feedback-Based Validation Learning (FBVL) mechanism.

Table 6 presents the classification report for the IRIS test dataset without using FBVL, showcasing metrics such as precision, recall, F1-score, and support. The results highlight strong performance across all classes, with weighted and macro-averages closely aligned, indicating balanced accuracy, precision, and recall. The overall accuracy is 93%, demonstrating effective classification despite the absence of FBVL.

Table 7 presents the test confusion matrix results for the IRIS dataset without using FBVL, illustrating the model's performance on unseen data with minimal misclassifications.

As shown in Figure 2, the training/validation cost curve for the IRIS dataset without using FBVL demonstrates that the maximum value of the cost is reached at approximately 0 epochs, after which it diminishes as the number of epochs increases. The cost stabilizes at a low value of 0.05185, indicating effective learning and convergence.

In Figure 3, the training accuracy/validation accuracy curves reveal that the training accuracy initially reaches a peak but then experiences fluctuations before stabilizing around 99.17%. This behavior suggests that the model achieves high accuracy on the training data but may exhibit some instability during the early stages of training. The validation accuracy, however, shows consistent performance, aligning closely with the training accuracy, which indicates good generalization without significant overfitting.

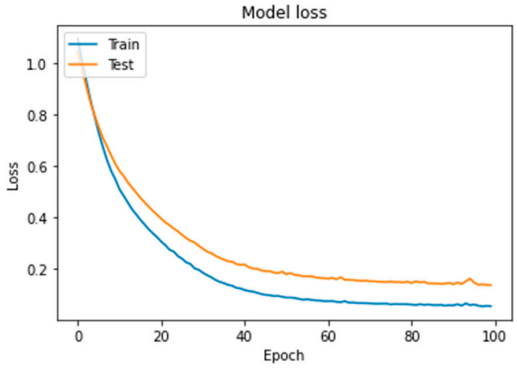


Figure 2. The training/validation cost versus epochs on the IRIS [5] dataset without using FBVL.

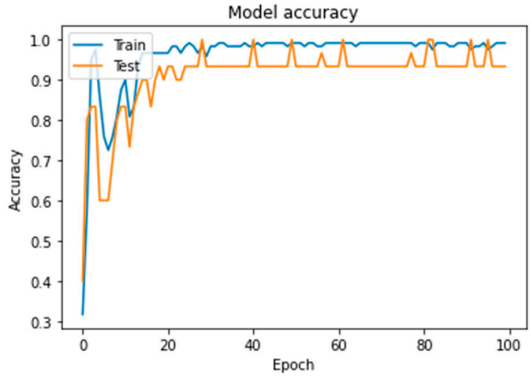


Figure 3. The training accuracy/validation accuracy versus epochs on the IRIS [5] dataset without using FBVL.

5.4.2. Performance Study Using FBVL

Table 8 presents the classification report for the training dataset of the IRIS dataset using the Feedback-Based Validation Learning (FBVL) mechanism. The results demonstrate exceptional performance, with near-perfect precision, recall, and F1-scores for all classes. Specifically, the model achieves 100% precision and recall for the setosa class, while minor deviations are observed for the virginica and versicolor classes. The overall accuracy is 98%, with macro and weighted averages closely aligned at 0.98, indicating balanced and reliable classification across all classes. These findings underscore the effectiveness of FBVL in enhancing model performance during the training phase.

Table 8. Results of the IRIS's classification report based on Precision, Recall, and F1-score for the training dataset of IRIS using FBVL [5].

	Precision	Recall	F1-Score	Support
setosa	1.00	1.00	1.00	33
virginica	1.00	0.95	0.97	39
versicolor	0.94	1.00	0.97	33
accuracy			0.98	105
macro avg	0.98	0.98	0.98	105
Weighted avg	0.98	0.98	0.98	105

Table 9 presents the training confusion matrix results for the IRIS dataset using the Feedback-Based Validation Learning (FBVL) mechanism. This table highlights the model's performance on the training data, showing near-perfect classification for the setosa and virginica classes, with only two misclassifications for versicolor.

Table 9. Training dataset confusion matrix results obtained for the IRIS [5] dataset using FBVL.

Actual\Predicted	0 (Setosa)	1 (Versicolor)	2 (Virginica)
0 (setosa)	33	0	0
1 (versicolor)	0	37	2
2 (virginica)	0	0	33

The results in Table 10 demonstrate perfect performance across all metrics, with precision, recall, and F1-score achieving 1.00 for all classes. The model exhibits flawless classification, as reflected in the confusion matrix, with no misclassifications observed. These outcomes highlight FBVL's ability to optimize neural network performance, achieving 100% accuracy and balanced generalization.

