An Online Transfer Learning Framework with Extreme Learning Machine for Automated Credit Scoring

Rana Alasbahi¹, XiaoLin Zheng², Editor IEEE
¹²Computer Science and Technology, Zhejiang University, Hangzhou, 310027 China.

Corresponding author: Rana Alasbahi (11721084@zju.edu.cn).

ABSTRACT Automated Credit Scoring (ACS) is the process of predicting user credit based on historical data. It involves analysing and predicting the association between the data and particular credit values based on similar data. Recently, ACS has been handled as a machine learning problem, and numerous models were developed to address it. In this paper, we address ACS issues concerning credit scoring in a batch of machine learning problems, namely, feature irregularities due to empty features in many records, class imbalance due to non-uniform statistical distributions of the records between classes, and concept drift due to changing statistical characteristics concerning certain classes and features with time. Considering the limited credit scoring data volume, we propose to address the challenge using the Transfer Learning with Lag (TLL) algorithm based on embedded shallow neural networks that enable knowledge transfer when the number of active features changes. Knowledge transfer is based on lags having an adaptive length that is changed based on performance change feedback. Furthermore, the framework proposes classifier aggregation and the chunk balancing mechanism for handling class imbalance. An evaluation was conducted using the Lending club, German, Default, and PPDai datasets. The results show the superiority of the proposed algorithm over the benchmarks in terms of the majority of classification metrics concerning both time series and overall results. TLL offered improvements of 58.6% and 28.2% over FA-ОСЕLM and OSELM using the Lending club dataset.

INDEX TERMS Credit Scoring, Machine Learning, Extreme Learning Machine, Probability of Default, Missing Features, Data Irregularities, Class imbalance.

I. INTRODUCTION
Managing credit risk and supporting credit application decision-making has become a demanding artificial intelligence and machine learning application. It comprises providing the probability of default for lending institutions’ clients and satisfying the minimum loss principle for business sustainability. Transitioning from manual borrowing application processing based on officers or expert-based credit scoring to establishing automated credit scoring helps create a more promising system to avoid credit and opportunity loss. Limiting user intervention is the general direction targeted using automated systems. The financial sector provides numerous examples of financial services having an automated credit-scoring decision-support system, e.g., internet banking firms in South Korea [1], based on a tablet banking system acting as a smart branch to enable various business functions concerning financial services. These services prompt the user to scan an ID using a mobile device camera after which, the user can access the majority of the bank services, eliminating the need to visit a branch for financial consultation or product services. According to [2], credit scoring is far from being a process implemented only by financial institutions. Other types of firms, such as mobile phone companies, insurance companies, or government departments, use similar approaches before accepting to provide their services. However, there is a concern about model choice and indicators to determine the best model and dynamics: how to introduce them to provide a figure concerning future risks. Automated credit scoring performance has been assessed using various approaches specified in the literature. Some studies used the binary classification problem. Others incorporated using data mining and machine learning techniques like discriminant analysis [3], neural network [4], support vector machine [5],
decision tree [6,7], logistic regression [8], fuzzy logic [9], genetic algorithm [10][11], Bayesian networks [12], hybrids methods [13][7][14][15], ensemble methods [16], and feature selection [17][18][5]. Researchers [19] indicated that previous studies used the binary classification system, which is insufficient to predict the score correctly. We highlight three additional issues, namely, feature irregularities, class imbalance, and concept drift. Applications might not provide complete information, leading to empty features on records, causing feature irregularities. Non-uniform statistical distributions of records cause class imbalance because most records belong to one class. Varying statistical characteristics of certain classes cause concept drift. Financial and economic features are highly dynamic with time. The literature around credit scoring does not jointly address these issues using a single framework to the best of our knowledge. This study aims to bridge this gap using a knowledge transfer integrated online extreme learning machine. This integration is accomplished using a single framework where knowledge transfer is combined with lag awareness. It enables avoiding concept drift by slowing model movement towards fast knowledge update. In addition, incorporating transfer learning enables using missing feature values to facilitate learning despite the absence of data. The remaining article is organised as follows: Section 2 discusses the literature review; Section 3 presents the methodology and the proposed model; Section 4 presents the experimental evaluation and results; lastly, Section 5 comprises concluding remarks and future work recommendations.

