CHAPTER 6

WASSERSTEIN DISTANCE DEEP TRANSFER LEARNING ALGORITHM

6.1 Wasserstein Distance

Wasserstein distance in deep transfer learning for air quality prediction has several practical implications that can enhance the model's performance. Wasserstein distance allows the model to better adapt to different regions or environmental conditions, especially when the data distributions differ significantly. In air quality prediction, this is particularly important as data from different regions or time periods can exhibit diverse statistical properties due to varying pollution sources, weather conditions, and geographic factors. This allows the model to minimize the discrepancy between these distributions, thus improving its ability to generalize across different settings. This adaptation is essential for deploying deep learning models trained in one domain to perform well in another with limited data, which is often the case in air quality forecasting where comprehensive data might not be available for all regions.

Furthermore, incorporating Wasserstein distance in deep transfer learning helps the model effectively learn from past data and apply this knowledge to predict future air quality levels in new environments. This becomes particularly valuable when real-time prediction is required in cities or regions with sparse air quality data. The metric aids in adjusting the model's learned features and parameters to align with the target domain's distribution, leading to more accurate predictions of pollutants and environmental conditions. By enabling transfer learning, Wasserstein distance enhances the flexibility and robustness of deep learning models, making them more applicable in real-world air quality monitoring systems.

6.2 Proposed Wasserstein Distance Deep Transfer Learning Algorithm (WD-DTL)

The WD-DTL is provided to reduce the time needed to acquire transfer learning between the source domain and target domain. The architecture of CNN-WD-DTL is shown in Figure 6.1.

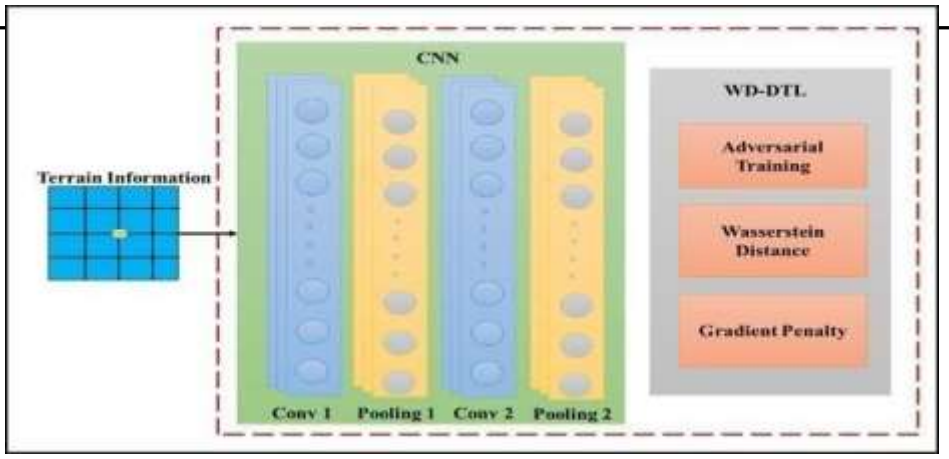

Figure 6.1. Architecture of CNN-WD-DTL

The Wasserstein distance was used to learn features that were consistent in both the source and target domains. Through an adversarial training process, it functioned as the distance measure to learn domain-invariant feature representations. Training often took more than twice as long as a non-transfer learning approach for modeling larger datasets because transfer learning frequently employed a larger base model or a higher-resolution base dataset than the target resolution. Initially, data from the source domain was used to train a basic CNN model. The CNN-based feature extractor was then optimized with the help of a discriminator to minimize the projected empirical Wasserstein distance. By using features that were transferable from a source domain where erroneous labels were known, adversarial learning enabled the diagnosis of a new but relevant problem without requiring a labeled sample.

To examine the weather conditions under which an air quality dataset might be used in other contexts, researchers employ transfer learning. The temperature, wind speed, and direction, average wind speed and average wind direction, relative humidity and altitude are the meteorological data. To get started, relevant data from the source domain is used to train a basic CNN model. Then developed a WD-DTL that can learn attributes that are consistent with both the input and output domains.

The source domain, and its labels are $X^T = \{y^T\}_{=1}^{T}$ of sample $Y^T = \{y^T\}_{i=1}^{n}{}_{J=1}$,where $n^T \in \emptyset$ is the total number of samples in the source domain GT Ds. In the meantime, the target domain $\mathcal{G}^T$Ds defines an unlabeled dataset $X^S = \{y^S\}i=1^{n^S}$ for the most part, the data size of the source domain is substantially bigger than the data size of the target domain, therefore $n^T \gg n^S$ is a reasonable assumption to make while training a CNN classifier. It is also pointed out that the data in both the source and target domains occupy the same feature space (Y^T, $Y^S \in £$), albeit with different marginal distributions ($¢(Y^T)$ G $¢(Y^S)$).