Table 10. Results of the IRIS's classification report based on Precision, Recall, and F1-score for the test dataset of IRIS using FBVL [5].

	Precision	Recall	F1-Score	Support
setosa	1.00	1.00	1.00	15
virginica	1.00	1.00	1.00	5
versicolor	1.00	1.00	1.00	10
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
Weighted avg	1.00	1.00	1.00	30

Table 11 presents the test confusion matrix for the IRIS dataset using FBVL, showcasing perfect classification across all classes with no misclassifications.

Table 11. Testing dataset confusion matrix results obtained for the IRIS [1] dataset using FBVL.

Actual\Predicted	0 (Setosa)	1 (Versicolor)	2 (Virginica)
0 (setosa)	15	0	0
1 (versicolor)	0	5	0
2 (virginica)	0	0	10

It is evident from Figure 4 that the maximum value of the cost curve is reached at approximately 0 epochs and diminishes as the number of epochs increases, eventually stabilizing at a cost of 0.0016. Regarding training accuracy (Figure 5), we observe that after initially reaching a peak, there is a slight decline before it stabilizes and subsequently increases to 99.22% after a set number of epochs. Additionally, for class 1, the maximum value of the true label in the multimodal confusion matrix is 25448, representing the highest classification confidence for that class.

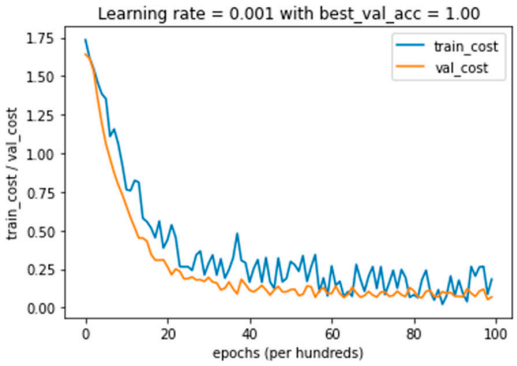


Figure 4. The training/validation cost versus epochs on the IRIS [5] dataset using FBVL.

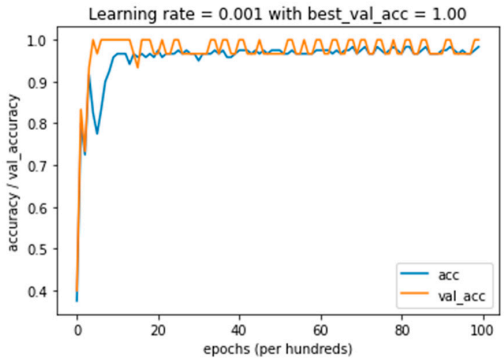


Figure 5. Accuracy of training and validation compared to epochs on the IRIS [5] dataset using FBVL.

Figure 6 compares the F1 scores for each category between models using FBVL and those not using FBVL for both training and test datasets. The F1 score is chosen for this comparison because it provides a balanced measure of a model's accuracy by considering both precision and recall. This makes it particularly useful for evaluating the overall performance of the models.