II. RELATED WORK
Managing credit risk and supporting credit application decision-making has become a demanding artificial intelligence and machine learning application. It comprises providing the probability of default for lending institutions’ clients and satisfying the minimum loss principle for business sustainability. Transitioning from manual borrowing application processing based on officers or expert-based credit scoring to establishing automated credit scoring helps create a more promising system to avoid credit and opportunity loss. Limiting user intervention is the general direction targeted using automated systems. The financial sector provides numerous examples of financial services having an automated credit-scoring decision-support system, e.g., internet banking firms in South Korea [1], based on a tablet banking system acting as a smart branch to enable various business functions concerning financial services. These services prompt the user to scan an ID using a mobile device camera after which, the user can access the majority of the bank services, eliminating the need to visit a branch for financial consultation or product services. According to [2], credit scoring is far from being a process implemented only by financial institutions. Other types of firms, such as mobile phone companies, insurance companies, or government departments, use similar approaches before accepting to provide their services. However, there is a concern about model choice and indicators to determine the best model and dynamics: how to introduce them to provide a figure concerning future risks. Automated credit scoring performance has been assessed using various approaches specified in the literature. Some studies used the binary classification problem. Others incorporated using data mining and machine learning techniques like discriminant analysis [3], neural network [4], support vector machine [5], decision tree [6,7], logistic regression [8], fuzzy logic [9], genetic algorithm [10][11], Bayesian networks [12], hybrids methods [13][7][14][15], ensemble methods [16], and feature selection [17][18][5]. Researchers [19] indicated that previous studies used the binary classification system, which is insufficient to predict the score correctly. We highlight three additional issues, namely, feature irregularities, class imbalance, and concept drift. Applications might not provide complete information, leading to empty features on records, causing feature irregularities. Non-uniform statistical distributions of records cause class imbalance because most records belong to one class. Varying statistical characteristics of certain classes cause concept drift. Financial and economic features are highly dynamic with time. The literature around credit scoring does not jointly address these issues using a single framework to the best of our knowledge. This study aims to bridge this gap using a knowledge transfer integrated online extreme learning machine. This integration is accomplished using a single framework where knowledge transfer is combined with lag awareness. It enables avoiding concept drift by slowing model movement towards fast knowledge update. In addition, incorporating transfer learning enables using missing feature values to facilitate learning despite the absence of data. The remaining article is organised as follows: Section 2 discusses the literature review; Section 3 presents the methodology and the proposed model; Section 4 presents the experimental evaluation and results; lastly, Section 5 comprises concluding remarks and future work recommendations.

I. Methodology
This section describes the methodology devised for accomplishing a credit scoring framework based on stream data while handling missing values (non-active features), data imbalance, and concept drift. The problem is formulated, followed by a general framework overview. Chunk balancing is described, following by preprocessing and normalisation. Subsequently, gamma generation and transfer learning are described. Lastly, ensemble learning, concept drift, and lag update are presented.
Table 1: Summary of the literature from the various aspect for addressing the problem of credit scoring

| Article                        | Missing features | imbalance | Concept drift | Online | Methods                                      | datasets                          |
|-------------------------------|------------------|-----------|---------------|--------|----------------------------------------------|-----------------------------------|
| Wenyu Zhang et al. [14]       | Average          | ✗         | ✗             | ✗      | Feature selection, Classifier selection, Genetic. | Australian, German PP Dai GMSC   |
| Joaquín Abellán et al. [23]   | ✗                | ✓         | ✗             | ✗      | Feature selection, Classifier ensemble, credal decision tree | Australian, German, Iranian Japanese Polish, UCDSD |
| Hongliang He et al. [24]      | ✗                | ✓         | ✗             | ✗      | Classifier ensemble, EBCA, PSO, RF, GBoost   | Australian, Japanese, German, Default Data, PP Dai Data, LC 2017 Q1 Data |
| Xia Yufei et al. [6]          | inherent sparsity aware splitting | ✓         | ✗             | ✓      | sequential ensemble, wrapperFS algorithm, XGBoost-TPE | German, Australian, Taiwan, P2P-A, P2P-B |
| Feng Shen et al. [28]         | ✗                | ✓         | ✗             | ✗      | Inference framework for rejection decision | Personal Chinese credit dataset |
| Pawel Plawiak et al. [22]     | ✗                | ✓         | ✗             | ✗      | Genetic selection, Feature extraction DGCEC | Australian |
| Sebastián Maldonado et al. [29]| ✗                | ✓         | ✗             | ✗      | three-way decision (Logistic Regression) | Chilean bank dataset |
| Xiaodong Feng et al. [16]     | mean             | ✓         | ✗             | ✓      | Dynamic ensemble classification (DECSP) | Australian, German, Japanese, Taiwan DCCC, Chinese 1,2,AER, Th02, PAKDD2010, Kaggle |
| Sebastián Maldonado et al. [5]| filter           | resampling | ✗             | ✓      | \( \text{l}_{1} \text{svm} \) \( \text{l}_{2} \text{svm} \) Feature selection | Chilean bank dataset |
| Leopoldo Melo Junior et al. [20]| mean/mode       | ✓         | ✗             | ✗      | Dynamic selection, RMKnn | German, Default, PP Dai, Iranian dataset, privat dataset, GiveMe, LC 2015 Q1 23 |
| Diwakar Tripathi et al. [21]  | unique integer number | ✗         | ✓             | ✗      | (EELM)                                    | Australian, Japanese, German-categorical, German-numerical |
| Pawel Plawiak et al. [27]     | ✗                | ✓         | ✗             | ✗      | DGHN L                                     | German |
| Jasmina Nalić et al. [7]      | remove           | ✓         | ✗             | ✗      | Feature selection, Ensemble classifier (GLM, DT, SVM, NB) | Microfinance institution dataset |
| Tong Zhang et al. [25]        | Mean             | ✓         | ✗             | ✗      | Heterogeneous Ensemble Classifier           | German, Default, Chilean, GMSC  |
| Wei Zhang et al. [30]         | Filling          | ✓         | ✗             | ✗      | CSMIL                                      | Chinese bank dataset |
| Xiaohong Chen et al. [26]     | Mean             | ✓         | ✗             | ✗      | GSCI                                        | Australian, Japanese, German, Default, RR Dai, ProsperLoan, Lending Club |
| Our Framework                 | Transfer Learning| ✓         | ✓             | ✓      | TLL-AD, TLL-ADWIN                           | German, Default, PP Dai, Lending Club |
A. Problem Formulation

