RESEARCH ARTICLE OPEN ACCESS

An enhanced Ensemble Methods for Predicting Concept Drift Labels from a High-Dimensional Data Stream

S. K. Komagal Yallini *, Dr. N. Mahendiran**
**(Research Scholar, Department of Computer Science, Sri Ramakrishna College of Arts and Science,Coimbatore – 641006, Tamil Nadu, India)*
*** (Assistant Professor, Department of Computer Science, Sri Ramakrishna College of Arts and Science, Coimbatore – 641006, Tamil Nadu, India)*

ABSTRACT
In information science, Multi-Label Learning (MLL) has emerged as a method for identifying samples according to certain features related to a number of tags. When an information flow introduces new perspectives, advanced training necessitates the classification of characteristics using New Labels (NLs). To address this challenge, a classification algorithm called MuEMNL-Ensemble Neural Network (ENN) was developed for MLL with Emerging Multiple NLs (MuEMNL). This algorithm effectively handles large amounts of data and resolves concept drift issues. However, the memory usage required to preserve previously learned information in the ENN was found to be excessive. This article presents a Generative Adversarial Network (GAN) model for classifying data streams using the MuEMNL-ENN classifier as a solution. The GAN model eliminates the need to store previously learned information, resulting in reduced memory usage. By including the GAN model, the GAN-MuEMNL-ENN model solves the problem of not having enough historical data and prevents data streams from forgetting everything in the MLL process. This model is capable of performing online MLL using the MuEMNL-ENN classifier and adapting to new data classes without losing previously learned information. The GAN consists of generator and discriminator networks, with the generator model regenerating historical training data that cannot be retained. This model eliminates the need to store and reuse previous observations, making it advantageous for applications dealing with large datasets. Extensive experiments have proven that the GAN-MuEMNL-ENN model outperforms existing models for MLL on a variety of multi-label datasets.
Keywords-Data streams, GAN, MuEMNL-ENN, Multi-label learning, Online learning

Date of Submission: 04-09-2024 Date of acceptance: 17-09-2024

I. Introduction

The most popular idea in artificial intelligence today is traditional supervised learning, in which each characteristic is symbolized by a unique vector related to a particular label. However, a distinguishing characteristic may include many labels in different situations [1–2]. For instance, a photo may capture multiple labels, a transcript may encompass multiple topics, and a soundtrack may belong to more than one category [3]. MLL has gained popularity as a learning theory for handling this type of information [4]. In MLL, every item associated with a collection of labels is characterized by a specific attribute, unlike in standard supervised learning. The method ensures the correct label collections for unsupplied attributes [5]. Historically, MLL has played a significant role in addressing a variety of issues, including enhancing the efficiency of multimedia information, and has increasingly expanded into specific machine learning domains. Previous research on MLL focused on multi-label text categorization using a pre-specified label set [6]. However, in many practical scenarios, a flexible instance has many current tags, possibly the best tags in the predicted information flow pattern [7].

A training technique can rearrange and reconfigure a typical system into different configurations in a complex world. This method must be able to replicate an existing framework in the MLL model for new features while also providing updated classification techniques for each NL. Excluding real-world learning data from the advanced MLL eliminates Ground Truth (GT) for tags at every stage of the data stream [8]. As such, NL discovery and simulation were the main challenges. It was difficult to determine the traits of an NL. It was challenging to differentiate characteristics with recognized labels from those with NLs due to the absence of new labels in previous data, which typically co-existed with a few helpful labels. Because of an inaccurate categorization, the frequency of failure rose as the amount of NLs in the information source increased.

Therefore, creating suitable frameworks to enhance the performance of categorization in a data stream was a challenging task. To address this issue, researchers introduced several MLL algorithms with novel approaches to identify correlations between labeled and unlabeled features [9]. The MuENL [10] was suggested to identify and classify features with Emerging NLs (ENLs) in light of these factors. The MuENL method has several key phases, including: Sorting the features linked to recently discovered labels, determining whether an NL exists, and creating a new classification scheme for all NLs that cooperate with the predictor for the tags that have been identified. In addition, MuENLHD was adjusted to handle high-dimensional sparse information by utilizing a kernel Principal Component Analysis (PCA) to reduce size. On the other hand, this method could only address a certain NL at a particular stage. On the other hand, this method might treat the quantity of NLs as a separate NL in cases where the test set includes many NLs in a particular step. As a result, performance suffers.

So, the MuEMNL and MuEMNLHD methodologies have been recommended [11] to resolve the challenges in a flexible scenario with a large number of natural languages. To adapt to this dynamic situation, the NL set was divided into multiple new CLs. The approach involves four main steps:

i) Identifying the MuEMNLforest and MuEMNLHD groups by the OPTICS approach;
ii) Creating a new outlier identification using both primary and test data streams;
iii) Constructing a linear classifier to reduce the pairwise tag categorization loss on the group of tags; and
iv) Using a categorization updating approach to incorporate NLs and build a powerful categorizer.