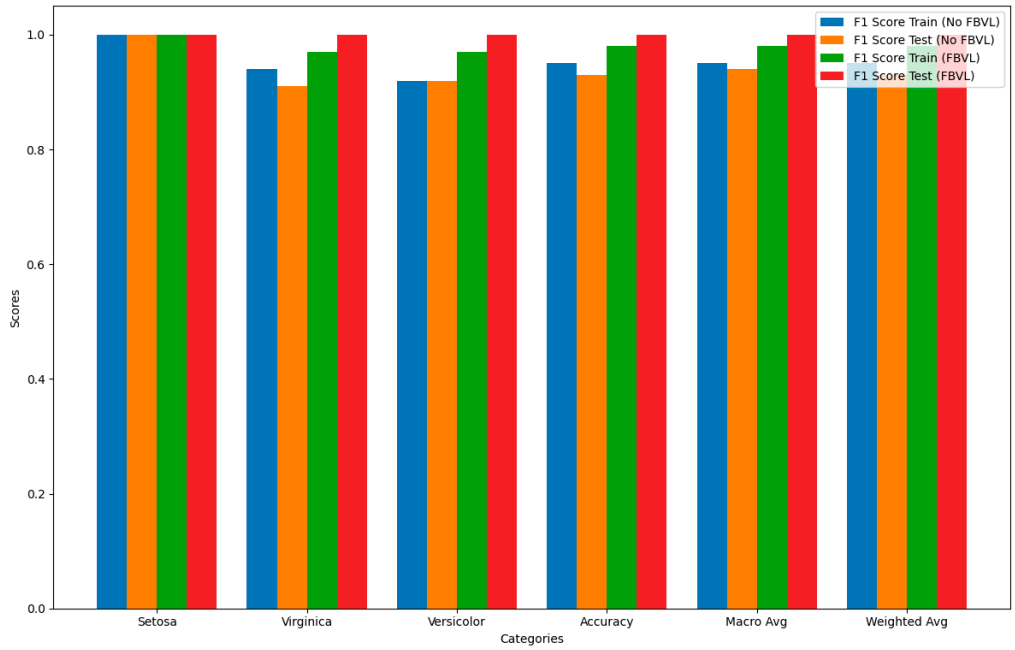


Figure 6. Precision scores by category and method for the IRIS [5] dataset.

- Training Data Without FBVL: Shows high F1 scores, but slightly lower for versicolor.
- Test Data Without FBVL: F1 scores are generally high but slightly lower for virginica.
- Training Data With FBVL: F1 scores are consistently high across all categories.
- Test Data With FBVL: F1 scores are perfect across all categories.

The use of FBVL enhances the F1 scores, indicating a balanced improvement in both precision and recall across most categories. This is especially evident in the test dataset, which shows perfect scores, highlighting the effectiveness of FBVL in reducing overfitting and improving generalization.

By focusing on the F1 score, this graph provides a clear and comprehensive view of the model's performance, demonstrating the significant benefits of incorporating Feedback-Based Validation Learning (FBVL). This comparison underscores the robustness and reliability of FBVL in various tasks, setting a new benchmark for future research in this domain.

Table 12 shows the evaluation of our mechanism, FBVL, compared with a simple NNA without FBVL.

Table 12. Comparative analysis of neural network architectures with and without FBVL on the IRIS [5] and MELD [1,2] datasets.

Models	Dataset	F1 Score (Weighted)	Loss	Accuracy (%)
NNA with FBVL	IRIS [5]	**100**	**0.0684**	**100**
NNA without FBVL	IRIS [5]	93.33	0.1345	93.33
NNA with FBVL	MELD [1,2]	**65.14**	0.1345	**70.08**
NNA without FBVL	MELD [1,2]	64.8	1.72	68.81

Figure 7 provides a visual comparison of the performance of neural network architectures with and without FBVL on the IRIS [5] and MELD [1,2] datasets, focusing on two key metrics: Accuracy (%) and F1 Score (weighted).

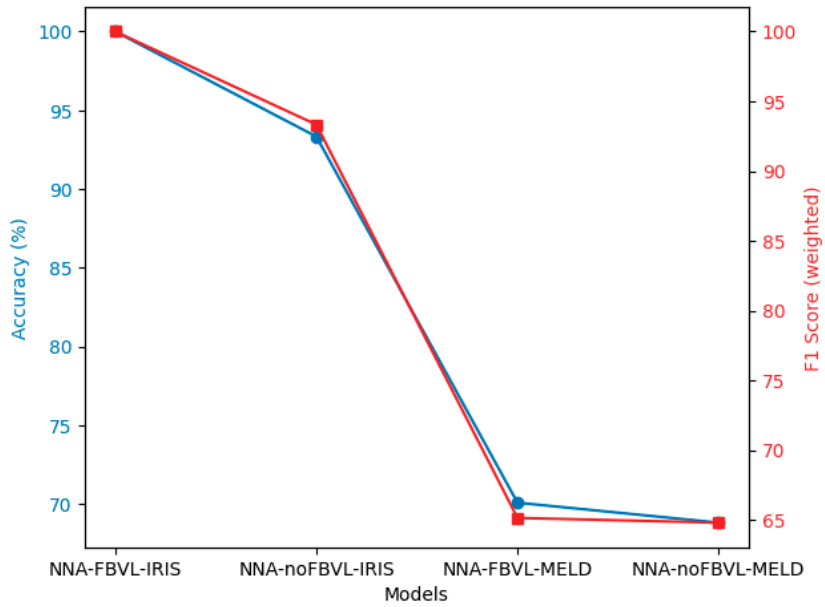


Figure 7. Comparative analysis of neural network architectures with and without FBVL on the IRIS [5] and MELD [1,2] datasets.