Consider a sequential dataset \( D = \{ (X_t, Y_t, t = 1, 2, \ldots, N) \} \), where \( X_t \) denotes a chunk arriving at time \( t \) and \( Y_t \) denotes chunk labelling information. \( Y_t \) might contain the ground truth of \( X_t \) when class information is available; otherwise, it contains the character \( \bot \), indicating that label information of \( X_t \) is not available.

\( X_t \in \mathbb{R}^{n \times m} \) and \( Y_t \in \mathbb{N}^{n \times 1} \cup \{ \bot \} \) where \( \mathbb{R} \) denotes the set of real numbers, \( \mathbb{N} \) denotes the set of natural numbers, and \( n \) denotes the number of rows in chunk \( t \). The objective is to predict the class of samples \((X_t, \bot)\) with a minimal percentage of false predictions.

B. Transfer Learning with Lag (TLL) Framework

The Transfer Learning with Lag (TLL) framework is depicted in Figure 1. It is observed that the arriving chunk first moves to a load balancing process. It is based on a buffer that stores and uses recent samples to balance chunks to maintain similar class percentages for the labelled samples. It is used to reduce bias caused by the imbalance in labelled sample distribution. Subsequently, a pre-processing stage is used for normalisation. Next, the Gamma generation stage is invoked; it codes active and inactive features.
using 1 and 0. Gamma generation information is used by the learning block (TL) to build the classifiers for the next time step using previous classifiers and memory information.

**Algorithm 3:** Pseudocode of generating Gamma for encoding feature availability

```
Input: featureSize, chunkCurrent
Output: Gamma
Start
1. Gamma = zeros(1, featureSize)
2. for each feature of chunkCurrent do
   3. Gamma(feature) = 1
end
3. return Gamma
End
```

Furthermore, memory is also updated using transfer learning to ensure that information is saved for future learning. After obtaining the classifiers for the next time step, an aggregator is used for prediction. Predictions are used to detect concept drift using available class information, while prediction is used to adjust lag values. Subsequent sections present internal block operations comprehensively. As depicted in Algorithm 1, the framework uses random weights to initiate the learners. Next, it uses the boosting data to update the knowledge of the learners. Boosting data represents labelled data used to provide initial knowledge to the system. Next, the generate Gamma process is applied to the boosting data to generate active feature indices; these are saved for subsequent chunks.

**Algorithm 4:** Pseudocode of transfer learning TL

```
Input: previousUnN, previousGamma, Memory, Gamma
Output: Memory, Learner
Start
1. for each feature do
   2. if (Gamma(feature) = 1 and previousGamma(feature) = 0) then
      3. if (findWeights(Memory)) then
         4. newWeights(feature) = restoreWeights(Memory)
      6. else
         7. newWeights(feature) = randInitiate();
   end
   9. else if (Gamma(feature) = 0 and previousGamma(feature) = 1) then
      11. Memory(feature) = previousNNWeights(feature)
   end
End
```

Transfer uses previous and current Gamma to determine the weights that need to be saved in memory or restored from memory. The transfer learning process outputs current moment learners used to predict current sample labels using aggregation, which comprises an ensemble learning rule that uses the current moment learners for prediction. Aggregation results are provided to the concept drift detector, which compares deviation values from the ground truth concerning the data and decides if drift has occurred. For a concept drift scenario, a lag update is performed, and free Memory is invoked. The function clears the memory of outdated weights and knowledge.

