IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 11, 2022, accepted April 8, 2022, date of publication May 2, 2022, date of current version May 5, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3171569

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

RANA ALASBAHI AND XIAOLIN ZHENG, (Senior Member, IEEE)
Computer Science and Technology, Zhejiang University, Hangzhou 310027, China
Corresponding author: Rana Alasbahi (alssallamy16@hotmail.com)

ABSTRACT Automated Credit Scoring (ACS) is the process of predicting user credit based on historical data. It involves analyzing 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.

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],

The associate editor coordinating the review of this manuscript and approving it for publication was Liangxiu Han.

decision tree [6], [7], logistic regression [8], fuzzy logic [9], genetic algorithm [10, 11], Bayesian networks [12], hybrids methods [7], [13]–[15], ensemble methods [16], and feature selection [5], [17], [18]. 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 organized as follows: Section II discusses the literature review, Section III presents the methodology and the proposed model, Section IV presents the experimental evaluation and results; lastly, Section VI comprises concluding remarks and future work recommendations.

II. BACKGROUND AND RELATED WORK

The problem of credit scoring is discussed in the literature from various aspects: the first one is the handling of the missing features, the second one is the concept drift awareness and handling, the third one is approaches based on single classifier and the last one is the ensemble learning based approaches. Dedicate a separate sub-section for reviewing each of the aspects in the literature.

A. FEATURE MISSING BASED APPROACHES

In the work of [14], multi-stage hybrid model that integrates feature and classifier selection based on optimization was proposed. The optimization has been proposed based on multi-population of genetic with enhancement of crossover, mutation, and adding niche points and migrations. However, the work has dealt with the problem as simple binary classification with non-handling of missing features in a proper way, i.e., it calculated the average to complete the missing values of features in pre-processing step. In addition, no handling or considerations to concept drift. In the work of [20], the problem of missing data has been handled based on decomposing the dataset into several nonoverlapping subsets based on the missing patterns. A feature selection is enabled based on logistic regression to perform joint feature learning on all subsets. In the work of [21], the feature missing is handled based on replacing the missing values with the average in the numerical data. This is a common way for many methods in the literature. Other researchers have replaced the missing features with the median value assuming that the fraction of missing features is lower than a certain percentage or decided to remove the entire feature when the fraction of missing values is higher than the considered percentage [22], [23],Other researchers have considered that Replacing the missing data in rejected applicants with the mean or median of accepted applicants may lead to confusing classification boundaries and consequently poor results. Therefore, the missing features were replaced with 0 for rejected samples to maintain the dissimilarity between accepted and rejected samples [24].

B. CONCEPT DRIFT BASED APPROACHES

Several works have been developed for credit scoring with an awareness of the concept drift matter. In the work of [25], a sample-based online learning ensemble (SOLE) for client credit assessment is proposed. A multiple time scale ensemble classifiers and a novel sample-based online class imbalance learning procedure are proposed to handle the potential concept drift and class imbalance in the client credit assessment data streams. In the work of [26], a comprehensive online active learning framework that includes an ensemble classifier, a drift detector, a label sliding window, sample sliding windows, and an initialization training sample sequence was proposed. Next, a variable threshold uncertainty strategy based on an asymmetric margin threshold matrix is designed to comprehensively address the problem that a given class can simultaneously be a majority to a given subset of classes while also being a minority to others. Lastly, a weighted formula was proposed that comprehensively considers the class imbalance ratio of the sample's category and the prediction difficulty. In the work of [27], Dynamic financial distress prediction (DFDP) was proposed. two DFDP approaches based on time weighting and Adaboost support vector machine (SVM) ensemble. One is the double expert voting ensemble based on Adaboost-SVM and Timeboost-SVM (DEVE-AT), which externally combines the outputs of an error-based decision expert and a time-based decision expert. The other is Adaboost SVM internally integrated with time weighting (ADASVM-TW), which uses a novel error-time-based sample weight updating function in the Adaboost iteration. These two approaches consider the time weighting of samples in constructing an Adaboost-based SVM ensemble in order to enable them for handling concept drift.

C. SINGLE CLASSIFIER BASED APPROACHES

In the work of [5], the support vector machine SVM wa used with the incorporation of incorporates a group penalty function in the SVM formulation in order to penalize the variables simultaneously that belong to the same group, assuming that companies often acquire groups of related variables for a given cost rather than acquiring them individually. In the work of [28], the credit scoring problem has been tackled as a classification problem using an extreme learning machine by proposing a new algebraic activation function that has the feature of convergence toward sign when the parameter goes to infinity. In addition, ELM has been optimized using an evolutionary bat algorithm. This work has not focused on the specific issues in the credit scoring like imbalance dataset, a feature missing, and concept drift.

The work in [29], constructs an ensemble credit scoring model with a single extreme learning machine (ELM) classifier to address the imbalanced problem issue in the data, In addition to the weighting method concerning their classification accuracy based on generalized fuzzy soft sets (GFSS) theory. The work proposed a cosine-based distance measurement algorithm of GFSS to calculate the weights of each ELM classifier. The work in [30], [31], investigates classic algorithms for predicting credit scoring through using single classifiers and compared it with benchmark models. In the work [32], deep learning ensample classification and synthetic minority oversampling technique SMOTE has been used to train credit data; the proposed model was competitive with the state of the art works in addressing the imbalanced credit risk problems. However, many models have demonstrated their ability to be more accurate than single classifier models through the development of aggregation systems or multiple classifiers based consensus approach [33], [34]. In the work [34], a new clustering method based on consensus classifiers proposed to combine Multiple Classifier Systems (MCS) of different classification algorithms whereby the ensemble classifiers can work and collaborate as a group in which their decisions shared amongst classifiers.

D. ENSEMBLE LEARNING BASED APPROACHES

In [23] an approach based on ensemble learning for selecting the best base classifier was proposed. It has dealt with only the imbalance issue with ignoring the concept drift, the online nature, and the missing features. In a similar concept that uses ensemble learning [24], the authors have used two classifiers, namely, random forest and extreme gradient boosting with particle swarm optimization to optimize the parameters of the base classifiers. In the work of [6], the problem of credit scoring has been tackled based on using ensemble learning with focusing on how to optimize the hyper-parameters of the classifiers. Hence, they developed a sequential boosting-based ensemble learning model using Gradient boost using Bayesian hyper-parameter optimization of the Gboost.

In the work of [28], three stages learning framework has been applied for credit scoring. In the first learning stage, the learning framework measures the similarities between each rejected sample and the accepted training subset. Then three-way decision theory is applied to divide the rejected samples into positive, boundary, and negative regions. In the second stage of the proposed framework, the rejected samples in the positive region and the accepted training data are combined to train the self-taught learning STL model. Note that the STL method only utilizes the credit data variables; that is, the training data label information is not used; therefore, the STL model is basically an unsupervised learning method. The higher-level accepted sample training and test subsets are transformed when the base vectors and the activations are derived. Finally, in the third binary classification stage, the supervised learning algorithms are trained using the reconstructed training data features, with the trained classifiers being validated with the test samples. In [22], a hybrid model that merges evolutionary computation, ensemble learning, and deep learning was developed. It comprises a novel 16-layer genetic cascade ensemble of classifiers: two types of SVM classifiers, normalization techniques, feature extraction methods, three types of kernel functions, parameter optimizations, and a stratified 10-fold cross-validation method. Their work does not assume sequential learning, and it does not handle the issues of concept drift or missing features.

Algorithm 1 Pseudocode of the General Framework of Transfer Learning With Lag

Input: boostingChunk, featureSizem Lag.
Output: Yp

```
Start
Initial Memory Initial Learner
Learner=updateKnowledge (learner, boostingChunk)
previousGamma =generateGamma (boostingChunk)
initiate Memory (featuresize)
for every time t do
    Memory =freeMemory(t, lag, Memory)
    foreach new arrived chunk do
      if the chunk is for prediction then
         Gamma=generateGamma(chunk, featureSize)
         learner, Memory=Transfer
            (Gamma, previousGamma, Memory,learner)
         Yp=performPrediction (learner)
         previousGamma=Gamma
      else if the chunk is for correction then
         learner = updateKnowledge
   (learner, chunkCurrent )
      end
    end
end
End
```

In the work of [20], an evaluation of the suitability of dynamic selection techniques for credit scoring problems to deal with an imbalanced dataset was proposed, and Reduced Minority k-Nearest Neighbours (RMkNN) was presented for enhancing state of the art in defining the local region of dynamic selection techniques for imbalanced credit scoring datasets. In the work of [27], an architecture for training and prediction named as named Deep Genetic Hierarchical Network of Learners (DGHNL) was proposed. It involves different learners, including Support Vector Machines (SVM), k-Nearest Neighbours (kNN), Probabilistic Neural Networks (PNN), and fuzzy systems. The model applies deep learning, ensemble learning, supervised training, layered learning, genetic selection of features (attributes), genetic optimization