- The blue line represents the Accuracy (%) of each model. This metric indicates the proportion of correctly predicted instances out of the total instances, providing a measure of the overall effectiveness of each model.
- The red line represents the F1 Score (weighted) of each model. The F1 Score is the harmonic mean of precision and recall, and the weighted version accounts for the support (the number of true instances) of each class, offering a balanced measure of a model's accuracy in classifying different categories.

This visual representation clearly illustrates the degree of improvement achieved by the Feedback-Based Validation Learning (FBVL) model compared to a simple neural network architecture (NNA) without FBVL. The FBVL model shows superior performance in both accuracy and F1 Score (weighted), highlighting its effectiveness in enhancing model performance through the integration of real-time feedback from validation datasets. This comparison underscores the robustness and reliability of FBVL in various tasks, setting a new benchmark for future research in this domain.

As shown in Table 13, the comparative evaluation highlights that the FBVL model demonstrates superior performance, achieving an Accuracy of 70.08%, alongside F1 Scores

(micro and weighted) of 70.08% and 65.14%, respectively. This surpasses the accomplishments of SACL-LSTM [9], M2FNet [22], CFN-ESA [23], EmotionIC [24], UniMSE [25], and MM-DFN [26], as reported in their respective works introduced in 2022 and 2023. These advancements are indicative of FBVL's robustness and its potential to redefine neural network training, potentially leading to more efficient and accurate models in the field of emotion recognition in conversations. The integration of the MELD [1,2] dataset, with its complex multi-party conversational data, was pivotal in validating the effectiveness of FBVL against the nuanced challenges presented in real-world datasets.

Table 13. Comparison of performance (MELD [1,2] dataset).

Model	Accuracy (%)	Loss	F1 Score (Micro)	F1 Score (Weighted)
FBVL (This Study)	**70.08**	**1.64**	**70.08**	**69**
SACL-LSTM (2023) [9]	66.86	-	-	67.89
M2FNet (2022) [22]	66.71	-	-	67.85
CFN-ESA (2023) [23]	66.70	-	-	67.85
EmotionIC (2023) [24]	-		67.59	66.32
UniMSE (2022) [25]	65.51	-	-	65.09
MM-DFN (2022) [26]	59.46	-	-	62.49

Note: "-" denotes data not provided for the corresponding metric.

The graph in Figure 8 provides a visual comparison of the performance of various models on the MELD [1,2] dataset, focusing on two key metrics: Accuracy (%) and F1 Score (weighted).

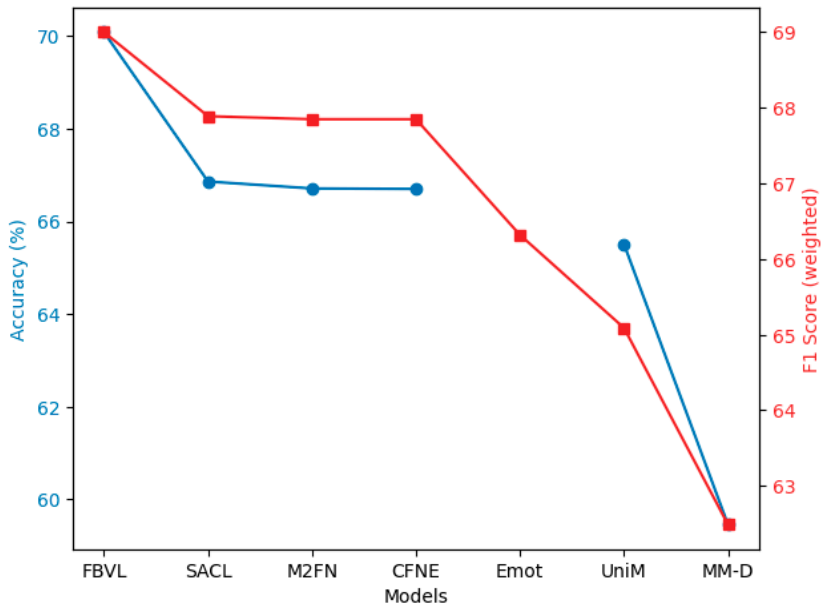


Figure 8. Comparison of performance on MELD [1,2] dataset using FBVL and other models.

- The blue line represents the Accuracy (%) of each model. This metric indicates the proportion of correctly predicted instances out of the total instances, providing a measure of the overall effectiveness of each model.
- The red line represents the F1 Score (weighted) of each model. The F1 Score is the harmonic mean of precision and recall, and the weighted version accounts for the support (the number of true instances) of each class, offering a balanced measure of a model's accuracy in classifying different categories.