**C. Chunk Balancing**

Balancing is the first step of the framework. Its role is to enable balanced training data for the learners from the label perspective. It executes based on the balancing period or condition. It uses class samples to calculate the ratio of each class against the entire sample count and uses the majority ratio to complement minorities using data from the buffer.

**D. Pre-processing and Normalisation**

Data pre-processing combines various steps. It starts by converting the data to numerical values. Every categorised feature is encoded using a binary scheme. Multi-category features are split into binary features under the assumption that they have a non-deterministic relation, leading to several binary features that are one count less than the number of categories. In addition, we eliminate statistical redundancy of features by calculating the correlation matrix and removing features having a correlation value of more than 0.95. It enables data compaction and provides a discriminative version. Subsequently, Equation 1 is used to normalise data:

\[
x = 2 \times \frac{x - \min(X_i)}{\max(X_i) - \min(X_i)} - 1 \tag{1}
\]
E. Gamma Generation

Assuming that the dataset is combined of a set of chunks ordered with respect to time, as shown in Equation 2:

\[ D = \{C_0, C_1, \ldots, C_{N_c}\} \] (2)

Where \( N_c \) denotes the number of chunks.

We consider that each chunk has the same active features, which means that there is no change between the active features among the chunk records. In the case of absent or missing features, we prefer avoiding dummy values to indicate an absent feature because it might affect prediction accuracy. Instead, we build an indicator vector for the missing feature. An active feature vector \( \Gamma \) denotes this vector with a size of \( 1 \times m \).

Here, \( m \) is the number of features in the data, and any component of this vector is binary or \( \forall x_i \in \Gamma \rightarrow x_i \in \{0,1\} \).

An active feature is indicated using a value of 1. It is depicted using Equation 3:

\[ D = \{(X_i, Y_i), \quad i = 0, 1, 2, \ldots, N_c\} \] (3)

\[ \forall x, y \in C_i, \quad \Gamma(x) = \Gamma(y) \] (4)

where \( \Gamma(x) \) denotes the set of features in vector \( x \).

F. Transfer Learning.

Transfer learning is used to create new learners for predicting current chunk labels. Hence, learner input must match the active feature of the current chunk. In addition, transfer learning is responsible for two tasks: (1) restore old knowledge from memory by inserting the weights connected to the new active features into the learners (lines 5-8 of the pseudocode); (2) maintain the memory by storing the weights connected to the non-active features from the previous learners (line 11 of the pseudocode).

G. Ensemble Learning (Aggregation)

Ensemble learning is responsible for aggregating the basic learners. It is based on majority decisions for an odd number of classifiers and performing classifiers for even classifiers. Algorithm 5 depicts the pseudocode.

H. Concept Drift Detection and Lag Update

Concept drift is detected using two methods: The first method is named the accuracy drop (AD) method. It calculates accuracy using labelled samples and triggers concept drift when a decline occurs over time. It is tested mathematically using Equation 5.

\[ \Delta Acc = Acc_{t-1} - Acc_t > \text{threshold} \] (5)

Where \( Acc_t \) denotes the calculated accuracy at the moment \( t \), we designate TLL with AD as (TLL-AD).

ADWIN is the second method that uses a moving window (buffer) with a fixed maximum length buffer for old samples to determine concept drift. The method iteratively drops samples from the window tail until a smaller window is obtained with no concept drift. A window without concept drift is one lacking statistical significance concerning the differences between all sub-window partitions. For efficient calculations, we use ADWIN2, which uses logarithmic partitioning for checking concept drift inside the window. We designate TLL with ADWIN2 as TLL-ADWIN2.

The lag update changes lag from one time moment to another based on the concept drift decision. Equation 6 is used for changing the lag.

\[ \text{Lag}(t) = \begin{cases} \text{Lag}(t-1) - 1 & \text{if concept drift exists} \\ \text{Lag}(t) & \text{otherwise} \end{cases} \] (6)

\[ \text{Lag}(0) = \text{Lag}_0 \]

Where

\( \text{Lag}(t) \) denotes the lag at moment \( t \).

\( \text{Lag}_0 \) denotes the initial lag.

Lastly, the free memory process removes older weights from memory based on the lag provided by the update.

II. Memory Freeing

Memory freeing cleans the memory of the weights related to outdated classifiers identified by \( \text{Lag}(t) \). Hence, for every moment \( t \), the free Memory process is supposed to remove all weights related to classifiers trained at the moment \( t - \text{Lag}(t) \) or older. This process ensures that older weights are not restored from memory and expired knowledge is not considered.