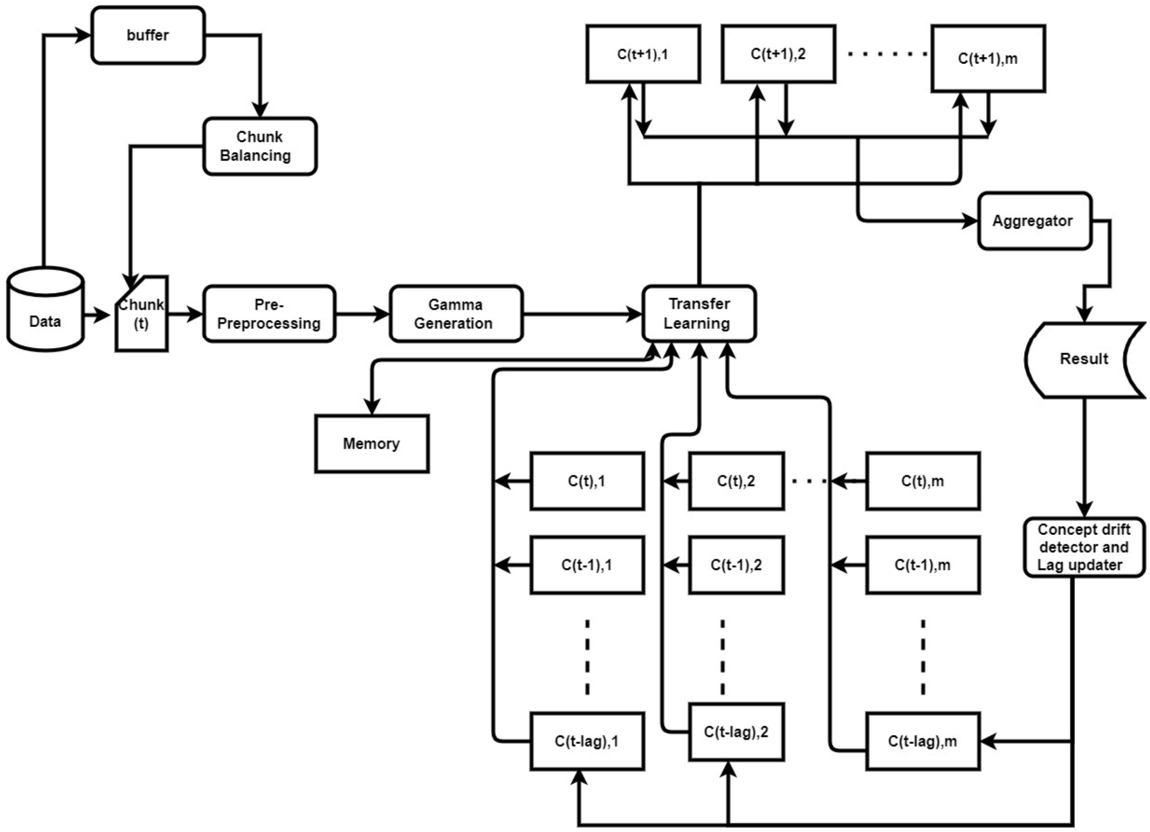

FIGURE 1. **Framework of transfer learning with lag, memory and classifiers aggregation for credit scoring (TLL).**

Algorithm 2 Pseudocode of Window-Based Balancing of Data

Input: Yp,TestingAccuracy(for each learner in every chunk)
Output: FinalOutput
1 **Start** 2 if mod(recordsCounter,windowSize) == 0 then
3 Compute ratio of each class in the chunk
4 maxNumberOfRecords = Get number of records of the maximum class ratio
5 helpingRecords = [NULL]
6 for each class do
7 numberOfNeededRecords = maxNumberOfRecords currentClassNumberOfRecords
8 helpingRecords = helpingRecords select random records from BufferscurrentClass
9 end
10 fixedChunk = fixedChunk; records(helpingRecords)
11 end
12 End

Algorithm 3 Pseudocode of Generating Gamma for Encoding Feature Availability

Input: featureSize,chunckCurrent
Output: Gamma
1 Start
2 Gamma=zeros(1,featureSize) 3 for each feature of chunkCurrent do
4 Gamma(feature)=1 5 end 6 return Gamma
7 End

of learners parameters, and novel genetic layered training (selection of learners) approaches used along with the cross-validation (CV). Their model is claimed to achieve accuracy of with a 29-layer structure is capable to achieve the prediction accuracy of 94.60% (54 errors per 1000 classifications) for the Statlog German. However, the model does not provide sequential learning capability and it has complex architecture. In the work of [7], five different feature selection algorithm with the technique if_any voting was applied and four different classification algorithms with soft voting was applied. In the work of [25], a model is on basis of five standard classifiers, namely LSVM, KNN, MDA, DT, LR, and adaptively selects the base classifiers with highest area under curve AUC according to the data distribution, then integrates all base classifiers to obtain a prediction.

The work of [26] heterogeneous ensemble model is based on the generalized Shapley value and the Choquet integral. To do this, the model first uses the fuzzy measure to express

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 Japanese PPDai GMSC
JoaquínAbellán et al [23]	×	√	×	×	Feature selection ,Classifier ensemble ,credal decision tree	Australian German Iranian ,Japanese Polish, UCSD
Hongliang He et al [24]	×	√	×	×	Classifier ensemble, EBCA ,PSO,RF, GBoost	Australian, Japanese, German, DefaultData, PPDaiData, LC2017Q1Data
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
SebastiánMaldonado et al [29]	×	×	×	×	three-way decision(Logistic Regression)	Chilean bank dataset
XiaodongFeng et al [16]	mean	√	×	×	Dynamic ensemble classification(DECSP)	Australian Germn,Japanese TaiwanDCCC Chinese1 ,2 ,AER Th02, PAKDD2010 Kaggle
SebastiánMaldonado et al [26]	$filter$	resampling	×	×	$l_1 l_\infty svm$, $l_2 l_\infty svm$ Feature selection	Chilean bank dataset
Leopoldo Melo Junior et al [20]	mean/mode	√	×	×	Dynamic selection , RMKnn	German, Default, PPDai, Iranian dataset, privat dataset, GiveMe, LC2015Q123
Diwakar Tripathi et al [21]	unique integer number	×	×	×	(EELM)	Australian, Japanese, German-categorical German-numerical
Pawel Plawiak et al [27]	×	√	×	×	DGHNL	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
WeiZhang et al. [30]	Filling	√	×	×	CSMIL	Chinese bank dataset
Xiaohong Chen Et Al.[26]	Mean	√	×	×	GSCI	Australian, Japanese German ,Default RRDai, ProsperLoan LendingClub
Our Framework	Transfer Learning	√	√	√	TLL-AD, TLL-ADWIN	German, Default PPDai, Lending club

the interactive characteristics between any two coalitions of base learners. A linear programming model for determining the fuzzy measure is built based on the accuracy and diversity objective function. The normal fuzzy number is employed to express the base learner predicted values to retain the original information as much as possible in the training stage. Then, the generalized Shapley Choquet integral (GSCI) aggregation operator is defined to calculate the comprehensive predicted value of the ensemble model. Based on the defined aggregation operator and linear programming model, a GSCI approach is proposed for ensemble credit scoring. However, this approach has not considered concept drift or any online update of the knowledge of the classifiers.

III. 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 pre-processing and normalization. Subsequently, gamma generation and transfer learning are described. Lastly, ensemble learning, concept drift, and lag update are presented.

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 $\perp$, indicating that label information of X_t is not available.

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

B. TRANSFER LEARNING WITH LAG 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 normalization. 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.

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.

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

Algorithm 4 Pseudocode of Transfer Learning TL

Input: previousNN,previousGamma,Memory,Gamma
Output: Memory,Learner

```
Start
for each feature do
  if (Gamma(feature)==1 and previousGamma
  (feature)==0) then
  if (findWeights(Memory)) then
     newWeights(feature)=restoreWeights(Memory)
   else
      newWeights(feature)=randInitiate();
  end
else if (Gamma(feature)==0 and previousGamma
(feature)==1) then
Memory(feature)=previousNNWeights(feature)
end
Learner.weights= newWeights
End
```

Algorithm 5 Pseudocode of Combination Between Learners

Input: Yp,TestingAccuracy(for each learner in every chunk)
Output: FinalOutput

```
Start
c = 1
for each chunk do
for each record do
 if the number of the individual classifiers is odd then
    The class of the record id predicted to be the majority
predictions among the individual classifiers
 else
    The class of the record is predicted to be the highest
testing accuracy predictions among the individual classifiers
 end
end
End
```

TABLE 2. 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
ΔAcc	0.5

TABLE 3. The Details of the used datasets.

Dataset	Default	German	PPDai	lending club
Record numbers	300,00	100,0	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

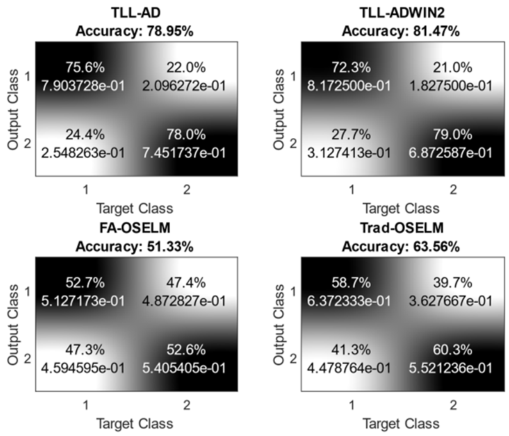

FIGURE 2. Lending club dataset confusion matrix.

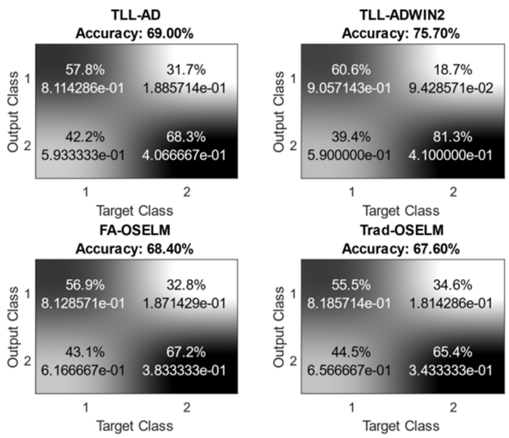

FIGURE 3. PPDai dataset confusion matrix.

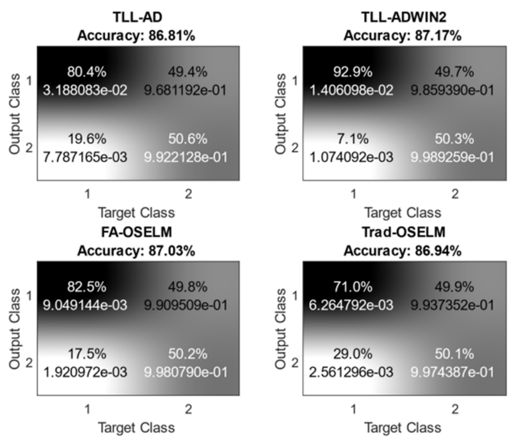

FIGURE 4. German dataset confusion matrix.