6.2.1 CNN based feature extractor

First, CNN is used to train the data, and the pre-trained model is denoted as Y^T, which was trained on a dataset annotated in the domain of origin. A new feature is computed by applying a filter with dimensions $w \in \mathbb{p}^l$ and a bias $B \in \mathbb{p}$ where l is the filter size. Through the use of the filter $\mathbb{Z}$ and the non-linear activation function$\mathbb{\Gamma}$, an output feature r_i is generated, as shown in the following expression (6.1)

$$r_i = \mathbb{\Gamma}((Æ * v_{kj} + b) \qquad (6.1)$$

which the jth term *is represented by the so*urce *domain and Sub vector* Y^T is represented by the input data, $vk \in 1 \times l$, where " * " is denoted as operation of convolutional parameters. Due to optimal convergence, the gradient decreases in activation functions such as the hyperbolic tangent (tanh) and the Rectified Linear Unit (ReLU). Hence, the feature map can be written as $Q = q1, q2, \ldots, qn,$ where,

M =(f N -t)/Jdw +1 where Jdw $\in \emptyset$ is the stride for convolution, and n is the number of features.

6.2.2 Classification with classifier

The creating a reliable classifier WD-DTL it is necessary to learn invariant features from the source domain data and target domain. It is denoted by symbol for features is $\mathcal{G}^S$. The representation learning strategies then incorporate a discriminator (Ganin et al., 2016) with two fully-connected layers to further narrow the gap from the distribution data. This process requires to set the parameters: batch size (n), and learning time. The rate of learning features time is denoted by β_1 for the domain alignment critic, second learning rate is denoted by the β_2, the feature extractor using CNN is denoted by θ_{ef}, the critic training of data is denoted in the step c_a, the coefficients balance is represented by γ and α. domain alignment critic: θ_{pc}, and predictive: θ_{cp}.

An expression for the gradient distance, g_{grad}, is

$$g_{grad} \leftarrow (\|\nabla_f p(f)\|_2 - 1)^2$$

$$li_{wd} = \frac{1}{n^T} \sum_{b \in X^T} c_p(c_f(X^T)) - \frac{1}{n^S} \sum_{a^S \in X^S} c_p(c_f(X^S)) \quad (6.2)$$

6.2.3 Algorithm of WD-DTL

Input: Target Station T; Set of locations coordinate LC, where, number of candidates n

Output: Set of locations L

Step1: Start the process

Step2: Assume a set of areas: A={a_1,a_2,…a_n}

Step3: Assume set of features: FS = {fs_1,fs_2,…..fs_m}

Step4: Compute area coordinate (AC)

AC_i = (a_i, l_i, m_i)

// considering latitude and longitude of area a_i are l_i and m_i

Step5: Calculate the distance between two locations

$D_{p,q}$ = $dist_{area}$ (AC_p, AC_q)

= $dist_{area}$((a_p,l_p,m_p), (a_q,l_q,m_q))

Step6: ASE receive data on air quality and meteorological conditions, whereas CNN receives data on terrain

// *Apply WD-DTL*

Step7: While θ_{fe}, θ_{cp}, and θ_{pp} has not converged do

Step8: Assume source dataset = $*a^{s}+^{n}{}_{1i=1}$

Step9: Assume target dataset = $*a^{t}+^{n}{}_{1i=1}$

Step10: For i=0,……Ct

Step11: $f^s \leftarrow p_f(Aq^s)$, $f^t \leftarrow p_f(Aq^t)$

Step12: $f \leftarrow \{f^s, f^t, f^r\}$

Step13: $g_{grad} \leftarrow (\|\nabla_f p(f)\|_2 - 1)^2$

Step14: $\theta_{cp} \leftarrow \theta_{cp} + \beta_1 \nabla\theta_{cp} li_w(a^s, b^t) - \tau g_{grad}(f)$

Step15: end for

Step16: $\theta_{pp} \leftarrow \theta_{pp} + \beta_2 \nabla\theta_{pp} l_c(a^s, b^s)$

Step17: $\theta_{fe} \leftarrow \theta_{fe} + \beta_2 \nabla\theta_{fe}[l_c(a^s, b^s) + \delta li_{wd}(Aq^s, Aq^t)]$