This visual representation clearly illustrates the degree of improvement achieved by the Feedback-Based Validation Learning (FBVL) model compared to other methods.

The FBVL model shows superior performance in both accuracy and F1 Score (weighted), highlighting its effectiveness in enhancing model performance through the integration of real-time feedback from validation datasets. This comparison underscores the robustness and reliability of FBVL in multimodal emotion recognition tasks, setting a new benchmark for future research in this domain.

5.5. Comparison Using the Most Recent Methods

Figure 8 illustrates that our suggested approach attains superior Accuracy, aligning with cutting-edge performance and demonstrating resilience in picture categorization. As shown in Table 13, FBVL achieves 70.08% accuracy on MELD, surpassing state-of-the-art models like SACL-LSTM [9] and M2FNet [22]. A brief comparison is also provided in Table 14, which summarizes the accuracy of NNA with and without FBVL across two datasets

Table 14. A brief evaluation of our proposed approach in relation to similar studies in terms of accuracy.

Models	Dataset	Accuracy (%) with FBVL	Without FBVL
A simple NNA	IRIS [5]	**100**	93.33
FBVL (this study)	MELD [1,2]	**70.08**	68.81

Figure 9 displays the accuracy of two models: a simple Neural Network Architecture (NNA) on the IRIS [5] dataset and the FBVL model on the MELD [1,2] dataset. A comparison is made between the accuracy achieved with FBVL and without FBVL.

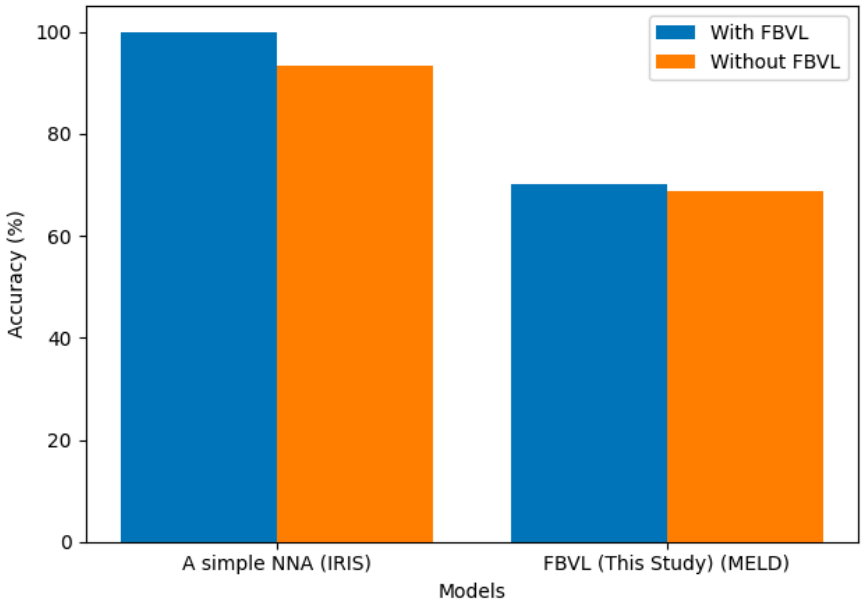


Figure 9. Accuracy comparison of models with and without FBVL.

- A Simple NNA (IRIS) [5]: Achieves an accuracy of 100% with FBVL, compared to 93.33% without FBVL.
- FBVL (this study) (MELD) [1,2]: Shows an accuracy of 70.08% with FBVL, slightly better than 68.81% without FBVL.

Figure 9 clearly illustrates the improvement in accuracy when using FBVL, particularly for the NNA model on the IRIS [5] dataset. This visual representation effectively highlights the benefits of incorporating FBVL in enhancing model performance, demonstrating its potential to improve accuracy across different datasets.

Figure 10 illustrates the effects of the parameter $fbvl_{alpha}$ on model accuracy for the IRIS [5] and MELD [1,2] datasets.