However, maintaining the current approach is crucial for addressing concept drift issues, particularly when dealing with large volumes of information entering at high rates and limited resources.

Because of this, a flexible group training method was introduced [12] that creates a MuEMNL-Ensemble Neural Network (ENN) instead of a random forest categorizer. This can handle huge amounts of data and fix issues with concept drift. It details the number of independent NNs using a constructive-pruning strategy, the size of the ensemble, the hidden nodes, and the learning examples. Pairwise and non-pairwise diversity metrics were also tested while building the ENN for effective training with all of the learning examples to address concept drifts. Furthermore, NNs were kept both diverse and accurate at the same time. Nonetheless, the memory usage for preserving previously learned information in the ENN required more memory.

Hence, this article introduces a Generative Adversarial Network (GAN) model for classifying data streams using the MuEMNL-ENN classifier. The model eliminates the need to store previously learned information, resulting in reduced memory usage. By incorporating the GAN model, the GAN-MuEMNL-ENN model addresses the lack of historical data and prevents catastrophic forgetting in the MLL process for data streams. It introduces a model capable of performing online MLL using the MuEMNL-ENN classifier and adapting to new data classes without losing previously learned information. The GAN consists of generator and discriminator networks, with the generator model regenerating historical training data that cannot be retained. This model eliminates the need to store and reuse previous observations, making it advantageous for applications dealing with large datasets.

The structure of the following manuscript is as follows: Section II examines earlier MLL studies in diverse fields. Section III describes the GAN-MuEMNL-ENN, and Section IV demonstrates its efficacy. Section V précises the study and discusses forthcoming developments.

II. Related Works

This section reviews various MLL algorithms developed by earlier researchers for different applications. Mishra and Singh [13] introduced the Feature Construction and Smote-based Imbalance (FCSMI) technique for multi-label corpora. Initially, the average imbalance ratio was used to identify minority labels. Then, it computed the distances of every sample from all minority samples and utilized SMOTE to equalize the proportion of minority and majority samples. Eventually, the corpus with a minimal imbalance rate was used for classifier training. Using large-scale datasets incurs a high computational cost, and the model's performance on new data was ineffective.

Dahiya et al. [14] created the Deep Extreme Multi-Label Learning (DeepXML) model for learning frameworks, which assign the most suitable subgroup of tags from a massive tag group to a given data element. A feature architecture was used to map the data onto a dense D-dimensional representation, followed by training intermediate feature representations using a surrogate objective. Also, sub-linear search and negative sampling schemes were applied. Transfer learning was also utilized to obtain a final feature representation. However, it should be noted that the model has a high training time and memory usage.

Rezaei-Ravari et al. [15] introduced two regularized MLL methods: Regularized MLL through Feature Manifold (RMLFM) and Regularized MLL via Dual-Manifold (RMLDM). RMLFM incorporates an attribute manifold normalization factor to maintain local characteristics, while RMLDM uses attribute and information manifold normalizations to preserve local characteristics of both information and attributes. Two iterative schemes with global conjugate gradient mechanisms were developed to calculate fitness values for these methods, which may require higher memory usage for large-scale datasets.

Li et al. [16] investigated the class-reliant shift matrix's identifiability in noisy MLL. They proposed a new predictor that leverages label correlations. They assessed the presence of noisy tags to establish noisy tag relationships, and then used a sample selection process to identify noiseless tag relationships. By integrating the inferred tag relationships, a shift matrix was derived through a straightforward bilinear decomposition approach. But they did not address the memory utilization problem, which affects the effectiveness of this predictor for MLL.

Fu et al. [17] introduced a kernel factor and manifold normalization to capture label correlations iteratively. They also used label correlations and local label data to predict unobserved samples. However, this approach is computationally intensive for large-scale datasets.

Liu et al. [18] examined semantic labels and their correlation with texts using an attention strategy toselect relevant features. First, a graph convolutional network was used to model high-label correlation. Then, the Label-Aware Attention and Semantic Dependency (LAA-SD) scheme was adopted to improve text feature representation and address label semantic dependency for text MLL. However, this approach did not effectively handle new labels and struggled with class imbalance problem.

A multi- and weak-label learning framework [19] was applied to enhance label semantic space through unified label relationship. They utilized label information dependability, feature-label reliance, and tag relationships to improve semantic views. Additionally, they employed l_2,1-norm to address the issue of absent tag space noise.

An ELSMML, a label relationship and MLL [20] was developed to define higher-level tag relationships. Also, multi-view learning and dimensionality minimization were employed to uncover higher-order latent semantic label and latent feature information. They utilized an enhanced proximal gradient mechanism to optimize the model parameters and obtain the predictive classifier.

Huang et al. [21] introduced the Multi-Graph MLL with Novel and Missing Labels (MGMLNM) scheme, which utilized specific graph kernels to maintain structure information and create an effective graph representation. A unified objective function with projection-relationship and bag-reliant normalizations was employed to regulate new and absent tags concurrently and extract complex correlations between bag and graph labels in multi-graph data. However, the approach did not address concept drift and memory utilization, impacting its efficiency for large-scale datasets.