Step18: end while

Step19: Find spatial relationships sequence

SRSS = {$D_{1,2}$, $D_{1,3}$, ….. $D_{n-1,n}$}, *//D_{ii}= 0*

// n is the number of spots, and the diagonal values are $D_{ii,}$ are Zero

Step20: Compute Feature Sequence Interval (FSI)

$F(a_i, fs_j, t_{vt,st}) = \{b(a_i, fs_j, t_{vt}), b(a_i, fs_i, t_{vt+1}),\ldots, b(a_i, fs_i, t_{vt})\}$

// $a_i$has feature fs_i that varies from start to finish (vt to st) time (vt<st); and $b(a_i,fs_j,t_x)$ represents the measured value of fs_i at t_x.

Step21: Find distance between feature sequences

$DS_{p,q,tvt,st}= dist_{seq}(F(a_p,fs_{target}, t_{vt,st}), F(a_q,fs_{target},t_{vt,st}))$

Step22: Compute temporal relations sequences (TRSS)

$$TRSS_{t_{vt,}} = *DS_{1,2,t_{vt,st}}, \ldots\ldots \ldots DS_{n-1,n,t_{vt,st}}$$

// The set of x locations with the minimum difference from location i is then chosen as TRSS_cand (i, x).

Step23: Evaluate spatial-temporal relations (STR)

$$STR\_(a_i, x) = SRSS\_cand(a_i, x) \cup TRSS\_cand(a_i, x)$$

Step24: Calculate spatial-temporal predictor (STP)

$$(STR\_(a_i, x)) ,t_{t_l,tq}- = F(a_i, fs_{target}, t_{vtF,stF})$$

Step25: Perform a feed forward pass, computing the activations for layers L_2, L_3, and so on up to the output layer L_{nl}

Step26: End the process.

However, there are several implementation challenges associated with using Wasserstein distance in deep transfer learning for air quality prediction. One significant challenge is the computational expense involved in calculating the Wasserstein distance, especially for high-dimensional data typical of air quality prediction tasks. The direct calculation of this distance can be resource-intensive, particularly when dealing with large datasets from multiple sources.

Another challenge lies in the quality and availability of data. Deep transfer learning relies heavily on sufficient, high-quality data from both the source and target domains to accurately estimate distributions. In the air quality prediction, this is often a limitation, as real-time or region-specific air quality data may be sparse, inconsistent, or noisy. The presence of missing data or outliers can distort the Wasserstein distance computation, leading to suboptimal model performance.

Scalability is also a concern when applying Wasserstein distance in deep transfer learning for air quality prediction. As the number of domains increases, the complexity of computing Wasserstein distance across all these domains grows, making it challenging to maintain the model's scalability. This requires careful optimization of the transfer learning framework and may involve trade-offs between accuracy and computational efficiency

6.3 Experimental Results

The proposed method, WD-DTL is compared with existing methods such as MMSL and TL-SBLSTM using various performance metrics including accuracy, precision, specificity, sensitivity, AUC, Matthew's Correlation Coefficient and Mean Absolute Error Rate.

6.3.1 Accuracy

In Figure 6.2, the accuracy of the proposed method WD-DTL is compared with existing methods, MMSL and TL-SBLSTM. MMSL achieves an accuracy of 96%, TL-SBLSTM achieves 98.3%, while the proposed method WD-DTL demonstrates notably high accuracy. WD-DTL has shown an improvement in accuracy by 1.4% compared to MMSL and by 2.3% compared to TL-SBLSTM in the air quality prediction system.