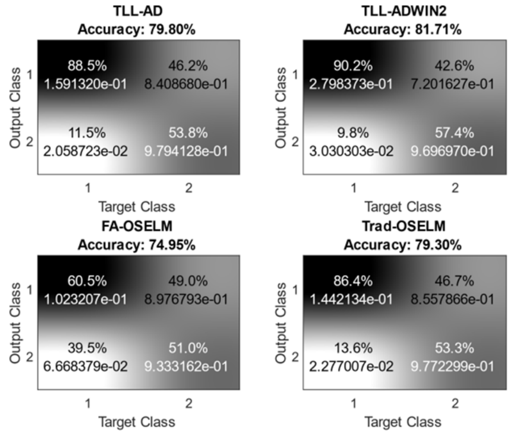

FIGURE 5. Default dataset confusion matrix.

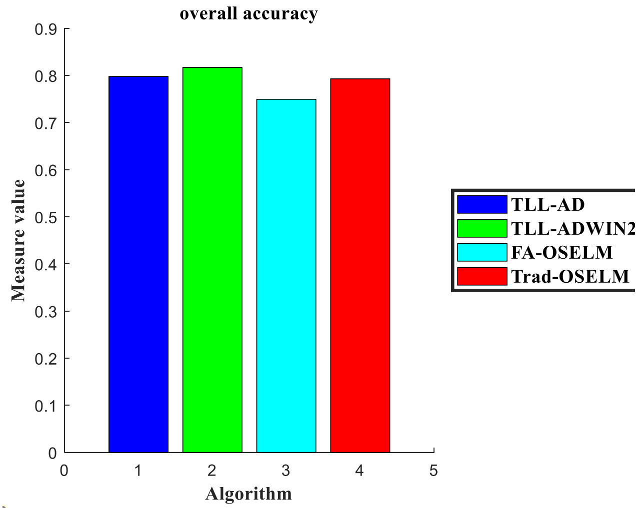

FIGURE 6. Accuracy result comparison of lending club dataset.

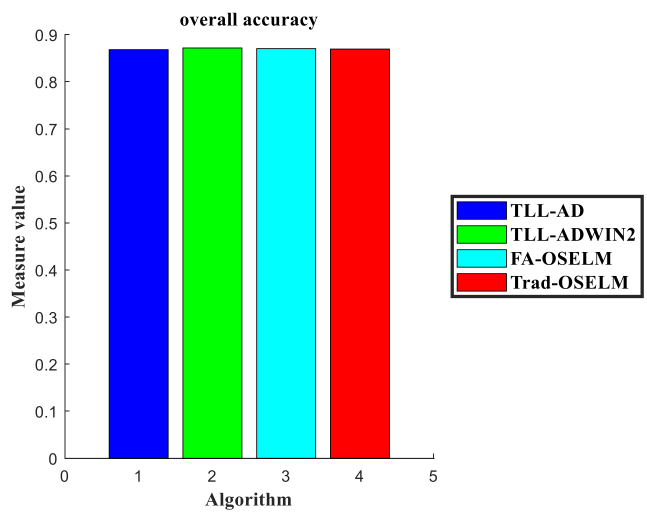

FIGURE 7. Accuracy result comparison of PPDai dataset.

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

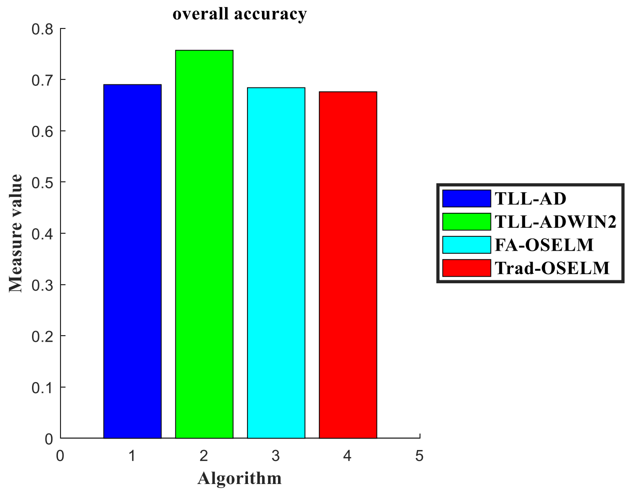

FIGURE 8. Accuracy result comparison of German dataset.

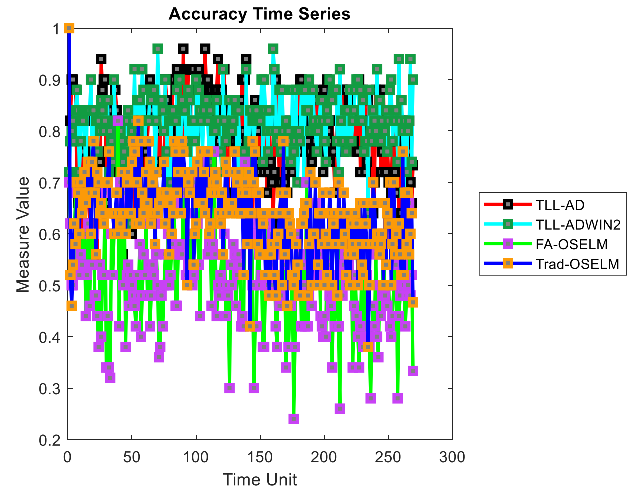

FIGURE 10. Accuracy club dataset time series comparison for Lending.

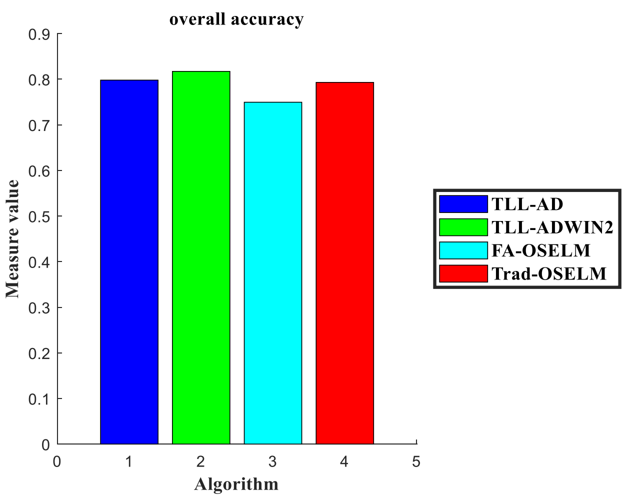

FIGURE 9. Accuracy result comparison of Default dataset.

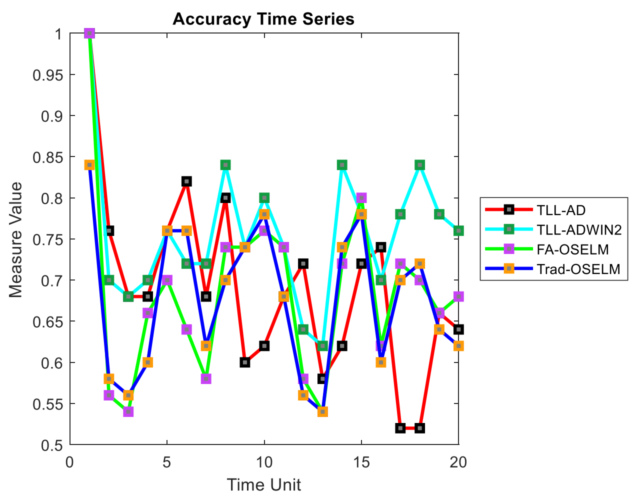

FIGURE 11. Accuracy time series comparison for German dataset.

class against the entire sample count and uses the majority ratio to complement minorities using data from the buffer.

D. PRE-PROCESSING AND NORMALIZATION

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:

$$x = 2\times\frac{x - \min(X_i)}{\max\,(X_i) - \min(min)} - 1 \qquad (1)$$

E. GAMMA GENERATION

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

$$D = \{C_0, C_1, ..C_t ... C_{N_c}\} \qquad (2)$$

where N_c denotes the number of chunks.

$$\bigcup_{foralli=0,1,\ldots,N} C_i = D$$

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 Γ 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 $\forall x_i \in \mathbf{\Gamma} \rightarrow x_i \in \{0, 1\}$.

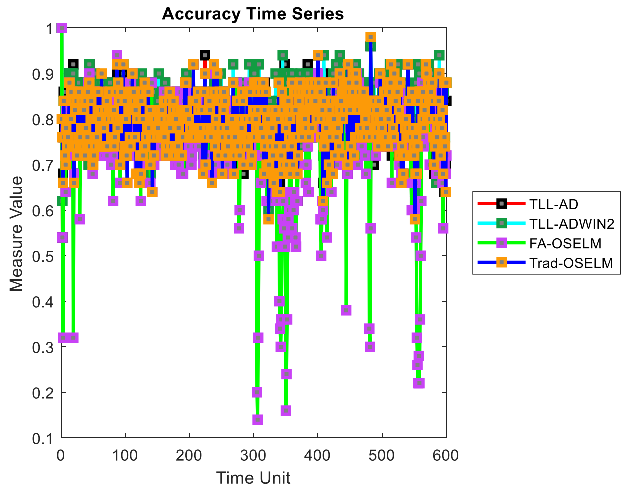

FIGURE 12. Accuracy time series comparison for Default dataset.