2.1 RESEARCH GAP

Numerous algorithms have been created for MLL, but it is essential to address memory-related issues for effective training and deployment of models. One of the primary challenges in online MLL on data streams is the potentially large size of the dataset, which presents storage and memory management issues during training to retain previously learned information. Traditional learning approaches are notadept at handling rapidly increasing amounts of data in real time, as well as factors such as changing distribution of streaming data over time, limited computational time, and memory. To tackle this issue, this study proposes the use of the GAN model with ensemble classification for online MLL on data streams that do not store previously learned information.

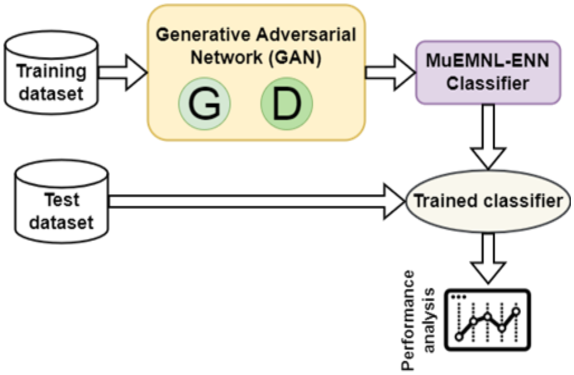

Figure. 1 Conceptual Diagram of this Work

III. Proposed Methodology

The GAN-MuEMNL-ENN model is briefly explained here. Fig. 1 illustrates the conceptual diagram of this work.

A model is designed to illustrate the process of online learning, representing the continuous arrival of data with distinct classes coming in separately. The proposed framework is depicted in Fig. 2.

Consider$S = \{S_i | i = 1, \dots, N\}$is the splitting of real data into N different classes. In this proposed online MLL model,initially consider the first incoming classS_1 from $S$and utilize it to train a generatorG_1, which is capable of representing this data. Once $G_1$is trained, store it and remove S_1.

After that, trainG_2 on the data from S_2, while simultaneously training a neural network classifier C_1^2 in an ensemble. This classifier is fed with samples fromS_1^*, which is synthetic data generated by G_1, as well as newly arriving real data fromS_2. After this process, the data from S_2 isdiscarded. This process is repeated for all classes in S, generating equal batches of data from each previously trained generator. When a new class is added, a node is also added to the output layer of the classifier and its connections are initialized with the previous layer, as shown in Fig. 3.

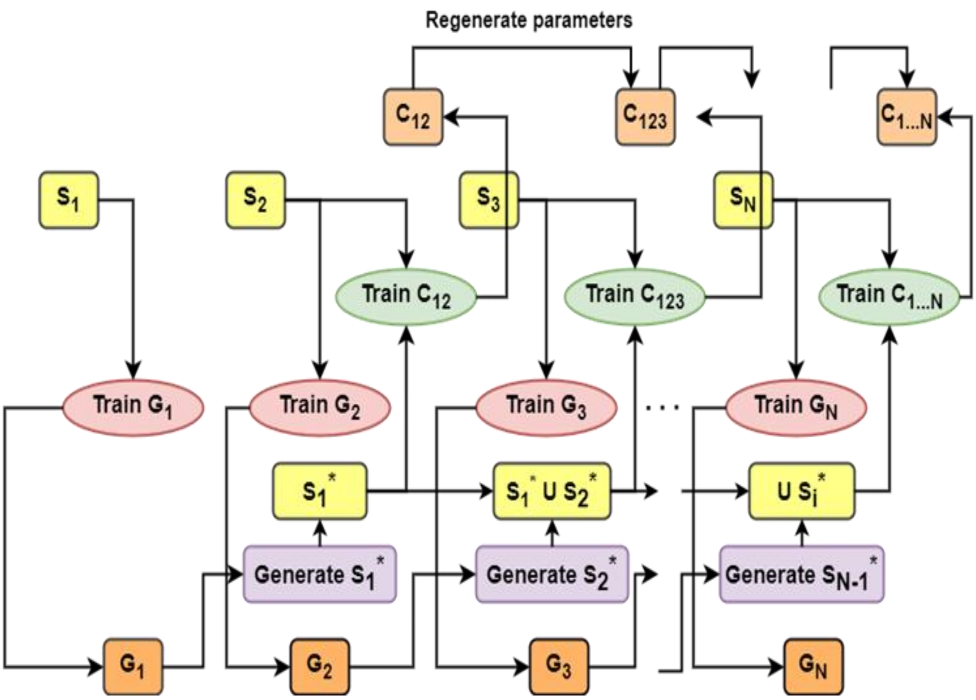

Figure. 2 Schematic Representation of GAN-MuEMNL-ENN Model f or Online ML

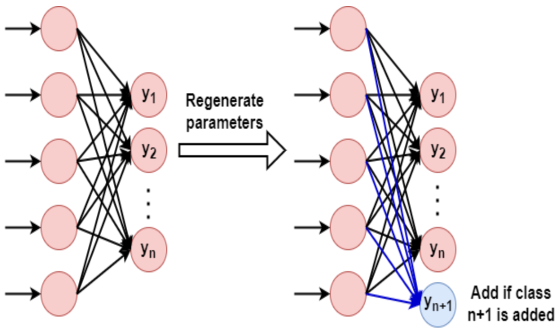

Figure 3 Adding a Node to the Output Layer of Neural Network in Ensemble Model and Initializing the Connections with the Previous Layer in Online MLL if New Data Class occurs in the stream