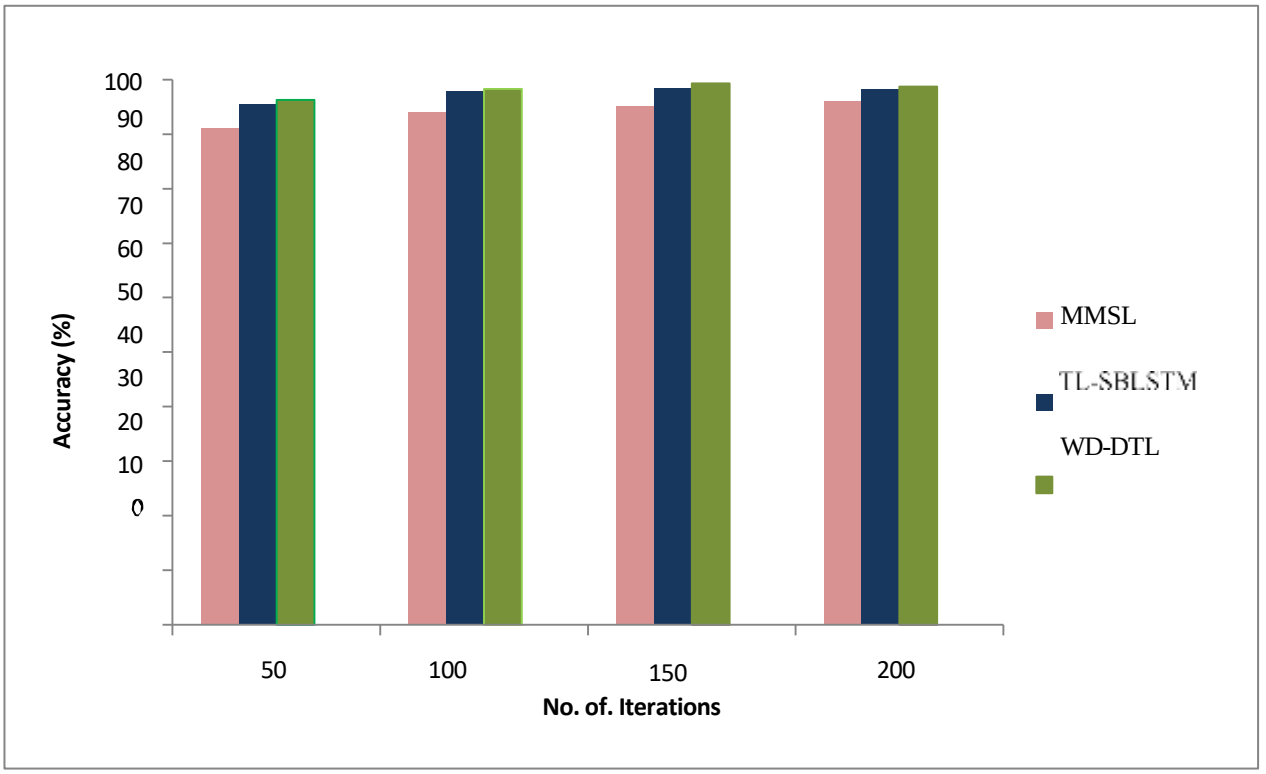

Figure 6.2 Comparison of Accuracy

6.3.2 Precision

In Figure 6.3, the precision of the proposed method WD-DTL is compared with existing methods MMSL and TL-SBLSTM. WD-DTL achieves a precision value of 95.8%, while MMSL and TL-SBLSTM achieve 92% and 94.3%, respectively. WD-DTL shows an improvement in precision by 1.5% compared to MMSL and by 2.3% compared to TL-SBLSTM. MMSL and TL-SBLSTM demonstrate lower precision values, whereas WD-DTL exhibits higher precision.