III. Experimental Evaluation and Results

For evaluation, we compared Transfer Learning with Lag TLL-OSELM using two benchmarks, namely, Feature Adaptive FA-OSELM and OSELM. Results were generated based on the parameters depicted in Table 2. We selected the
sigmoid activation function, 100 hidden neurons, 50 records per chunk, 50 time-units as initial tag, and 0.5 $\Delta$Acc.

Table 3: The parameters of the experimental evaluation

| Parameter name       | Value   |
|----------------------|---------|
| Activation function  | sigmoid |
| Number of hidden neurons | 100     |
| Number of records per chunk | 50      |
| Initial lag          | 3       |
| $\Delta$Acc          | 0.5     |

Four different datasets, namely, Default, German, PPDai, and Lending Club, were used to compile the results. The details for every dataset are provided in Table 3. The Lending Club dataset is the largest, while German is the smallest. Datasets have different numbers of missing attributes, which means a different percentage of missing features.

Table 4: The Details of the used datasets

| Dataset        | Default | German | PPDai  | Lending club |
|----------------|---------|--------|--------|--------------|
| Record numbers | 300,000 | 100,000| 555,96 | 200,409,1    |
| Class numbers  | 2       | 2      | 2      | 8            |
| Missing data   | 2.4%    | 1.9%   | 2.8%   | 13.4%        |
| Imbalance      | 28.40   | 42.91  | 14.83  | 5.74         |

1. CONFUSION MATRIX

We present the confusion matrix for every learner: the proposed TLL OSELM and the two benchmarks FA- OSELM and OSELM. We find that TLL-AD accomplished an accuracy of 75.20% and 76.7% for the first and second class in Lending Club, superior to FA-OSELM and OSELM. Additionally, TLL-ADWIN accomplished the best overall accuracy for datasets at 81.47% for the Lending Club dataset. However, both FA-OSELM and OSELM were inferior to the proposed TLL model; these methods had accuracy values of 52.7% and 52.6% for the two classes in FA-OSELM and 58.7% and 60.3% for the two classes 1 and 2 in OSELM. Similarly, for the German dataset, TLL-AD obtains an accuracy of 69.00% for both classes and 57.8% and 68.3% for classes 1 and 2, superior to FA-OSELM that achieved 56.9% and 67.2%. Similar superior TLL performance was observed for PPDai data and default datasets. TLL-ADWIN achieved the highest accuracy for the PPDai dataset at 92.9% and 50.3% for classes 1 and 2. However, FA-OSELM accuracy stood at 82.5% and 50.2%, respectively, while FA- OSELM accuracy values were 73.1% and 50.1% for the two classes.
2. ACCURACY

Figures 5-8 depict overall accuracy for Lending Club, German, default, and PPDai datasets, respectively. The graphs for all datasets indicate that TLL outperformed both FA-OSELM and OSELM in terms of the reached accuracy. Also, TLL-ADWIN2 reached an accuracy close to 80% for three datasets, namely, Lending Club, German, and Default, while accuracy for PPDai was about 88%. Moreover, we observe that FA-OSELM and OSELM had accuracy values between 50% and 80%, while OSELM attained 86.9% for the PPDai dataset.

Another observation is that OSELM outperformed FA-OSELM in terms of accuracy for all datasets except the German dataset. It indicates that transfer learning concerning FA is not adequate, considering that it transfers knowledge from one neural network to the next without considering lag to handle the evolving nature of data. TTL provided better results regarding behaviour interpretation by integrating three algorithm functionalities, i.e., data imbalance processing using the window technique, ensemble learning, and handling concept drift using knowledge transfer based on lag and memory.

We elaborate on the predicted time series using Figures 9-12. The plots indicate that accuracy oscillates for the four approaches and is caused by dynamic changes to data characteristics. Consequently, learner performance degrades. We also observe that in time intervals, FA outperformed OSELM, while the opposite happened for others. Nevertheless, the plots

| 1) Lending Club dataset confusion matrix | 2) PPDai dataset confusion matrix |
|----------------------------------------|----------------------------------|
| ![Lending Club confusion matrix](image1) | ![PPDai confusion matrix](image2) |

| 3) German dataset confusion matrix |
|-----------------------------------|
| ![German confusion matrix](image3) |

| 4) Default dataset confusion matrix |
|------------------------------------|
| ![Default confusion matrix](image4) |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
indicate that TLL methods are generally superior to FA and OSELM.