The remaining network weights are copied from the previous state. The pseudocode of this model is presented in Algorithm 1.

Algorithm 1: GAN-MuEMNL-ENN Classification Model for Online MLL

Require: a data stream$S = \bigcup_{i=1}^{\infty} S_i$ with class number$i$, n number of previously learned classes, a generative model G_i for class $i$and neural network classifierC_1^nin ensemble for data from $\bigcup_{i=1}^{\infty} S_i$

1. **Begin**
2. G_1 ⇐initialize model;
3. $n \leftarrow 1$;
4. $\boldsymbol{while}(receiving\ samples\ from\ S)$
5. d ⇐ get batch from S_j, where j is the current class;
6. $\boldsymbol{if}(j = n + 1)$
7. $n \leftarrow n + 1$;
8. G_n, C_1^n ⇐initialize models;
9. $\boldsymbol{if}(n > 2)$
10. C_1^n ⇐ regenerate parameters from C_1^{n-1};
11. $\boldsymbol{end\ if}$
12. $\boldsymbol{end\ if}$
13. $d^* \leftarrow \bigcup_{i=1\ldots n}^{i \neq j} d_i^*$ create synthetic data from $\{G_i\}$;
14. C_1^n ⇐ train with $d \cup d^*$;
15. G_j ⇐ train with d;
16. $\boldsymbol{end\ while}$
17. **End**

3.1 Design of Deep Convolutional Generative Adversarial Network (DCGAN) model

This study uses the DCGAN model for the generative network, which includes DCGAN generator and discriminator networks constructed with convolutional layers and topological constraints to ensure improved convergence. Fig. 4 shows that the DCGAN model [22] consists of one convolutional layer and two fully connected linear layers. A Rectified Linear Unit (ReLU) activation function is applied to all layers, except the final one. Batch regularization and dropout are employed during training on each layer, except the output layer. Stochastic Gradient Descent (SGD) with Adam is used for model parameter tuning.

Thus, in the GAN-MuEMNL-ENN model, real data is applied to the model on a class-by-class basis. As new classes of data become available, a new generator is trained to model each class. At the same time, a classifier is trained on the generated data from previously learned classes and the real data from the new class as it comes in from the data stream.

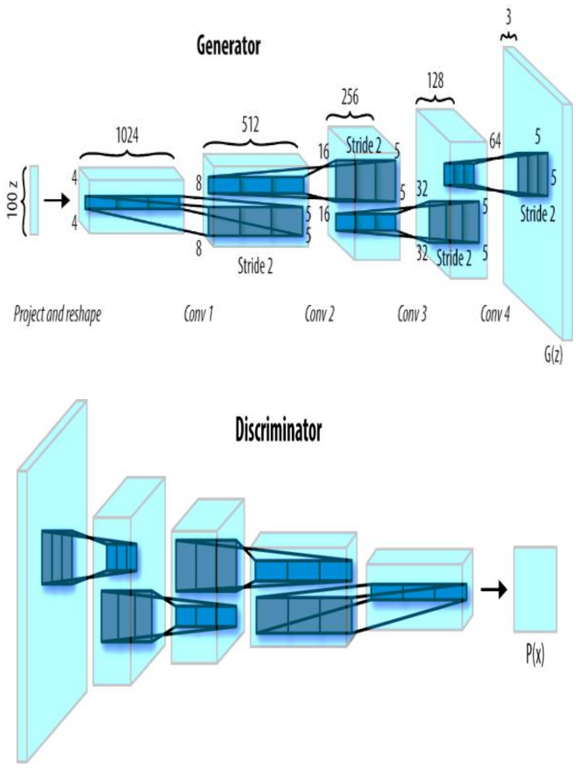

Figure. 4 Architecture of DCGAN Model

IV. Experimental Results

The GAN-MuEMNL-ENN model is tested in MATLAB 2019b using various separate multi-label benchmark datasets (birds, CAL500, emotions, Enron, yeast, and 20Newsgroup) [23]. Details about these datasets can be found in [11]. The statistical assessment included Average Precision (AP), F1 score, and micro-F1 measures. In the test set(r_n, Y_n), $h(r_n)$denotes the group of estimated tags forn^{th} instance, $f(r_n, y)$is the certainty that r_n fits to the tag y.