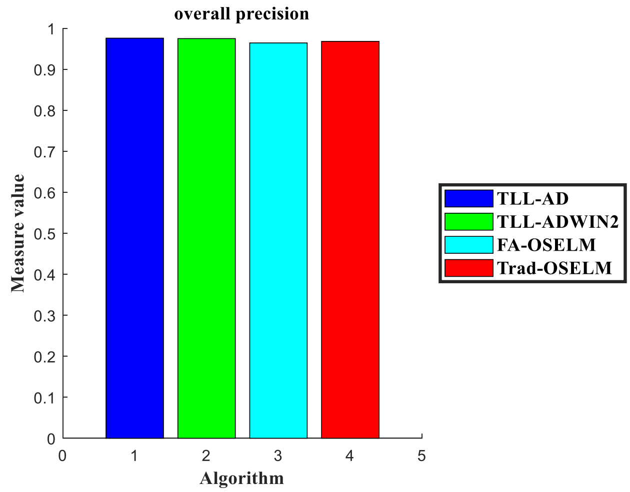

FIGURE 14. Overall precision comparison result of Lending Club dataset.

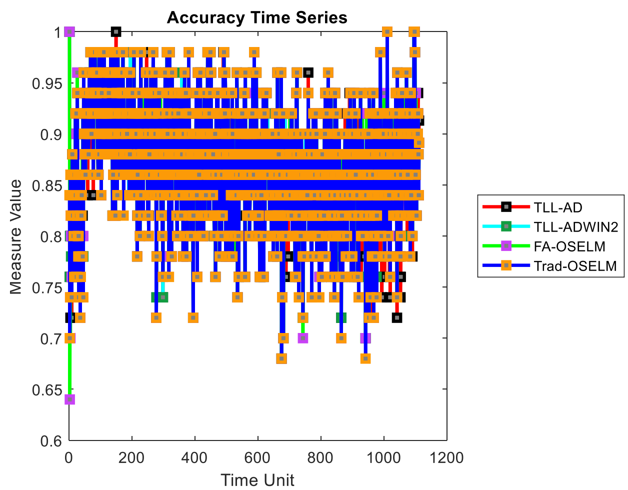

FIGURE 13. Accuracy time series comparison for PPDai dataset.

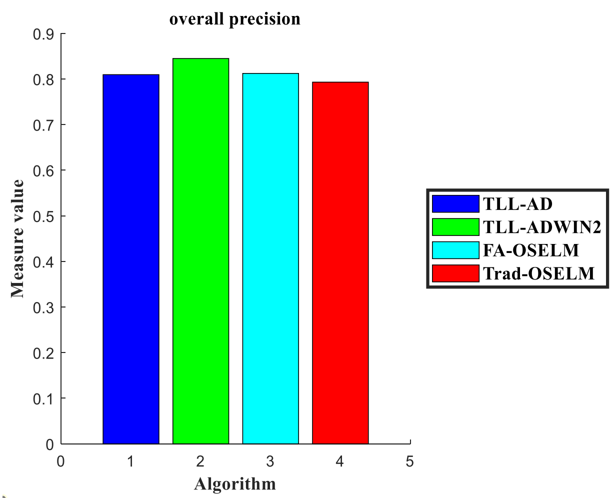

FIGURE 15. Overall precision comparison result of PPDai dataset.

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

$$D = \{ (X_i, Y_i), i = 0, 1, 2, \ldots N_c \} \tag{3}$$

s.t

$$\forall x, y \in C_i, \Gamma(x) = \Gamma(y) \tag{4}$$

where Γ(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.

$$\Delta Acc = Acc_{t-1} - Acc_t > threshold \tag{5}$$

where

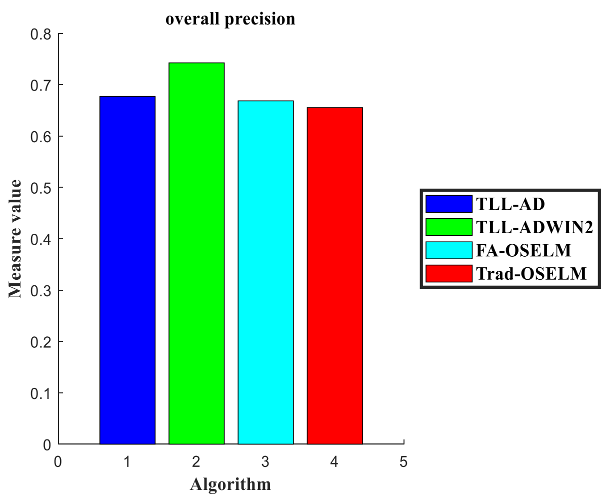

FIGURE 16. Overall precision comparison result of German data.

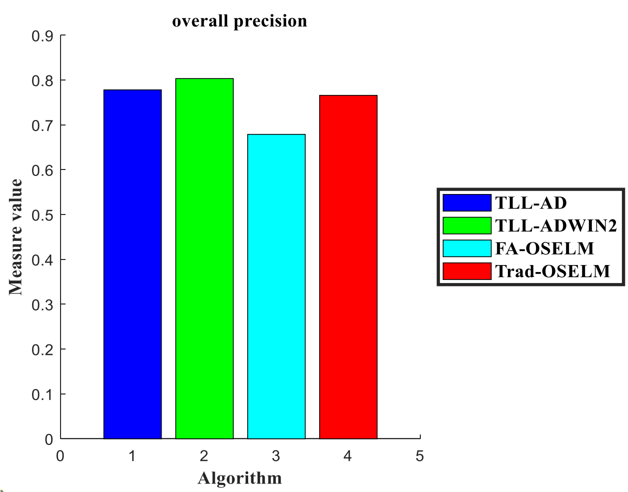

FIGURE 17. Overall precision comparison result of Default dataset.

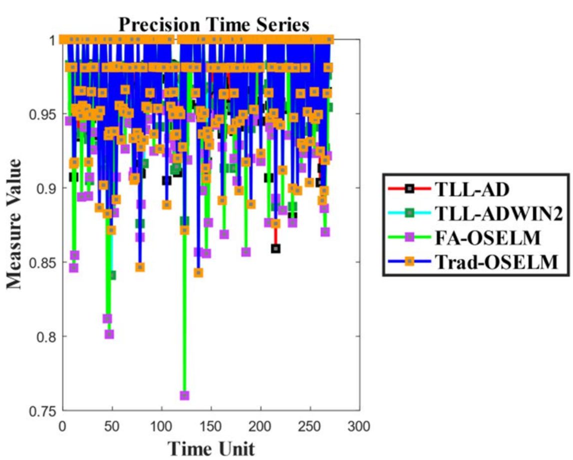

FIGURE 18. Precision time series comparison for Lending club dataset.

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

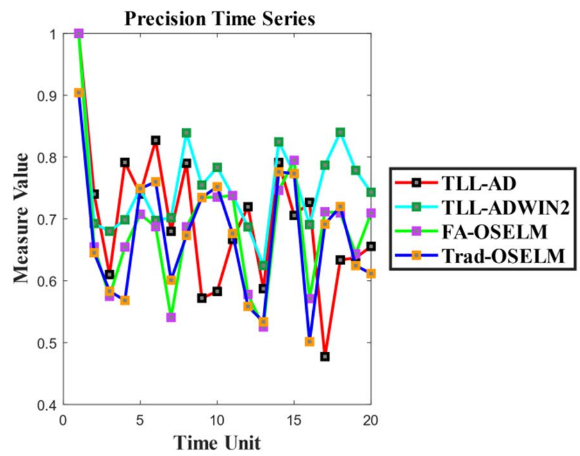

FIGURE 19. Precision time series comparison for German dataset.

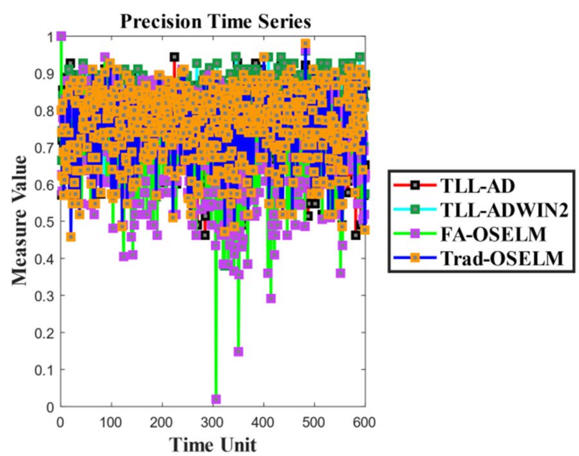

FIGURE 20. Precision time series comparison for default dataset.

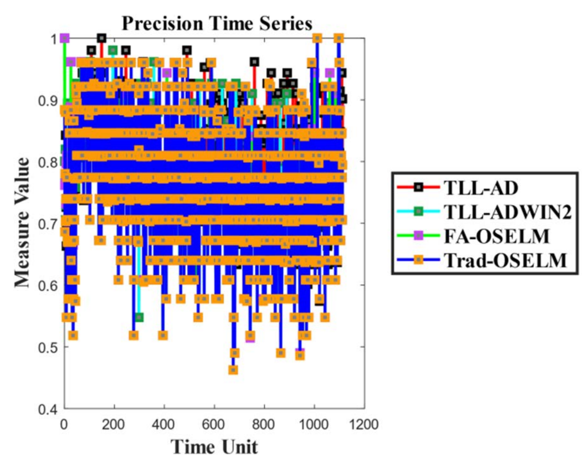

FIGURE 21. Precision time series comparison for PPDai dataset.

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

TABLE 4. 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

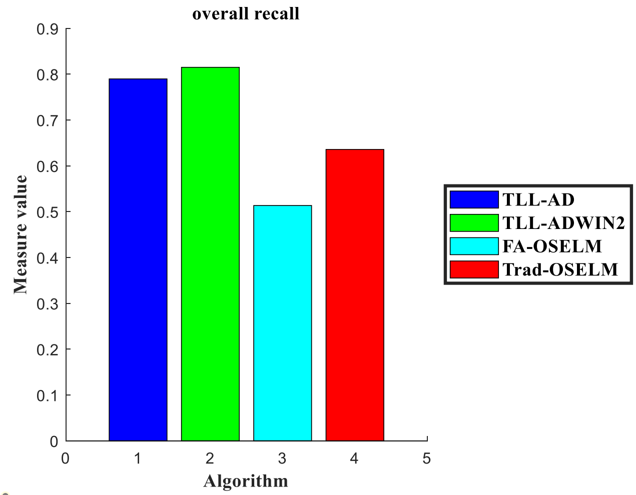

FIGURE 22. Overall recall comparison result of Lending Club dataset.