5) Accuracy result comparison of lending club dataset

6) Accuracy result comparison of PPDai dataset

7) Accuracy result comparison of German dataset

8) Accuracy result comparison of Default dataset
3. **OVERALL PRECISION**

Precision indicates the percentage of actual positive predicted values against all positive predicted values, as depicted in Figure 13-16. This metric is essential to indicate the learner's level of avoiding bias for negative samples that indicate the majority class. We see that TTL accomplished the best precision for the Lending Club dataset, close to 100% levels. However, the difference between the approaches in terms of precision is more pronounced for other datasets. For example, Figure 15 indicates that TLL-ADWIN2 obtained a precision of about 86%, compared to slightly lower values for FA and OSELM. Like accuracy, we present the time series for precision using separate plots, depicted using Figures 17-20.

We elaborate performance further by presenting detailed time series data using Figure 17-20. As seen, the approaches have volatility because of the dynamic characteristics of actual data. However, it is evident from the German dataset that TLL-ADWIN2 was maintaining better performance levels.

4. **OVERALL RECALL**

Recall provides the percentage of actual predicted positive records from all actual positive records, depicted using Figures 21-24. It is evident that the PPDai dataset produced the best results independent of the approach. There is an exception concerning other data, where FA and OSELM recall values declined compared to the proposed approach. It indicates the bias of FA and OSELM compared to TTL, which provided better recall. Figures 25-28 depict the time series plots for all methods and datasets to elaborate recall performance.
13) Overall precision comparison result of Lending Club dataset

14) Overall precision comparison result of PP Dai dataset

15) Overall precision comparison result of German data

16) Overall precision comparison result of Default dataset

17) Precision time series comparison for Lending club dataset

18) Precision time series comparison for German dataset
5. OVERALL SPECIFICITY

Overall specificity provides the percentage of correctly identified negatives, illustrated using Figures 29-32. It is evident that TLL-AD and TLL-ADWIN2 approaches accomplished the best overall specificity for all datasets, compared to FA and OSELM. It indicates that the TLL approach has significantly higher overall specificity than FA and OSELM, indicating the model’s ability to predict true negatives of each available category. Figures 33-36 depict specificity for time series data for all datasets and methods.

6. OVERALL NPV

Negative predictive values are used to measure the accuracy of a negative test result. The results are illustrated in Figures 37-40. We observe that both TLL-AD and TLL-ADWIN2 have the best NPV results compared to FA and OSELM. TLL-ADWIN2 accomplished the best NPV of 89% for the Lending Club dataset, slightly superior to FA and OSELM. It indicates that FA and OSELM have relatively poor performance concerning NPV. Figures 41-44 present the time series for all methods and datasets to depict overall NPV performance.

7. OVERALL F-MEASURE

The F-Measure is regularly used to evaluate the performance of imbalanced classification algorithms as F-Measure. The results are illustrated in Figures 45-48. As we observe, TLL outperformed both FA-OSELM and OSLEM in terms of the reached F-measure for all datasets except for PPDai, where the F-measure of FA and OSELM increased with respective precision. We see that the best accomplished F-measure for TLL was for both Lending Club and default datasets at a level close to 88%, compared to a slightly lower precision for FA and OSELM. It indicates that FA and OSELM have poor performance compared to TLL, which provides a significantly higher F-measure. Figures 49-52 present the time series of all methods and datasets to depict overall F-measure performance.

8. OVERALL G-MEAN

G-Mean measures the balance between classification performance for both majority and minority classes. The results are illustrated in Figures 53-56. It is evident that the proposed TLL approach outperformed both FA-OSELM and OSELM in terms of the reached G-mean. It also indicates the poor performance of both FA and OSELM compared to TLL. Figures 57-60 present the time series of all methods and datasets to elaborate overall specificity performance.

Table 5 Summary of improvement percentage over the benchmarks.

| IP      | PPDai | DEFAULT | GERMAN | LENDING CLUB |
|---------|-------|---------|--------|--------------|
| TLL-AD  | 0.4146| 2.39    | 9.7    | 3.217        |
| FA      | 0.1608| 9.0     | 10.6   | 58.6         |
| TD      | 0.2645| 3.0     | 10.7   | 28.2         |
21) Overall recall comparison result of Lending Club dataset

22) Overall recall comparison result of PPDai dataset

23) Overall recall comparison result of German dataset

24) Overall recall comparison result of Default dataset
25) Recall time series comparison for Lending club dataset

26) Recall time series comparison for German dataset

27) Recall time series comparison for default dataset

28) Recall time series comparison for PPDai dataset
29) Overall specificity comparison result of Lending club data

30) Overall specificity comparison result of PPDAI data

31) Overall specificity comparison result of German data

32) Overall specificity comparison result of Default data

33) Specificity time series comparison for Lending club dataset

34) Specificity time series comparison for German dataset
35) Specificity time series comparison result for Default dataset