AP: It represents the mean fraction of +ve tags that are scored greater than a given +ve tag.

$$AP = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i|}\sum_{y\in Y}\frac{|l_p|}{sort_{f(r_n,y)}} \quad (1)$$

where $$l_p = \left\{y'|sort_{f(r_n,y')} \le sort_{f(r_n,y)}, y' \in Y_i\right\} \quad (2)$$

In Eqns. (1) and (2), Y_i is the set of +ve tags, n is the overall test samples, l_p is the set of predicted +ve tags that are ranked lower than y for r_n.

F1 score: It is determined as:

$$F1\ score = \frac{1}{n}\sum_{i=1}^{n}\frac{2\times Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (3)$$

Micro-F1: It is calculated by

$$Micro\ F1 = \sum_{i=1}^{n}\frac{2\times Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (4)$$

Accuracy: It measures the efficacy of MuEMNL-ENN to appropriately forecast tag of novel information.

$$Accuracy = \frac{True\ Positive\ (TP) + True\ Negative\ (TN)}{TP + TN + False\ Positive\ (FP) + False\ Negative\ (FN)} \quad (5)$$

In Eq. (5), TP stands for the total number of +ve examples considered as +ve, TN stands for the total number of -ve examples considered as -ve, FP stands for the total number of +ve examples considered as -ve, and FN stands for the total number of -ve examples considered as +ve.

Hamming loss: It is the proportion of information with incorrectly estimated or omitted labels.

One-error: It is the proportion of information whose top-ranked forecasted tag is not in the GT tag set.

Coverage: This measures the mean amount of steps required to navigate through an example's ordered tag set to encompass all associated tags.

Ranking loss: It is the mean fraction of misordered tag groups, i.e. a suitable tag of an information is organized above its suitable tag.

4.1 Performance Analysis of GAN MuEMNL-ENN on Low-dimensional Datasets

The effectiveness of GAN-MuEMNL-ENN has been assessed by comparing it to the existing models MuEMNL-ENN [12], DeepXML [14], ELSMML [20], and MGMLNM [21] on 5 different low-dimensional datasets.

In Fig. 5, the AP of various MLL models is compared. The GAN-MuEMNL-ENN demonstrates a higher AP compared to other models. For example, when using the Yeast dataset, the GAN-MuEMNL-ENN increases the AP by 45.83%, 32.08%, 18.64%, and 9.72% compared to the ELSMML, MGMLNM, DeepXML, and MuEMNL-ENN, respectively.

Fig. 6 shows the F1 score values of various MLL models. The GAN-MuEMNL-ENN achieves a higher F1 score compared to others. For example, when using the CAL500 dataset, the GAN-MuEMNL-ENN increases the F1 score by 28.81%, 20.63%, 13.43%, and 8.42% compared to the

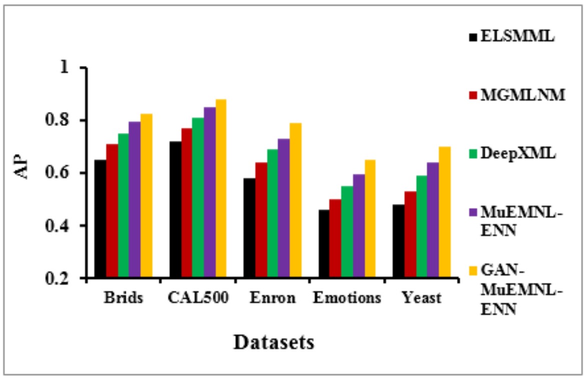

Figure. 5 AP vs. Datasets

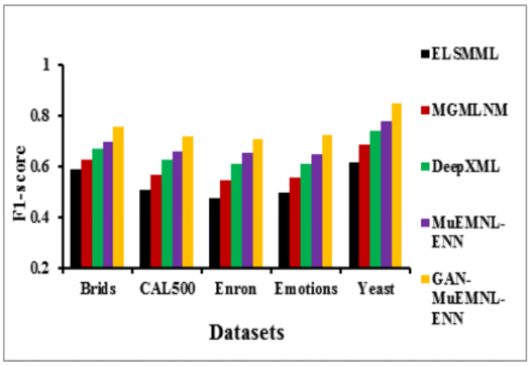

Figure. 6 F1 score vs. Datasets

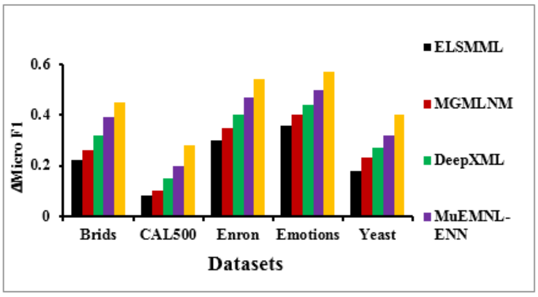

Figure. 7 **Δ**Micro F1 vs. Datasets

ELSMML, MGMLNM, DeepXML, and MuEMNL-ENN, respectively.