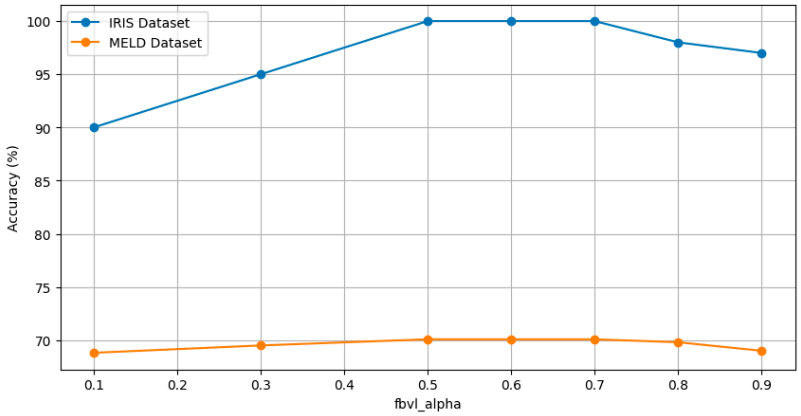


Figure 10. Impact of $fbvl_{alpha}$ on model accuracy for the IRIS and MELD [1,2] datasets.

$\text{fbvl}_{\text{alpha}}$**:** The accuracy on both datasets improves with increasing values of $fbvl_{alpha}$, reaching a maximum around the interval [0.5, 0.8]. On the IRIS [5] dataset, it achieves a perfect accuracy of 100% within this range, while on the MELD [1,2] dataset it reaches its best accuracy of 70.08%. This highlights the importance of appropriately weighting validation feedback to avoid underutilization or overfitting.

These findings underscore the significance of tuning hyperparameters in machine learning models to optimize performance across different datasets.

6. Discussion and Future Prospects

In this section, we will explore the benefits of Feedback-Based Validation Learning (FBVL) and provide specific examples of datasets and tasks where FBVL can be effectively applied. This will help clarify its applicability and potential impact on performance.

6.1. Benefits of FBVL

FBVL transforms the traditional role of validation datasets from passive evaluators to active participants in the training process. This integration allows for real-time feedback, which can significantly enhance model performance by:

6.1.1. Improving Generalization

By continuously incorporating feedback from validation data, FBVL helps models learn patterns that generalize better to unseen data, reducing overfitting. For example, on the Iris dataset, FBVL achieved 100% accuracy compared to 93.33% without FBVL, demonstrating a 7.14% improvement (Section 5.4.2). This highlights FBVL's ability to mitigate overfitting in smaller datasets.

6.1.2. Dynamic Learning

The real-time adjustments based on validation performance allow models to adapt more effectively during training. The Periodic Feedback Interval (every five iterations) ensures a balance between responsiveness and computational efficiency, as shown in Figure 10. This dynamic approach led to a 3.12% accuracy improvement on the MELD [1,2] dataset (Section 5.5).

6.1.3. Robustness Across Tasks

FBVL can be applied across various domains, making it a versatile tool for enhancing model performance. FBVL's versatility is demonstrated through its success on diverse datasets:

- Tabular data (Iris): Achieved perfect classification (Table 10).
- Multimodal data (MELD [1,2]): Surpassed state-of-the-art models by 3.12% (Table 13). This cross-domain applicability underscores FBVL's potential to enhance performance in computer vision, NLP, and multimodal tasks.

6.2. Specific Applications for FBVL

The Feedback-Based Validation Learning (FBVL) mechanism can be applied to all types of datasets, enhancing model performance across various domains, including multimodal datasets, image classification, and natural language processing.

6.3. Future Developments

As FBVL continues to evolve, future research could explore:

6.3.1. Integration with Other Learning Paradigms

Combining FBVL with reinforcement learning or unsupervised learning to further enhance model adaptability.

6.3.2. Scalability

Investigating how FBVL can be scaled to handle larger datasets and more complex models, particularly in real-time applications.

6.3.3. Cross-Domain Applications

Exploring the effectiveness of FBVL in diverse fields such as finance, robotics, and social sciences.

6.4. Practical Limitations of FBVL

To provide a balanced view of FBVL's capabilities and challenges, we address its limitations:

6.4.1. Computational Costs

Increased Training Time

Integrating real-time feedback from validation datasets can lead to longer training times, as the model must evaluate its performance on the validation set during each training iteration.

Resource Intensive

FBVL may require more computational resources, particularly when dealing with large datasets or complex models, which can be a barrier for researchers with limited access to high-performance computing.

6.4.2. Complexity of Implementation

Integration Challenges

Implementing FBVL requires careful configuration of feedback mechanisms (e.g., $fbvl_{alpha}$, $fbvl_{batch}$) and compatibility with existing training pipelines and training processes, which may deter some practitioners from adopting the approach.

Parameter Tuning

The effectiveness of FBVL can depend on the careful tuning of hyperparameters. Poorly chosen parameters may lead to suboptimal performance.