36) Specificity time series comparison result for PPDai dataset

37) Overall NPV comparison result of Lending club dataset

38) Overall NPV comparison result of PPDai dataset
39) Overall NPV comparison result of German dataset

40) Overall NPV comparison result of Default dataset

41) NPV time series comparison for Lending Club dataset

42) NPV time series comparison for German dataset

43) NPV time series comparison for Default dataset

44) NPV time series comparison for PPDai dataset
45) Overall F-Measure time series comparison for Lending club dataset

Overall F-Measure time series comparison for PPDai dataset

46) Overall F-Measure time series comparison for PPDai dataset

47) Overall F-Measure time series comparison for German dataset

48) Overall F-Measure time series comparison for Default dataset
49) F-measure time series comparison for Lending club dataset

50) F-measure time series comparison for German dataset

51) F-measure time series comparison for Default dataset

52) F-measure time series comparison for PPDai dataset
Overall G-Mean result comparison for Lending Club dataset

Overall G-Mean result comparison for PPDai dataset

Overall G-Mean result comparison for German dataset

Overall G-Mean result comparison for Default dataset

G-Mean time series comparison for Lending Club dataset

G-Mean time series comparison for German dataset
IV. Summary and Conclusion

This article handles the credit scoring problem as a batch learning problem. We considered three specific problems: feature irregularities due to empty features in many records, class imbalance due to non-uniform statistical distributions of records among classes, and concept drift due to varying statistical characteristics for specific classes based on certain features with respect to time. The article proposed transfer learning to handle evolving features and changes concerning active/disabled features across batches. It also incorporated lag to remove outdated knowledge and focus on new knowledge based on adaptive lag and accuracy-change feedback. Furthermore, the framework proposes a chunk balancing mechanism and classifier aggregation for handling class imbalance. Additionally, window-based chunk balancing was incorporated to augment imbalance handling. The evaluation was conducted based on the Lending Club, German, Default, and PPDai datasets. The results show the superiority of the proposed algorithm over the benchmarks in terms of the majority of classification metrics concerning both time series and overall results. The highest improvement percentage was 53% over OSELM and 65% over FA-OSELM. Future work should incorporate feature selection to handle dynamic changes concerning relevant features and high dimensional data. In addition, the developed framework should be evaluated on other machine learning fields that share the same issues concerning the credit scoring problem.

V. References:

[1] H. Chang and M. Park, “A Smart e-Form for Effective Business Communication in the Financial Industry,” *Bus. Commun. Res. Pract.*, vol. 1, no. 2, pp. 95–101, 2018.

[2] D. Guegan and B. Hassani, “Regulatory learning: How to supervise machine learning models? An application to credit scoring,” *J. Financ. Data Sci.*, vol. 4, no. 3, pp. 157–171, 2018.

[3] R. Levy, Aldo and Baha, “Credit risk assessment: a comparison of the performances of the linear discriminant analysis and the logistic regression,” *Int. J. Entrep. Small Bus.*, vol. 42, pp. 169–186, 2021.

[4] A. Byanjankar, M. Heikkila, and J. Mezei, “Predicting credit risk in peer-to-peer lending: A neural network approach,” *Proc. - 2015 IEEE Symp. Ser. Comput. Intell. SSCI 2015*, pp. 719–725, 2015.

[5] S. Maldonado, C. Bravo, J. López, and J. Pérez, “Integrated framework for profit-based feature selection and SVM classification in credit scoring,” *Decis. Support Syst.*, vol. 104, pp. 113–121, 2017.

[6] Y. Xia, C. Liu, Y. Li, and N. Liu, “A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring,” *Expert Syst. Appl.*, vol. 78, pp. 225–241, 2017.

[7] D. Ž. Nalić, Jasmina, Goran Martinović, “New hybrid data mining model for credit scoring based on feature selection algorithm and ensemble classifiers,” *Adv. Eng. Informatics*, vol. 45 (2020):

[8] H. Sutrisno and S. Halim, “Credit Scoring Refinement Using Optimized Logistic Regression,” *2017 Int. Conf. Soft Comput.*
Intell. Syst. Inf. Technol., no. 2, pp. 26–31, 2017.

[9] G. Bennouna and M. Tkouat, “Fuzzy logic approach applied to credit scoring for microfinance in Morocco,” Procedia Comput. Sci., vol. 127, pp. 274–283, 2018.

[10] C. S. Ong, J. J. Huang, and G. H. Tzeng, “Building credit scoring models using genetic programming,” Expert Syst. Appl., vol. 29, no. 1, pp. 41–47, 2005.