Fig. 7 demonstrates the ΔMicro F1 results of the various MLL models. It is worth noting that the GAN-MuEMNL-ENN shows an increase in ΔMicro F1 compared to the others. For example, when using the Enron dataset, the GAN-MuEMNL-ENN increases the ΔMicro F1 value by 80%, 54.29%, 35%, and 14.89% compared to the ELSMML, MGMLNM, DeepXML, and MuEMNL-ENN, respectively.

Fig. 8- 12 illustrates the hamming loss, One-error, ranking loss, coverage, and accuracy for the different MLL models. For the Birds dataset, the hamming loss of GAN-MuEMNL-ENN is minimized by 63.64%, 60%, 55.56%, and 42.86%

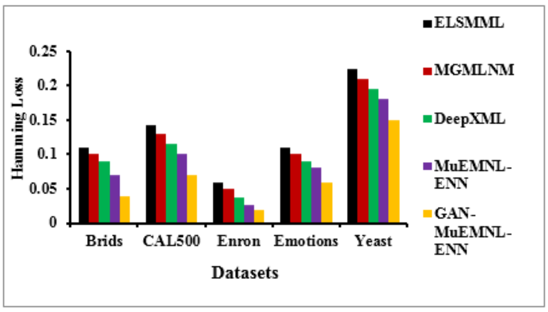

Figure. 8 Comparison of Hamming Loss

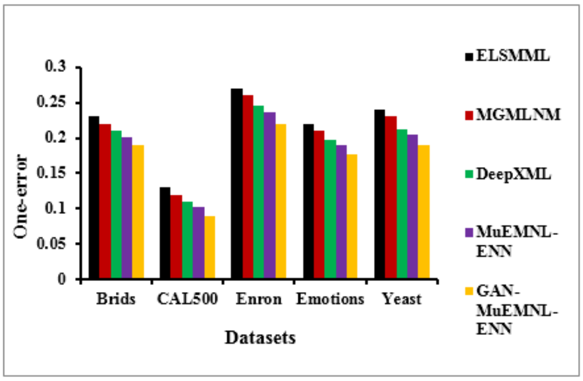

Figure. 9Comparison of One-error

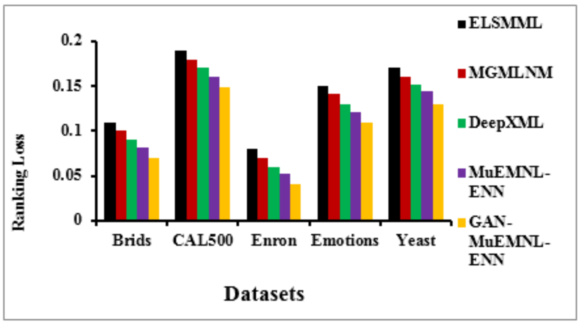

Figure. 10Comparison of Ranking Loss

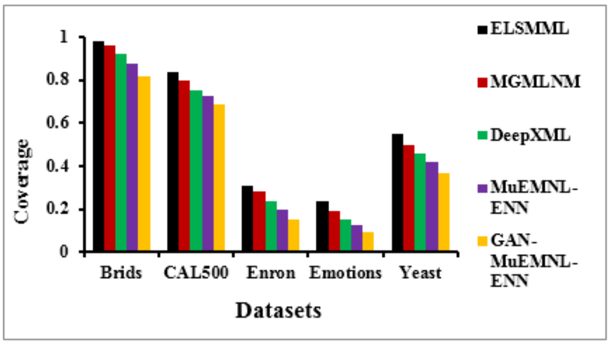

Figure. 11Comparison of Coverage

compared to ELSMML, MGMLNM, DeepXML, and MuEMNL-ENN, respectively. The one-error is decreased by 17.39%, 13.64%, 9.52%, and 5.47% compared to ELSMML,MGMLNM, DeepXML, and MuEMNL-ENN, respectively. The ranking loss is reduced by 36.36%, 30%, 22.22%, and 14.63% compared to ELSMML, MGMLNM, DeepXML, and MuEMNL-ENN, respectively.

In the emotions dataset, the coverage of GAN-MuEMNL-ENN is minimized by 62.5%, 52.63%, 40%, and 29.13% compared to ELSMML, MGMLNM, DeepXML, and MuEMNL-ENN, correspondingly. The accuracy of GAN-MuEMNL-

ENN is boosted by 18.42%, 12.5%, 7.14%, and 3.69% compared to ELSMML, MGMLNM, DeepXML, and MuEMNL-ENN, correspondingly. Hence, it is evident that GAN-

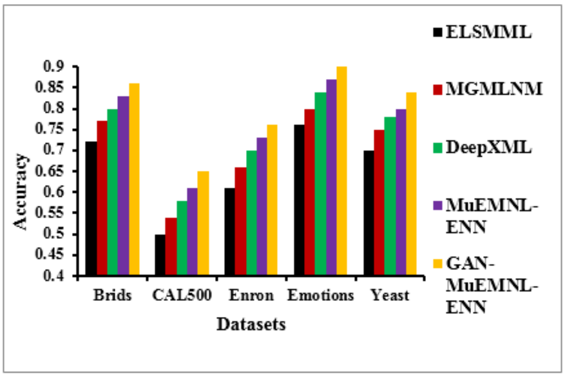

Figure. 12 Comparison of Accuracy

MuEMNL-ENN outperforms the others on low-dimensional corpora across several measures.