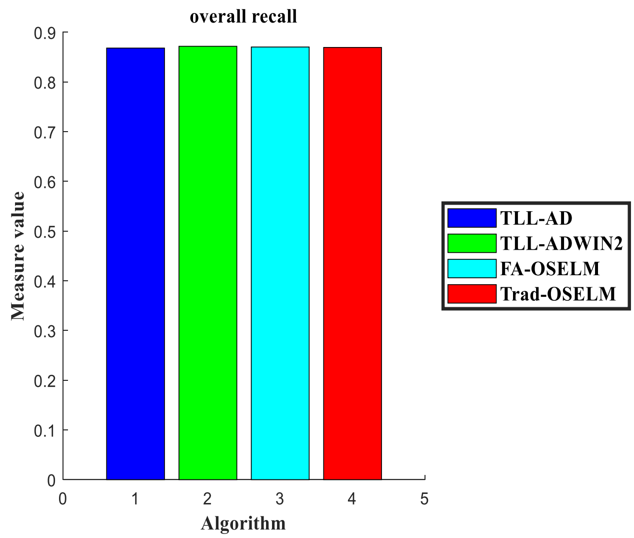

FIGURE 23. Overall recall comparison result of PPDai dataset.

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

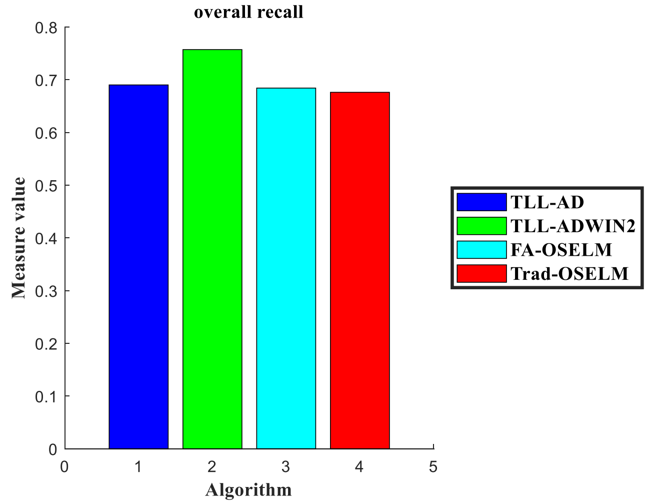

FIGURE 24. Overall recall comparison result of German dataset.

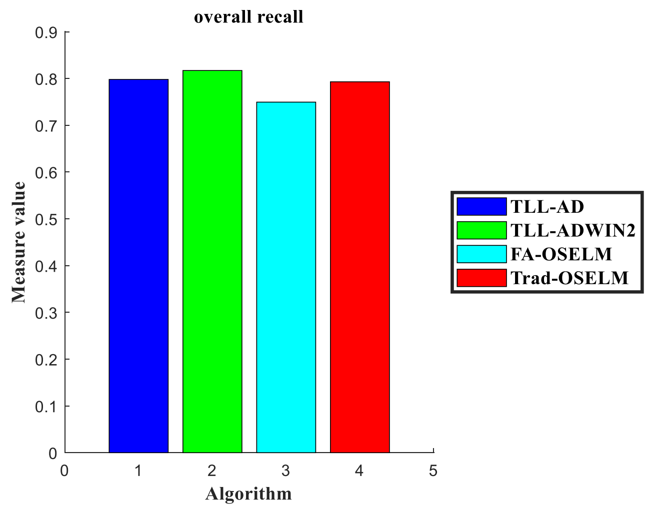

FIGURE 25. Overall recall comparison result of default dataset.

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

$$Lag(t) = \begin{cases} Lag(t-1) - 1 & \textit{if concept drift exists} \\ Lag(t) & \textit{otherwise} \end{cases}$$
$$Lag(0) = Lag_0 \tag{6}$$

where

$Lag\,(t)$ denotes the lag at moment t.

Lag_0 denotes the initial *lag*.

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

IV. MEMORY FREEING

Memory freeing cleans the memory of the weights related to outdated classifiers identified by $Lag\,(t)$. Hence, for every moment t, the free Memory process is supposed to remove all weights related to classifiers trained at the moment

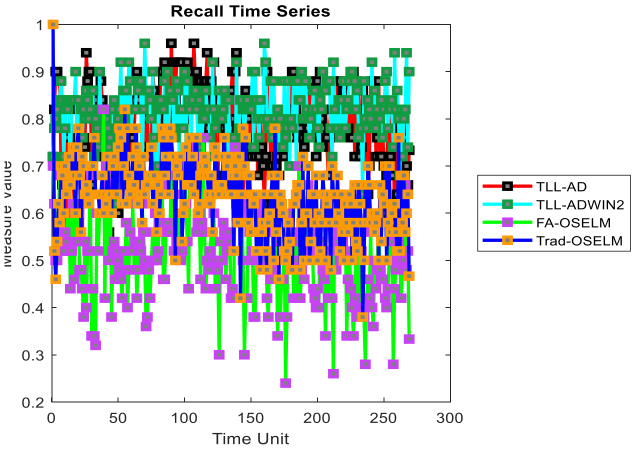

FIGURE 26. Recall time series comparison for lending club dataset.

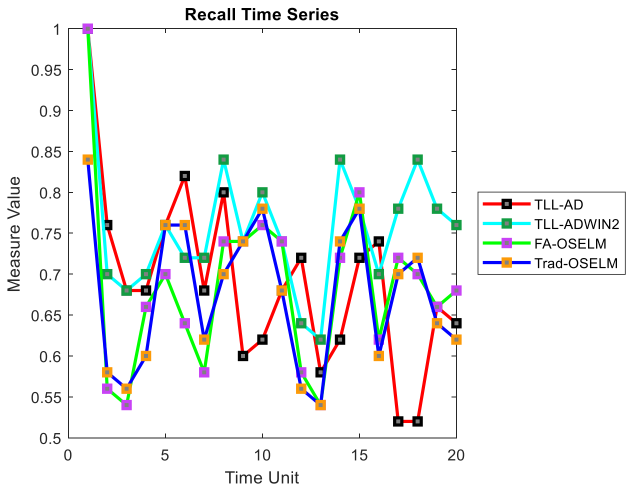

FIGURE 27. Recall time series comparison for german dataset.

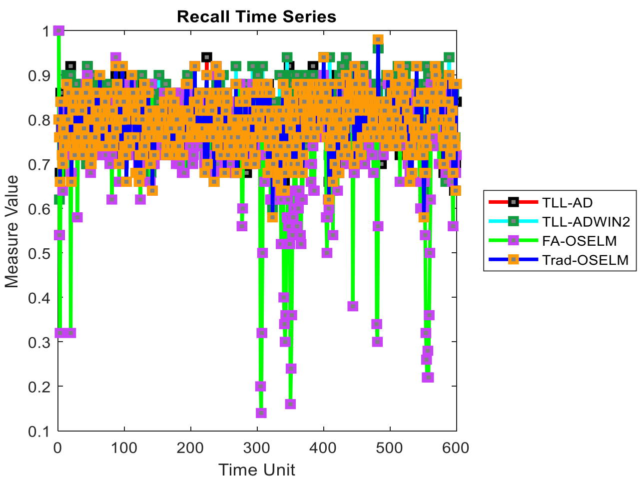

FIGURE 28. Recall time series comparison for default dataset.

$t - Lag\,(t)$ or older. This process ensures that older weights are not restored from memory and expired knowledge is not considered.

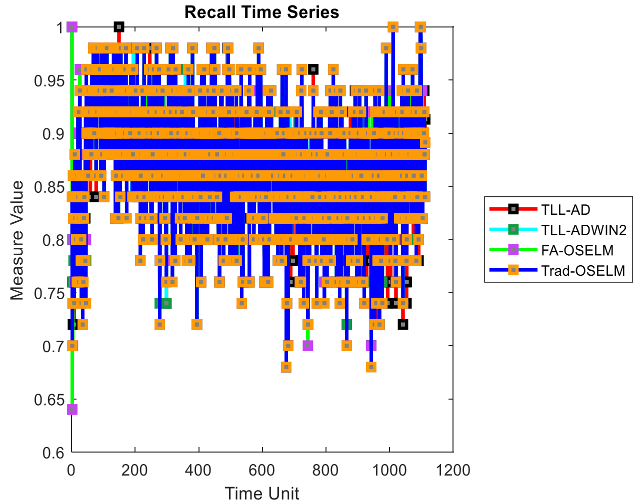

FIGURE 29. Recall time series comparison for PPDai dataset.

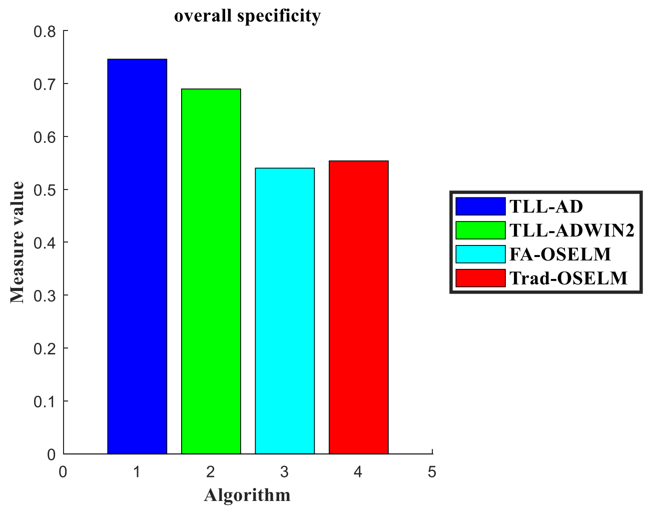

FIGURE 30. Overall specificity comparison result of lending club data.