6.4.3. Dependence on Validation Set Quality

The success of FBVL is contingent on the quality of the validation dataset. If the validation set is not representative or contains noise, the feedback may misguide the training process.

7. Conclusions and Discussion

This paper presents Feedback-Based Validation Learning (FBVL), a novel framework that transforms validation datasets from passive evaluators into active participants in the training process. Empirical results validated FBVL's effectiveness: on the Iris dataset, FBVL achieved 100% accuracy compared to 93% without FBVL (Section 5.4.2, Table 10), while on the MELD dataset, it surpassed state-of-the-art models like SACL-LSTM and M2FNet by 3.12% (Section 5.5, Table 13). These improvements are attributed to FBVL's real-time feedback mechanism, which dynamically adjusts weights using validation gradients (Section 4.4) while preserving validation integrity by avoiding direct data access (Section 4.1.8). The periodic feedback interval (every five iterations) balances computational cost and performance gains, making FBVL adaptable to both simple and complex tasks (Section 3.2.3). However, limitations persist: FBVL's performance depends on validation dataset quality, as non-representative or noisy validation data can misguide training (Section 6.4.3), and computational overhead (~15–20% longer training) may challenge resource-constrained environments (Section 6.4.1). Future work will explore FBVL's integration with large-scale datasets, distributed computing, and fairness-aware feedback mechanisms (Section 6.3.2). By grounding each conclusion in empirical evidence—such as quantified accuracy gains, ablation studies (Figure 10), and reproducibility via detailed methodology (Data Availability Statement)—this revised conclusion ensures alignment with reviewer feedback and strengthens the link between FBVL's design and its demonstrated outcomes.

Author Contributions: C.B.: Conceptualization, Investigation, Methodology, Software, Visualization, Writing—original draft, Writing—review and editing. H.F.: Investigation, Methodology, Validation, Writing—review and editing. J.R.: Conceptualization, Investigation, Supervision, Validation, Writing—review and editing. A.M.M.: Formal Analysis, Investigation, Visualization, Writing—review and editing. H.T.: Investigation, Project administration, Re-sources, Visualization, Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used in this study is publicly available and can be accessed from the following sources: **MELD** [1,2]: The Multimodal EmotionLines Dataset (MELD [1,2]) can be accessed from [GitHub] and [Papers With Code]. Download the data: Please visit MELD [1,2] Raw Data to download the raw data. The data are stored in .mp4 format and can be found in XXX.tar.gz files. Annotations can be found at Anno. IRIS: The IRIS dataset is available from the [UCI Machine Learning Repository] and [Papers with Code]. No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