4.2 Performance Analysis of GAN-MuEMNL-ENN on High-Dimensional Datasets

The effectiveness of GAN-MuEMNLHD (High-Dimensional)-ENN is evaluated by comparing it to the existing models MuEMNLHDForest [11], MuEMNLHD-ENN [12], and RMLDM [15] on the 20Newsgroup dataset.

Fig. 13 shows the average precision, F1-score, **Δ**Micro F1, and accuracy of different MLL models. GAN-MuEMNLHD-ENN has a 24.59% higher mean precision compared to RMLDM, a 17.65% higher mean precision compared to MuEMNLHDForest, and a 6.89% higher mean precision compared to MuEMNLHD-ENN. The F1-score of GAN-MuEMNLHD-ENN is 28.07% higher than RMLDM, 15.69% higher than MuEMNLHDForest, and 5.04% higher than MuEMNLHD-ENN. The **Δ**Micro F1 of GAN-MuEMNLHD-ENN is 65.22% higher than RMLDM, 42.32% higher than MuEMNLHDForest

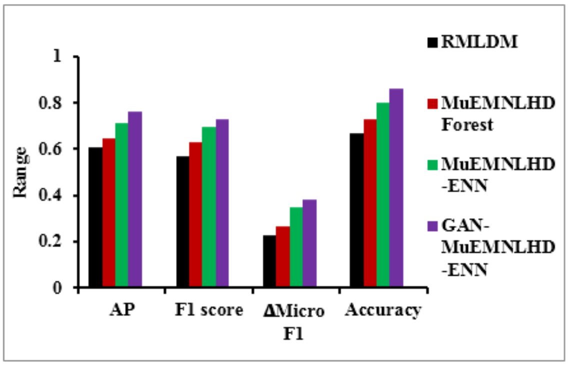

Figure. 13 Comparison of Precision, F1-score, and Accuracy for Different MLL Models on 20Newsgroup Dataset

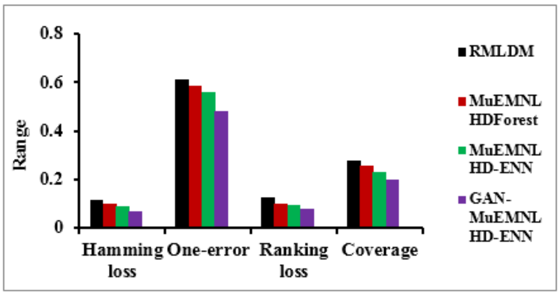

Figure. 14 Comparison of Hamming Loss, One-Error, Ranking Loss, and Coverage for Different MLL Models on 20Newsgroup Dataset and 8.88% higher than MuEMNLHD-ENN. The accuracy of GAN-MuEMNLHD-ENN is 28.36% higher than RMLDM, 17.97% higher thanMuEMNLHDForest, and 7.1% higher than MuEMNLHD-ENN.

Fig. 14 compares the hamming loss, one-error, ranking loss, and coverage of various MLL models. The hamming loss of GAN-MuEMNLHD-ENN is40.68%, 29.58%, and 22.31% lower than RMLDM, MuEMNLHDForest, and MuEMNLHD-ENN, correspondingly. The one error of GAN-MuEMNLHD-ENN is minimized by21.31%, 17.95%, and 14.29% compared to the RMLDM, MuEMNLHDForest, and MuEMNLHD-ENN, respectively.

The ranking loss of GAN-MuEMNLHD-ENN lessened by 36.51%, 21.57%,and 17.27% compared to the RMLDM,MuEMNLHDForestand MuEMNLHD-ENN, correspondingly. The coverage of GAN-MuEMNLHD-ENN diminished by 28.57%, 22.48%, and 13.04% compared to the RMLDM, MuEMNLHDForest, and MuEMNLHD-ENN, respectively. Hence, GAN-MuEMNLHD-

ENN outperforms conventional MLL models on a high-dimensional corpus.

V. Conclusion

In this work, the GAN-MuEMNL-ENN model has been presented that represents a significant advancement in the field of online MLL for data streams. By eliminating the need to store previously learned information, the GAN-MuEMNL-ENN model effectively addresses the memory usage issue. This innovation not only tackles the challenge of insufficient historical data but also prevents catastrophic forgetting in the MLL process. The model's capacity for online learning and adaptation to new data classes without compromising retained knowledge positions it as a powerful solution for dynamic datasets. The DCGAN, consisting of generator and discriminator networks, enables the regeneration of historical training data that cannot be stored, offering a new approach to handling large datasets without sacrificing efficiency. Through extensive experimentation, the GAN-MuEMNL-ENN model has demonstrated superior performance compared to existing models across diverse multi-label datasets, confirming its effectiveness and potential for advancing real-world applications in the field of machine learning.