V. 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 ΔAcc.

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.

A. 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

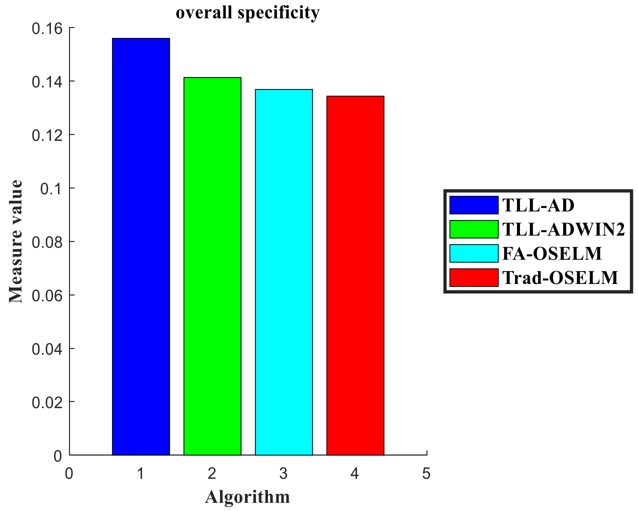

FIGURE 31. Overall specificity comparison result of PPDai data.

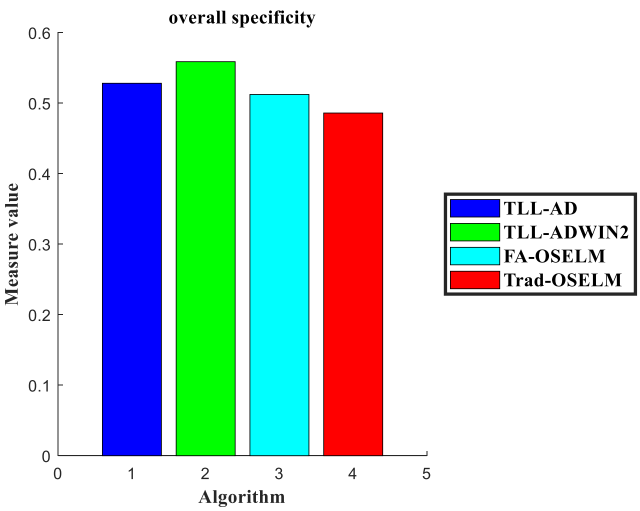

FIGURE 32. Overall specificity comparison result of german data.

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.

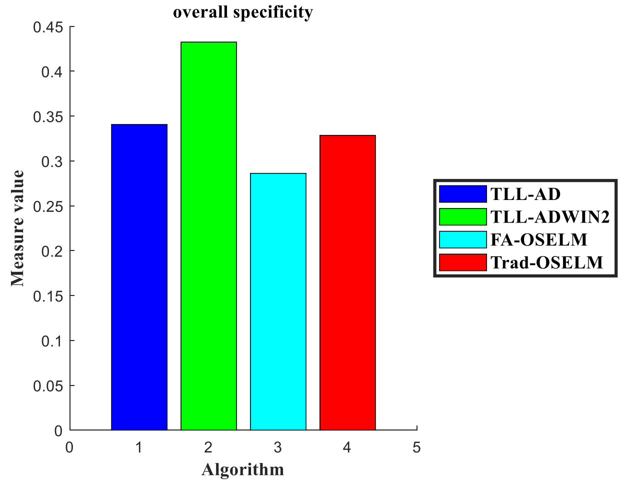

FIGURE 33. Overall specificity comparison result of default data.

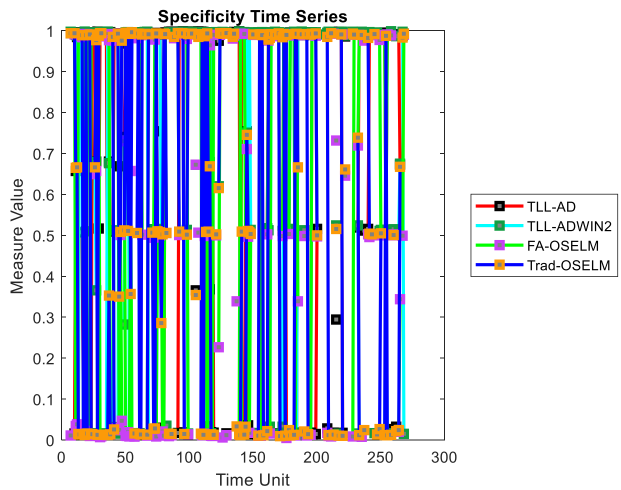

FIGURE 34. Specificity time series comparison for lending club dataset.

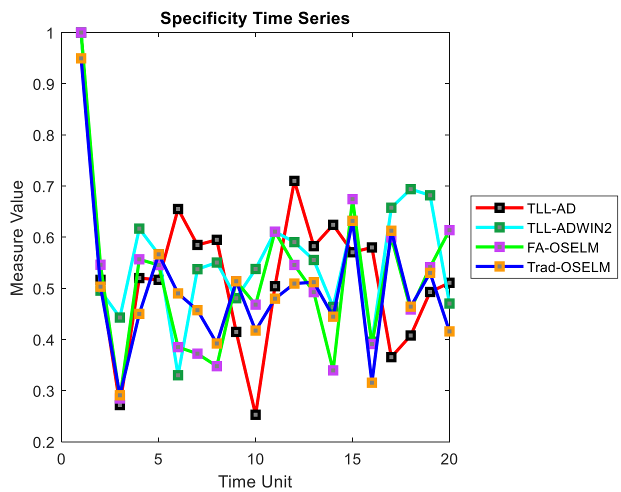

FIGURE 35. Specificity time series comparison for german dataset.

B. ACCURACY

Figures 6-9 depict overall accuracy for Lending Club, German, default, and PPDai datasets, respectively. The

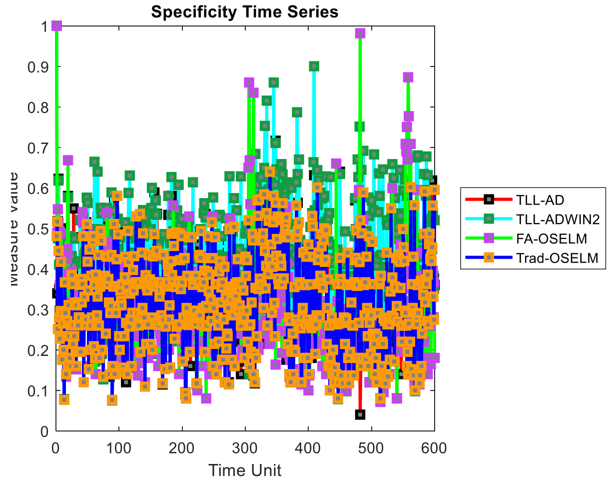

FIGURE 36. Specificity time series comparison result for default dataset.

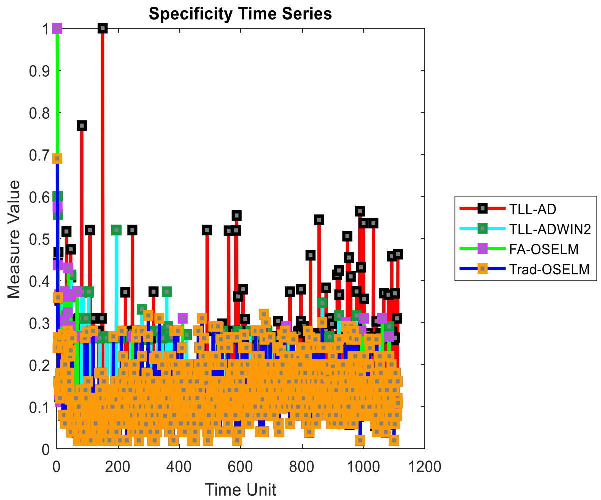

FIGURE 37. Specificity time series comparison result for PPDai dataset.

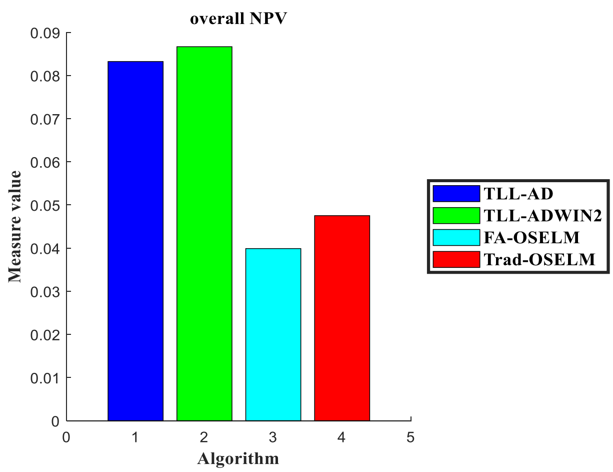

FIGURE 38. Overall NPV comparison result of lending club dataset.

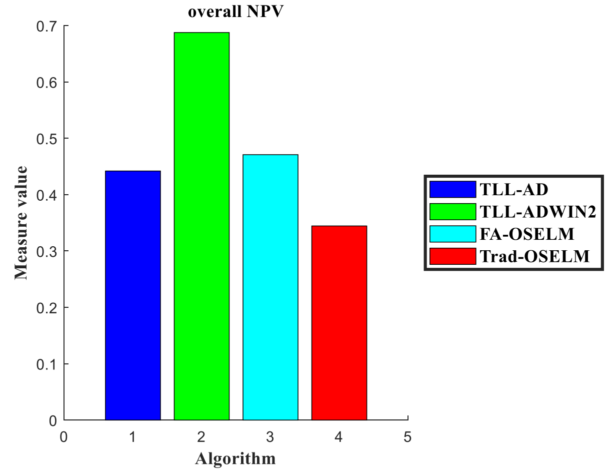

FIGURE 39. Overall NPV comparison result of PPDai dataset.