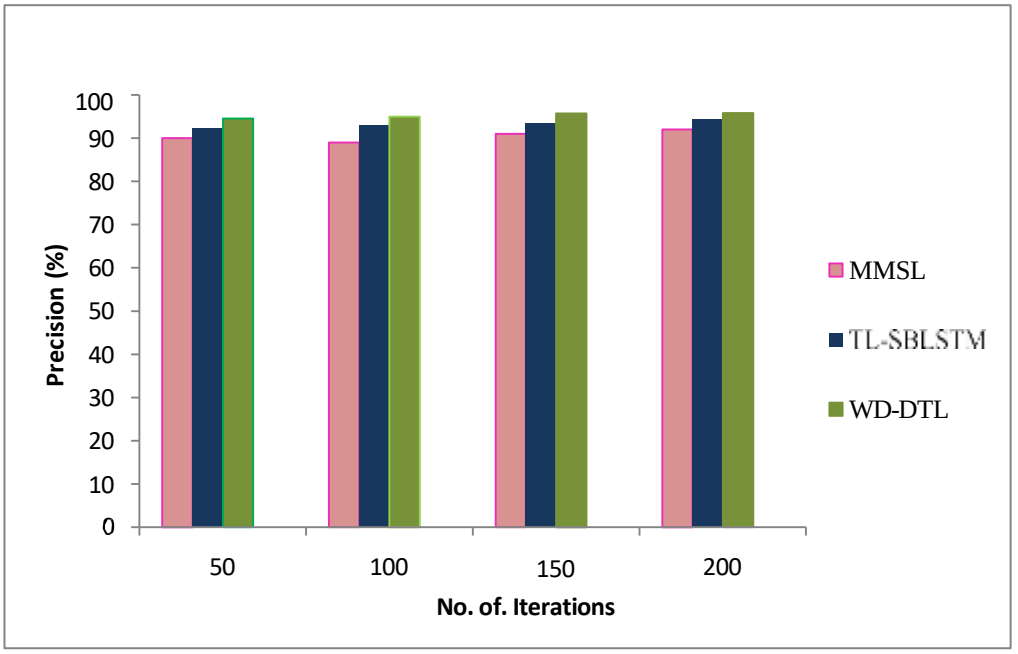

Figure 6.3 Comparison of Precision

6.3.3 Specificity

In Figure 6.4, the specificity of the proposed method WD-DTL is compared with existing methods MMSL and TL-SBLSTM across various iterations. The proposed method WD-DTL achieves a specificity value of 94%, whereas MMSL and TL-SBLSTM achieve 87% and 90.3%, respectively. WD-DTL shows an improvement in specificity by 0.4% compared to MMSL and by 3.3% compared to TL-SBLSTM in the air quality prediction system. MMSL and TL-SBLSTM exhibit lower specificity values, while WD-DTL demonstrates higher specificity.