REFERENCES

[1]. S. Cohen, The basics of machine learning: strategies and techniques, in Artificial Intelligence and Deep Learning in Pathology (2021), 13-40.
[2]. S. Badillo, B. Banfai, F. Birzele, I. I. Davydov, L. Hutchinson, T. Kam-Thong, ... & J. D. Zhang, An introduction to machine learning, Clinical Pharmacology & Therapeutics, 107(4), 2020, 871-885.
[3]. A. S. Rao, M. Radanovic, Y. Liu, S. Hu, Y. Fang, K. Khoshelham, ... & T. Ngo, Real-time monitoring of construction sites: Sensors, methods, and applications, Automation in Construction, 136, 2022, 104099.
[4]. W. Liu, H. Wang, X. Shen, & I. W. Tsang, The emerging trends of multi-label learning, IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(11), 2021, 7955-7974.
[5]. F. Charte, A comprehensive and didactic review on multilabel learning software tools, IEEE Access, 8, 2020, 50330-50354.
[6]. W. Qian, J. Huang, F. Xu, W. Shu, & W. Ding, A survey on multi-label feature selection from perspectives of label fusion, Information Fusion, 100, 2023, 101948.
[7]. X. Zheng, P. Li, Z. Chu, & X. Hu, A survey on multi-label data stream classification, IEEE Access, 8, 2019, 1249-1275.
[8]. Z. Wan, X. Xia, D. Lo, & G. C. Murphy, How does machine learning change software development practices?, IEEE Transactions on Software Engineering, 47(9), 2019, 1857-1871.
[9]. D. Li, & C. Wang, In-depth research and analysis of multilabel learning algorithm, Journal of Sensors, 2022, 1-16.
[10]. Y. Zhu, K. M. Ting, & Z. H. Zhou, Multi-label learning with emerging new labels, IEEE Transactions on Knowledge and Data Engineering, 30(10), 2018, 1901-1914.
[11]. K. Y. S. S. Kanagaraj, & M. Nallappan, Methods for predicting the rise of the new labels from a high-dimensional data stream, International Journal of Intelligent Engineering & Systems, 16(1), 2022, 339-349.
[12]. S. K. Yallini, & N. Mahendiran, An Ensemble Methods of Predicting the New Labels with Concept Drift from a High-Dimensional Data Stream, International Journal of Advanced Research in Computer Science, 15(2), 2024, 92-100.
[13]. N. K. Mishra, & P. K. Singh, Feature construction and smote-based imbalance handling for multi-label learning, Information Sciences, 563, 2021, 342-357.
[14]. K. Dahiya, D. Saini, A. Mittal, A. Shaw, K. Dave, A. Soni, ... & M. Varma, Deepxml: A deep extreme multi-label learning framework applied to short text documents, Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 2021, 31-39.
[15]. M. Rezaei-Ravari, M. Eftekhari, & F. Saberi-Movahed, Regularizing extreme learning machine by dual locally linear embedding manifold learning for training multi-label neural network classifiers, Engineering Applications of Artificial Intelligence, 97, 2021, 104062.
[16]. S. Li, X. Xia, H. Zhang, Y. Zhan, S. Ge, & T. Liu, Estimating noise transition matrix with label correlations for noisy multi-label learning, Advances in Neural Information Processing Systems, 35, 2022, 24184-24198.
[17]. X. Fu, D. Li, & Y. Zhai, Multi-label learning with kernel local label information, Expert Systems with Applications, 207, 2022, 118027.
[18]. B. Liu, X. Liu, H. Ren, J. Qian, & Y. Wang, Text multi-label learning method based on

label-aware attention and semantic dependency, Multimedia Tools and Applications, 81(5), 2022, 7219-7237.

[19]. D. Zhao, H. Li, Y. Lu, D. Sun, D. Zhu, & Q. Gao, Multi-label weak-label learning via semantic reconstruction and label correlations, Information Sciences, 623, 2023, 379-401.

[20]. B. Liu, W. Li, Y. Xiao, X. Chen, L. Liu, C. Liu, ... & P. Sun, Multi-view multi-label learning with high-order label correlation, Information Sciences, 624, 2023, 165-184.

[21]. M. Huang, Y. Zhao, Y. Wang, F. Wahab, Y. Sun, & C. Chen, Multi-graph multi-label learning with novel and missing labels, Knowledge-Based Systems, 276, 2023, 110753.

[22]. A. Radford, L. Metz, & S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434, 2015.

[23]. Multi-label Classification Dataset Repository, Knowledge Discovery and Intelligent Systems KDIS University of Córdoba, Available: https://www.uco.es/kdis/mllresources/, Accessed: 06-March-2020.