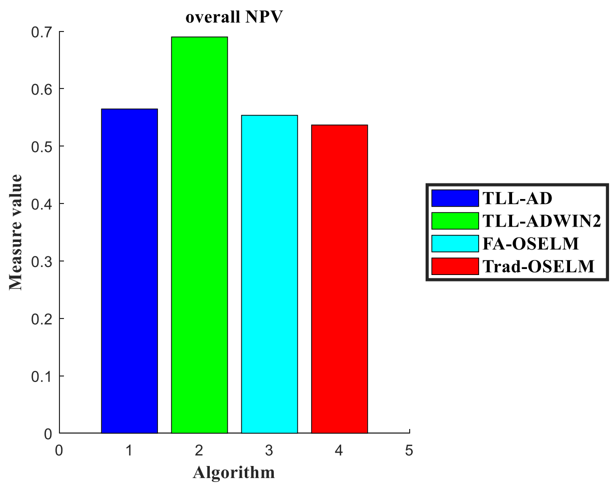

FIGURE 40. Overall NPV comparison result of german dataset.

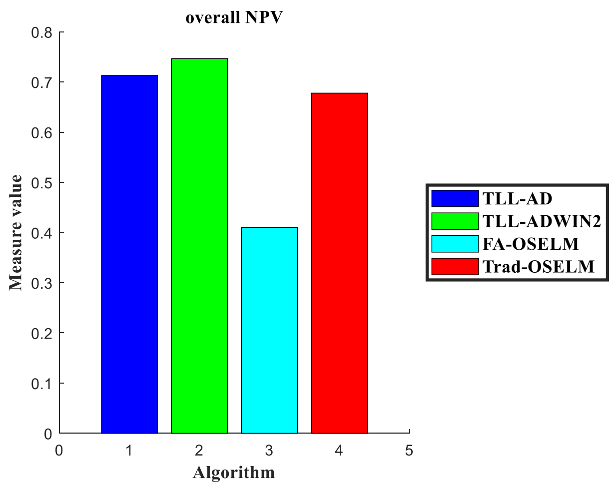

FIGURE 41. Overall NPV comparison result of default dataset.

graphs for all datasets indicate that TLL outperformed both FA-OSELM and OSLEM in terms of the reached accuracy. Also, TLL-ADWIN2 reached an accuracy close to 80% for three datasets, namely, Lending Club, German, and

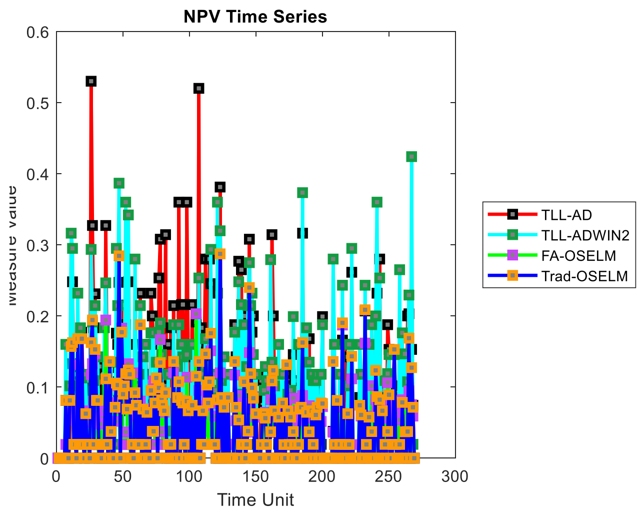

FIGURE 42. NPV time series comparison for lending Club dataset.

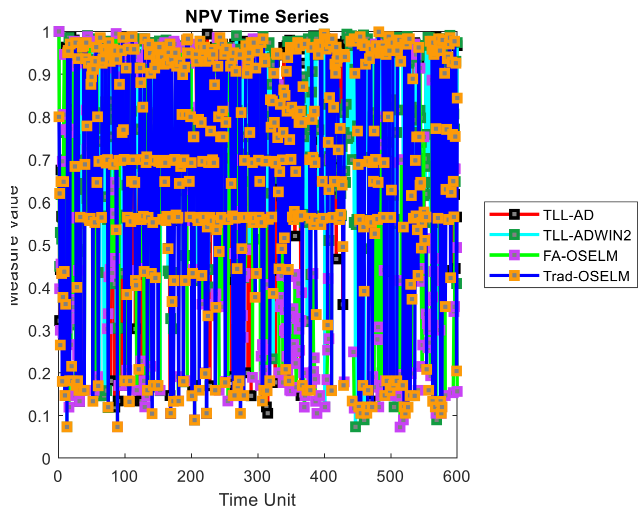

FIGURE 44. NPV time series comparison for default dataset.

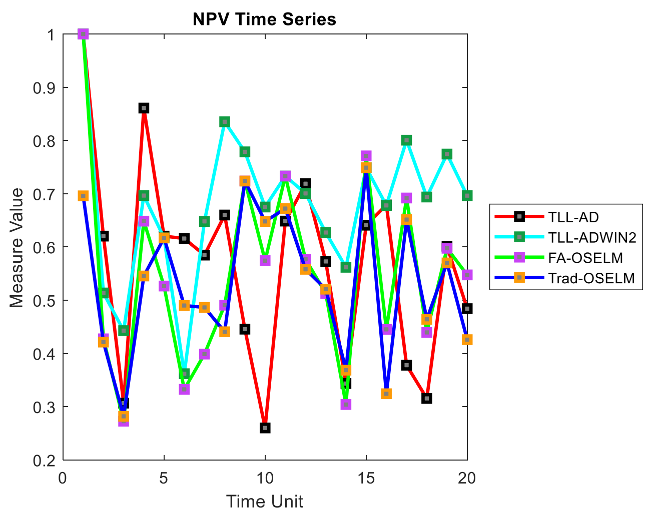

FIGURE 43. NPV time series comparison for german dataset.

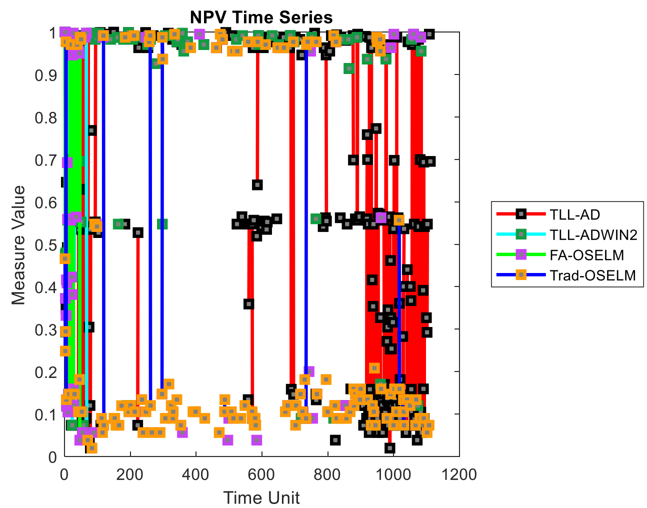

FIGURE 45. NPV time series comparison for PPDai dataset.

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-OSLEM 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 10-13. 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 indicate that TLL methods are generally superior to FA and OSELM.

C. OVERALL PRECISION

Precision indicates the percentage of actual positive predicted values against all positive predicted values, as depicted in Figure 14-17. 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 16 indicates that TLL-ADWIN2 obtained a precision of about 86%, compared to slightly lower values for FA

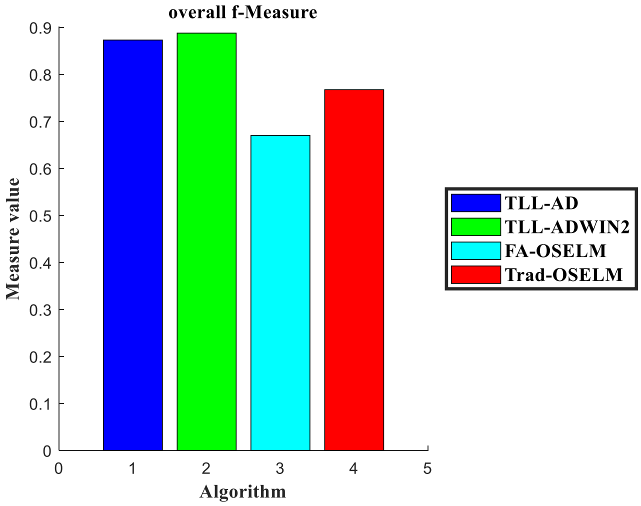

FIGURE 46. Overall F-Measure time series comparison for lending club dataset.

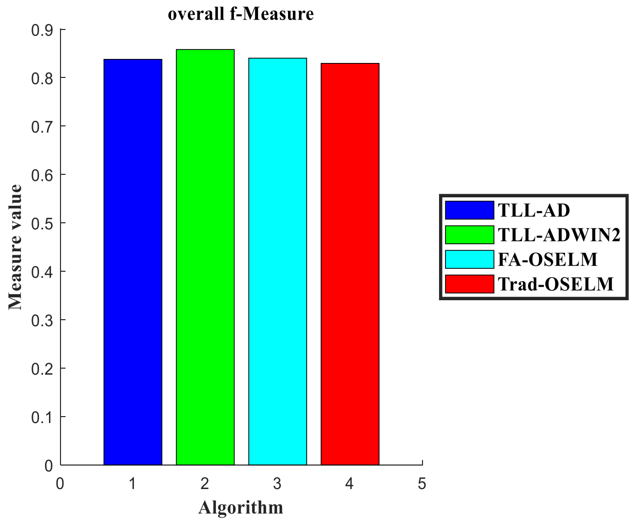

FIGURE 47. Overall F-Measure time series comparison for PPDai dataset.