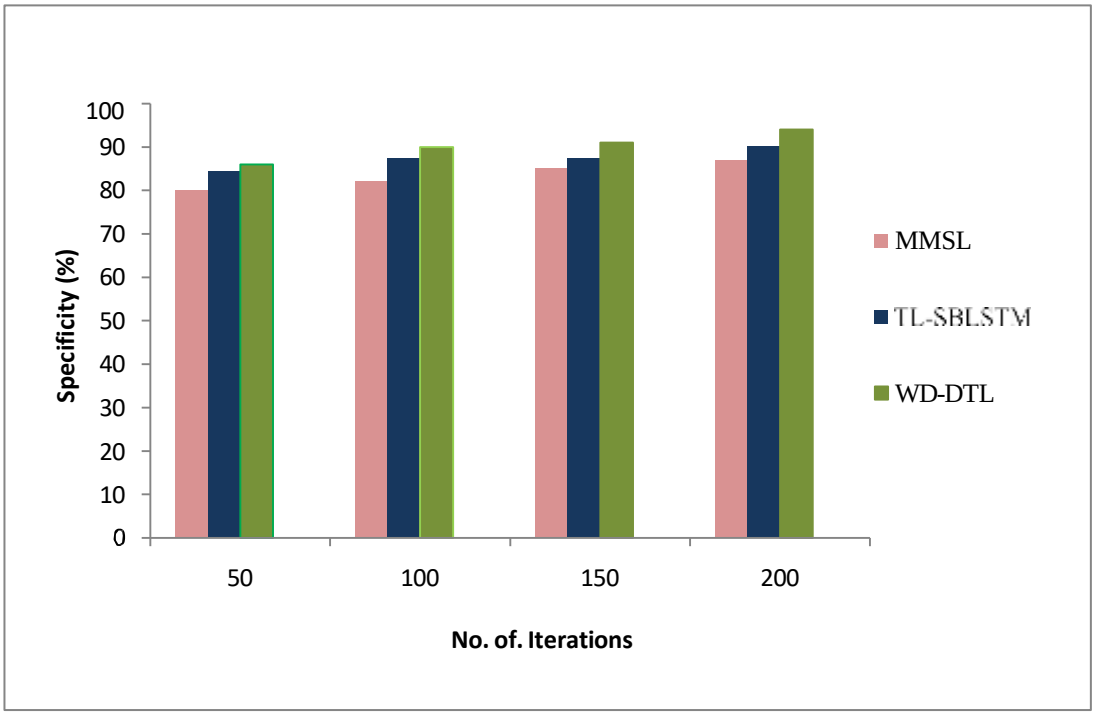

Figure 6.4 Comparison of Specificity

6.3.4 Sensitivity

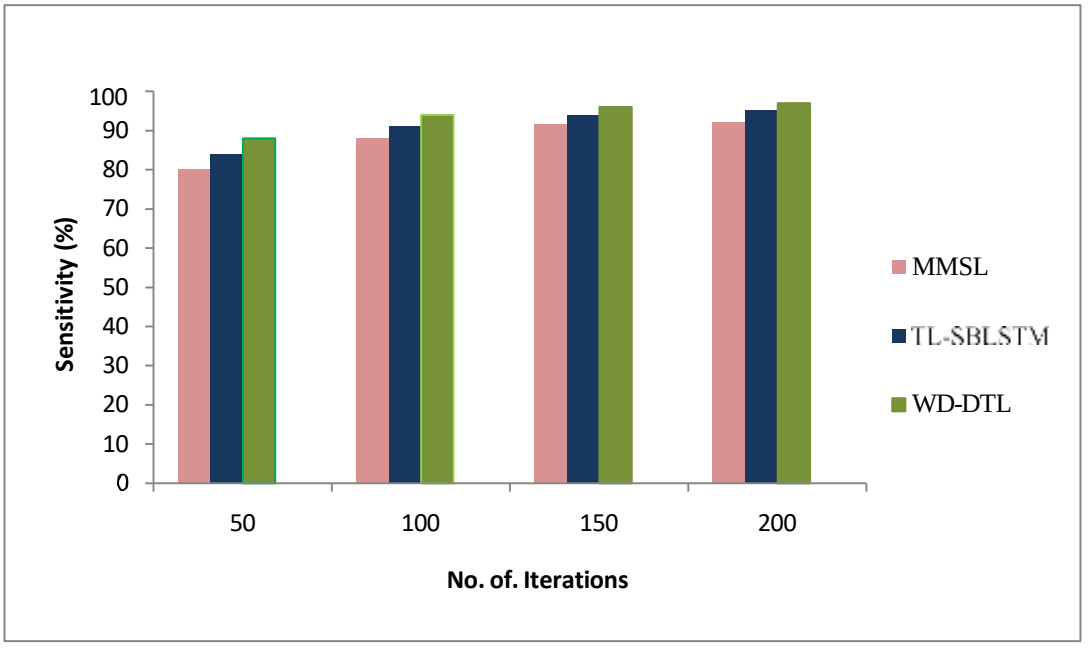

Figure 6.5 Comparison of Sensitivity

Figure 6.5 compares the sensitivity of existing techniques MMSL and TL-SBLSTM with the proposed method WD-DTL. The results show that the sensitivity of MMSL is 92%, TL-SBLSTM is 95%, and WD-DTL achieves an improved sensitivity of 97%. In the air quality prediction system, the sensitivity of WD-DTL has improved by 0.2% compared to MMSL and by 0.3% compared to TL-SBLSTM. MMSL and TL-SBLSTM exhibit lower sensitivity values, while WD-DTL demonstrates higher sensitivity.

6.3.5 Area Under Curve (AUC)

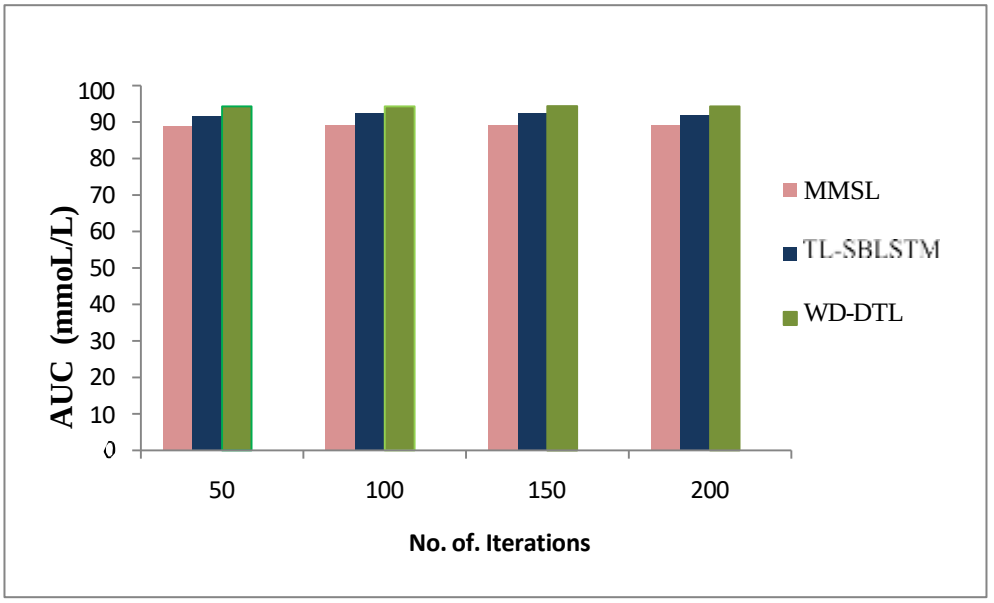

Figure 6.6 Comparison of AUC

Figure 6.6 compares the Area Under Curve (AUC) values of existing techniques MMSL and TL-SBLSTM with the proposed method WD-DTL. From the figure it is evident that AUC of MMSL is 88% and TL-SBLSTM is 91% and for the proposed method it is 94%. The AUC of WD-DTL has improved by 2.28% and 3.3% compared to MMSL and TL-SBLSTM respectively. WD-DTL indicates perfect discrimination, meaning the model perfectly distinguishes between positive and negative classes. The figure indicates that the two existing models have low performance as the models struggle to differentiate between the positive and negative classes.