[11] K. Tran, T. Duong, and Q. Ho, “Credit scoring model: A combination of genetic programming and deep learning,” 2016 Futur. Technol. Conf., no. December, pp. 145–149, 2016.

[12] et al. Teles, Germanno, “Artificial neural network and Bayesian network models for credit risk prediction,” J. Artif. Intell. Syst., vol. 2 (2020):

[13] B. Zhu, W. Yang, H. Wang, and Y. Yuan, “A hybrid deep learning model for consumer credit scoring,” 2018 Int. Conf. Artif. Intell. Big Data, ICAIBD 2018, pp. 205–208, 2018.

[14] W. Zhang, H. He, and S. Zhang, “A novel multi-stage hybrid model with enhanced multi-population niche genetic algorithm: An application in credit scoring,” Expert Syst. Appl., vol. 121, pp. 221–232, 2019.

[15] M. Ala’raj and M. F. Abbod, “A new hybrid ensemble credit scoring model based on classifiers consensus system approach,” Expert Syst. Appl., vol. 64, pp. 36–55, 2016.

[16] X. Feng, Z. Xiao, B. Zhong, J. Qiu, and Y. Dong, “Dynamic ensemble classification for credit scoring using soft probability,” Appl. Soft Comput., vol. 65, pp. 139–151, 2018.

[17] X. Li, L. Chen, Z. Yang, H. Wang, V.-S. Ha, and H.-N. Nguyen, “Credit scoring with a feature selection approach based deep learning,” 2016 7Th Int. Conf. Mech. Ind. Manuf. Technol. (Mimt 2016), vol. 54, no. 54, pp. 1–7, 2016.

[18] S. Maldonado, J. Pérez, and C. Bravo, “Cost-based feature selection for support vector machines: An application in credit scoring,” Eur. J. Oper. Res., vol. 261, no. 2, pp. 656–665, 2017.

[19] L. Munkhdalai, T. Munkhdalai, O.-E. Namsrai, J. Y. Lee, and K. H. Ryu, “An empirical comparison of machine-learning methods on bank client credit assessments,” Sustainability, vol. 11, no. 3, p. 699, 2019.

[20] L. Melo Junior, F. M. Nardini, C. Renso, R. Trani, and J. A. Macedo, “A novel approach to define the local region of dynamic selection techniques in imbalanced credit scoring problems,” Expert Syst. Appl., vol. 152, p. 113351, 2020.

[21] D. Tripathi, D. R. Edla, V. Kuppili, and A. Bablani, “Evolutionary Extreme Learning Machine with novel activation function for credit scoring,” Eng. Appl. Artif. Intell., vol. 96, p. 103980, 2020.

[22] P. Plawiak, M. Abdar, and U. Rajendra Acharya, “Application of new deep genetic cascade ensemble of SVM classifiers to predict the Australian credit scoring,” Appl. Soft Comput., vol. 84, p. 105740, 2019.

[23] J. Abellán and J. G. Castellano, “A comparative study on base classifiers in ensemble methods for credit scoring,” Expert Syst. Appl., vol. 73, pp. 1–10, 2017.

[24] H. He, W. Zhang, and S. Zhang, “A novel ensemble method for credit scoring: Adaption of different imbalance ratios,” Expert Syst. Appl., vol. 98, pp. 105–117, 2018.

[25] T. Zhang and G. Chi, “A heterogeneous ensemble credit scoring model based on adaptive classifier selection: An application on imbalanced data,” Int. J. Financ. Econ., 2020.

[26] X. Chen, S. Li, X. Xu, F. Meng, and W. Cao, “A Novel GSCI-Based Ensemble Approach for Credit Scoring,” IEEE Access, vol. 8, pp. 222449–222465, 2020.

[27] P. Plawiak, M. Abdar, J. Plawiak, V. Makarenkov, and U. R. Acharya, “DGHNLS: A new deep genetic hierarchical network of learners for prediction of credit scoring,” Inf. Sci. (Ny), vol. 516, pp. 401–418, 2020.

[28] F. Shen, X. Zhao, and G. Kou, “Three-stage reject inference learning framework for credit scoring using unsupervised transfer learning and three-way decision theory,” Decis. Support Syst., vol. 137, p. 113366, 2020.

[29] S. Maldonado, G. Peters, and R. Weber, “Credit scoring using three-way decisions with probabilistic rough sets,” Inf. Sci. (Ny), vol. 507, pp. 700–714, 2020.

[30] W. Zhang, W. Xu, H. Hao, and D. Zhu, “Cost-sensitive multiple-instance learning
method with dynamic transactional data for personal credit scoring,” *Expert Syst. Appl.*, vol. 157, p. 113489, 2020.