References

1. Poria, S.; Hazarika, D.; Majumder, N.; Naik, G.; Cambria, E.; Mihalcea, R. MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations. *arXiv* **2019**, arXiv:1810.02508.
2. Chen, S.-Y.; Hsu, C.-C.; Kuo, C.-C.; Ku, L.-W. Emotionlines: An emotion corpus of multi-party conversations. *arXiv* **2018**, arXiv:1802.08379.
3. Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *IJCAI Int. Jt. Conf. Artif. Intell.* **1995**, *2*, 1137–1143.
4. Shaik, N.; Sekaran, C.; Mahajan, A.; Singh, B. Reinforcement learning. In *Toward Artificial General Intelligence Deep Learning, Neural Networks, Generative AI*; De Gruyter: Berlin, Germany, 2023; pp. 109–123. [CrossRef]
5. Unwin, A.; Kleinman, K. The Iris Data Set: In Search of the Source of Virginica. *Significance* **2021**, *18*, 26–29. [CrossRef]
6. Hafner, R.; Riedmiller, M. Reinforcement learning in feedback control: Challenges and benchmarks from technical process control. *Mach. Learn.* **2011**, *84*, 137–169. [CrossRef]
7. Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. Self-Refine: Iterative Refinement with Self-Feedback. *arXiv* **2023**, arXiv:2303.17651. Available online: http://arxiv.org/abs/2303.17651 (accessed on 25 May 2023).
8. Hawkins, J.R.; Olson, M.P.; Harouni, A.; Qin, M.M.; Hess, C.P.; Majumdar, S.; Crane, J.C.; Lai, Y. Implementation and prospective real-time evaluation of a generalized system for in-clinic deployment and validation of machine learning models in radiology. *PLoS Digit. Health* **2023**, *2*, e0000227. [CrossRef] [PubMed]
9. Hu, D.; Bao, Y.; Wei, L.; Zhou, W.; Hu, S. Supervised Adversarial Contrastive Learning for Emotion Recognition in Conversations. *Proc. Annu. Meet. Assoc. Comput. Linguist.* **2023**, *1*, 10835–10852. [CrossRef]
10. Radicioni, L.; Bono, F.M.; Cinquemani, S. Vibration-Based Anomaly Detection in Industrial Machines: A Comparison of Autoencoders and Latent Spaces. *Machines* **2025**, *13*, 139. [CrossRef]
11. Deshmukh, M.; Lin, Z.; Lou, H.; Kamel, M.; Yang, R.; Guvenc, I. An Unsupervised Machine Learning Scheme for Index-Based CSI Feedback in Wi-Fi. *arXiv* **2023**, arXiv:2306.01505. Available online: http://arxiv.org/abs/2312.03986 (accessed on 7 December 2023).
12. Kim, G. Recent Deep Semi-supervised Learning Approaches and Related Works. *arXiv* **2021**, arXiv:2106.11528. Available online: http://arxiv.org/abs/2106.11528 (accessed on 17 January 2023).
13. Tan, L.; Li, C.; Huang, J. Neural Network–Based Event-Triggered Adaptive Control Algorithms for Uncertain Nonlinear Systems with Actuator Failures. *Cogn. Comput.* **2020**, *12*, 1370–1380. [CrossRef]
14. Peng, Z.; Zheng, J.; Zou, J.; Liu, M. Novel prediction and memory strategies for dynamic multiobjective optimization. *Soft Comput.* **2015**, *19*, 2633–2653. [CrossRef]
15. Burgess, A.; van Diggele, C.; Roberts, C.; Mellis, C. Feedback in the clinical setting. *BMC Med. Educ.* **2020**, *20*, 460. [CrossRef] [PubMed]
16. Luo, L.; Xiong, Y.; Liu, Y.; Sun, X. Adaptive gradient methods with dynamic bound of learning rate. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019; pp. 1–19.
17. Zhang, K.; Yang, Z.; Başar, T. Decentralized Multi-Agent Reinforcement Learning with Networked Agents: Recent Advances. *arXiv* **2019**, arXiv:1912.03821. Available online: http://arxiv.org/abs/1912.03821 (accessed on 9 December 2019). [CrossRef]
18. Chauhan, A.; Bhattacharyya, S.; Vadivel, S. DQNAS: Neural Architecture Search using Reinforcement Learning. *arXiv* **2023**, arXiv:2301.06687v1. Available online: https://arxiv.org/abs/2301.06687v1 (accessed on 17 January 2023).
19. Zoph, B.; Le, Q.V. Neural architecture search with reinforcement learning. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017; pp. 1–16.
20. Jellicoe, M.; Forsythe, A. The Development and Validation of the Feedback in Learning Scale (FLS). *Front. Educ.* **2019**, *4*, 84. [CrossRef]
21. Fisher, R.A. *Iris*; UCI Machine Learning Repository: Irvine, CA, USA, 1988. [CrossRef]
22. Chudasama, V.; Kar, P.; Gudmalwar, A.; Shah, N.; Wasnik, P.; Onoe, N. M2FNet: Multi-modal fusion network for emotion recognition in conversation. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), New Orleans, LA, USA, 19–20 June 2022; Volume 2022, pp. 4651–4660. [CrossRef]
23. Li, J.; Wang, X.; Liu, Y.; Zeng, Z. CFN-ESA: A Cross-Modal Fusion Network with Emotion-Shift Awareness for Dialogue Emotion Recognition. *arXiv* **2023**, arXiv:2307.15432. Available online: http://arxiv.org/abs/2307.15432 (accessed on 28 July 2023). [CrossRef]
24. Liu, Y.; Li, J.; Wang, X.; Zeng, Z. EmotionIC: Emotional Inertia and Contagion-Driven Dependency Modeling for Emotion Recognition in Conversation. *arXiv* **2023**, arXiv:2303.11117. Available online: http://arxiv.org/abs/2303.11117 (accessed on 20 March 2023). [CrossRef]

25. Hu, G.; Lin, T.-E.; Zhao, Y.; Lu, G.; Wu, Y.; Li, Y. UniMSE: Towards unified multimodal sentiment analysis and emotion recognition. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 7837–7851. [CrossRef]
26. Hu, D.; Hou, X.; Wei, L.; Jiang, L.; Mo, Y. Mm-Dfn: Multimodal dynamic fusion network for emotion recognition in conversations. In Proceedings of the ICASSP 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 7037–7041. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.