6.3.6 Matthew's Correlation Coefficient (MCC)

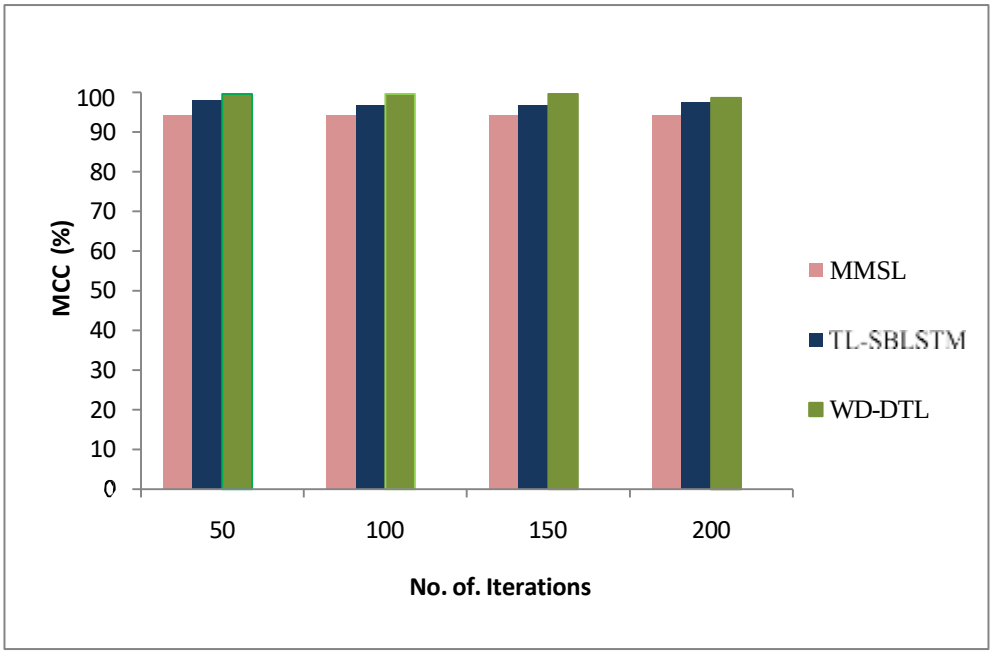

Figure 6.7 Comparison of MCC

In Figure 6.7, the Matthew's Correlation Coefficient (MCC) values of MMSL and TL-SBLSTM are compared with WD-DTL across all different iterations. The MCC of WD-DTL has shown improvement by 1.28% compared to MMSL and by 3.15% compared to TL-SBLSTM in the air quality prediction system. WD-DTL exhibits higher MCC values, while MMSL and TL-SBLSTM demonstrate lower MCC values.

6.3.7 Mean Absolute Error Rate (MAER)

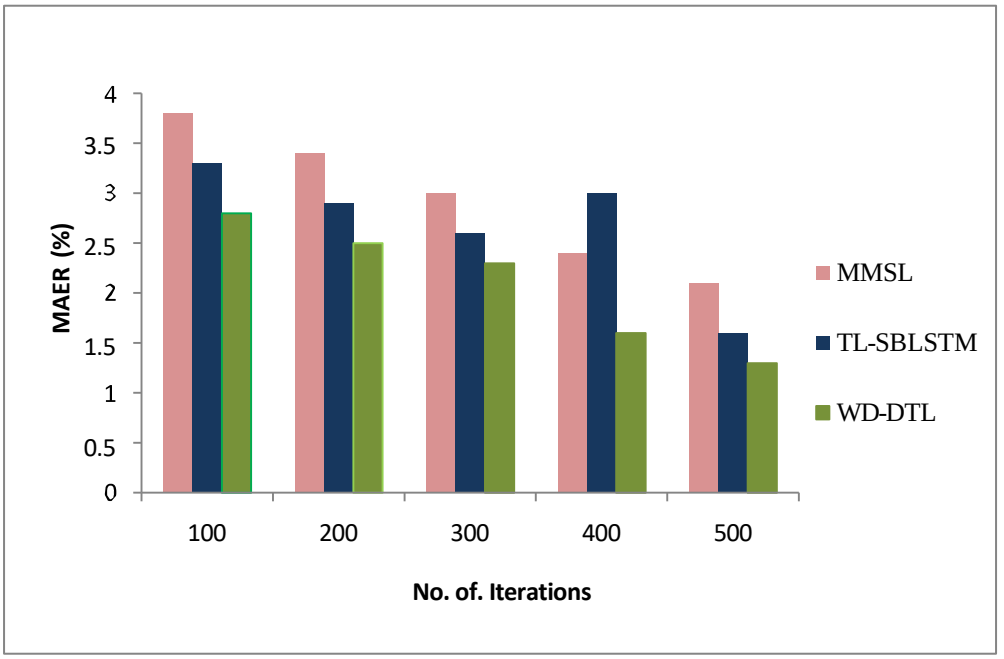

Figure 6.8 Comparison of MAER

Figure 6.8 depicts the percentage of errors made by the MMSL,TL-SBLSTM, and WD-DTL models under 500 training epochs. The results reveal that WD-DTL exhibits a significantly lower error rate compared to MMSL and TL-SBLSTM. The X-axis represents the number of iterations, while the Y-axis represents the error rate. In contrast to MMSL and TL-SBLSTM, which show high error rate values, WD-DTL demonstrates low error rate values.

6.3.8 F-Measure

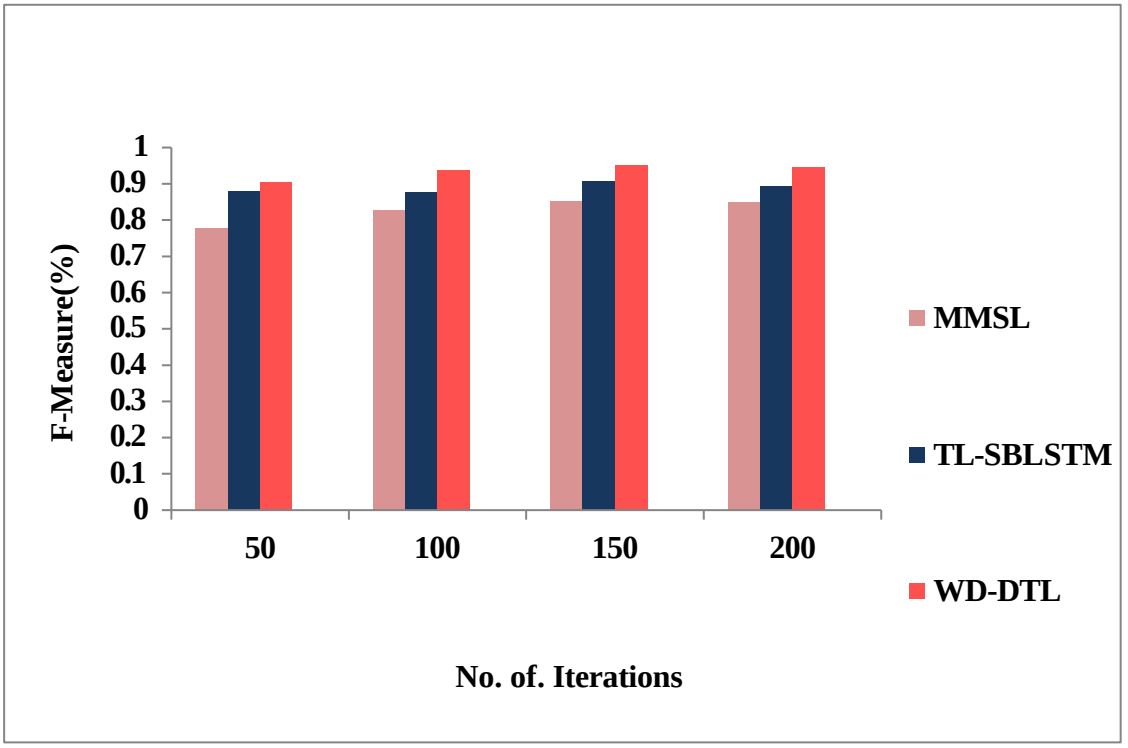

Figure 6.9 Comparison of F-Measure

Figure 6.9 compares the F-Measure of the MMSL, TL-SBLSTM and WD-DTL approaches over a range of iterations, indicating which one provides the most reliable air quality forecasts. When compared to MMSL and TL-SBLSTM, the F-Measure of WD-DTL for air quality prediction is increased by 11.55% and 6.05% values respectively, at an iteration count of 200. Air quality Prediction sensitivity has been measured, and it has been found that the proposed strategy WD-DTL performs better than current methods.