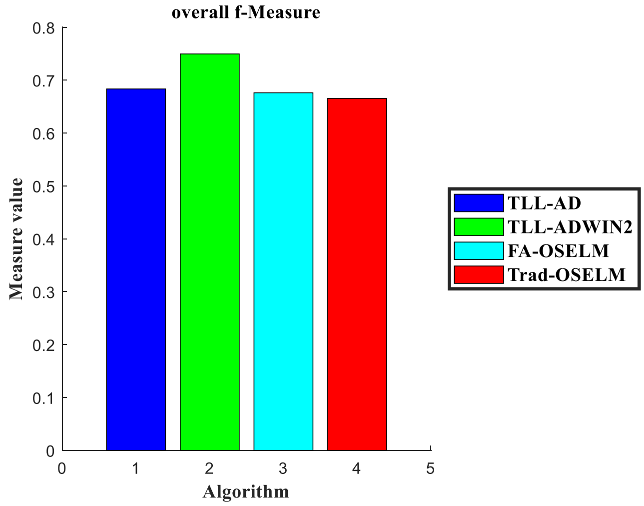

FIGURE 48. Overall F-Measure time series comparison for german dataset.

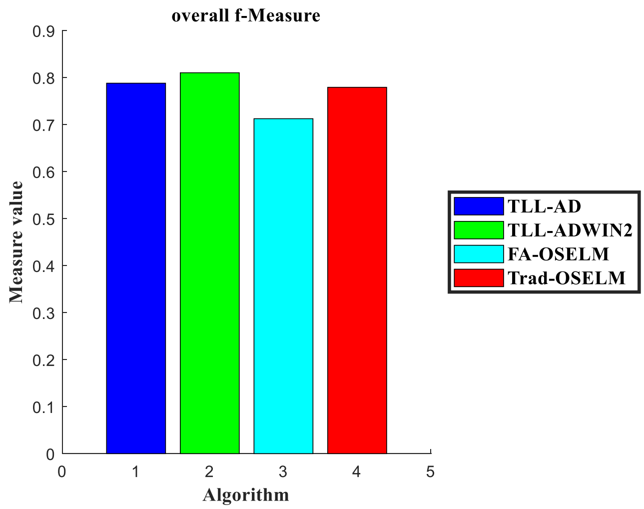

FIGURE 49. Overall F-Measure time series comparison for Default dataset.

and OSELM. Like accuracy, we present the time series for precision using separate plots, depicted using Figures 18-21.

We elaborate performance further by presenting detailed time series data using Figure 18-21. 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.

D. OVERALL RECALL

Recall provides the percentage of actual predicted positive records from all actual positive records, depicted using Figures 22-25. 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 26-27 depict the time series plots for all methods and datasets to elaborate recall performance.

E. OVERALL SPECIFICITY

Overall specificity provides the percentage of correctly identified negatives, illustrated using Figures 30-33. 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.

F. OVERALL NPV

Negative predictive values are used to measure the accuracy of a negative test result. The results are illustrated in

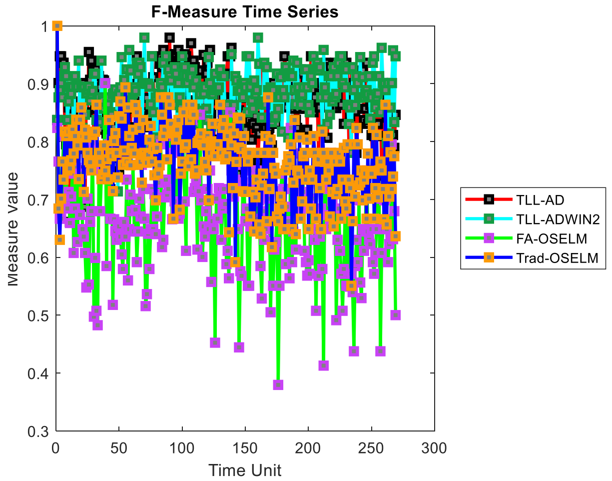

FIGURE 50. F-measure time series comparison for lending club dataset.

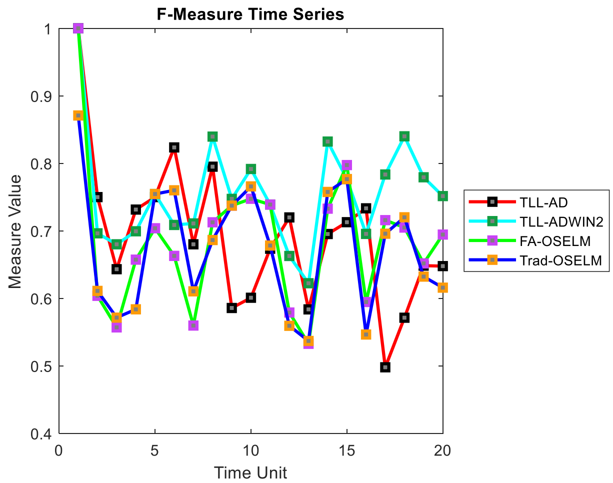

FIGURE 51. F-measure time series comparison for german dataset.

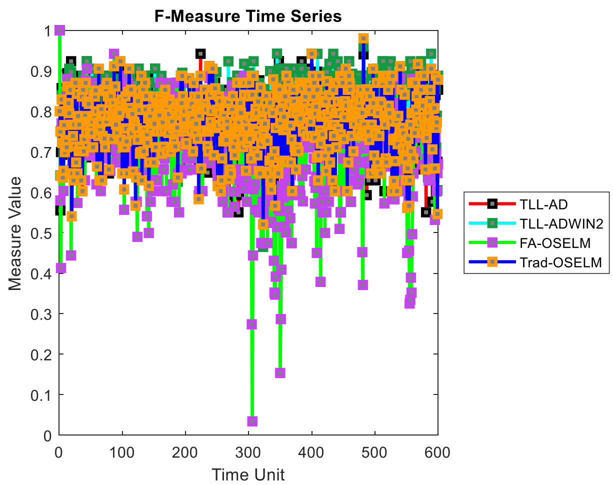

FIGURE 52. F-measure time series comparison for default dataset.

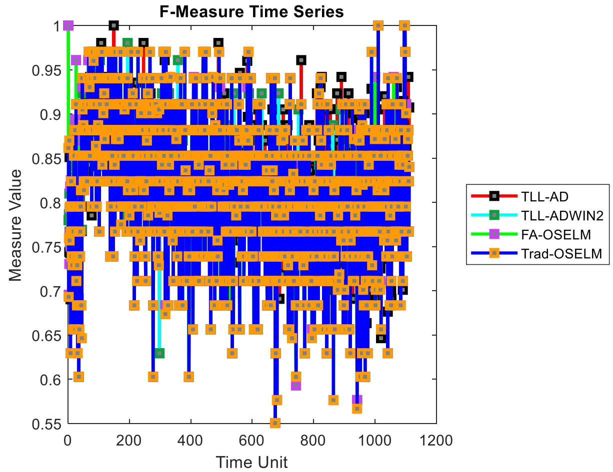

FIGURE 53. F-measure time series comparison for PPDai dataset.

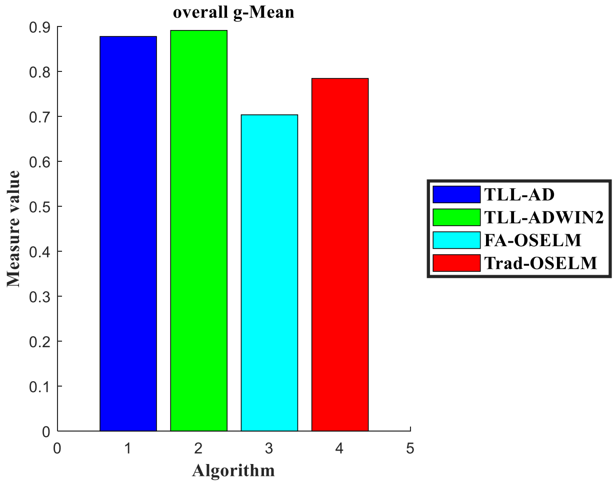

FIGURE 54. Overall G-Mean result comparison for lending club dataset.

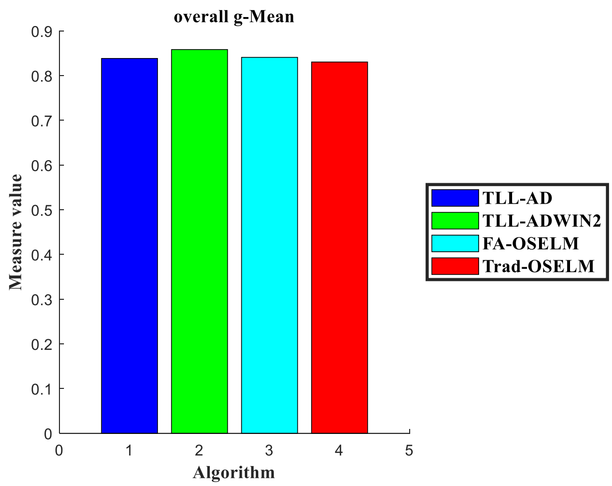

FIGURE 55. Overall G-Mean result comparison for PPDai dataset.

Figures 38-41. 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

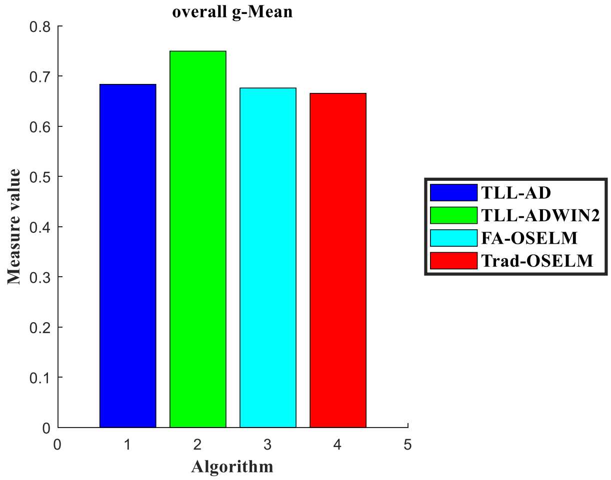

FIGURE 56. Overall G-Mean result comparison for german dataset.

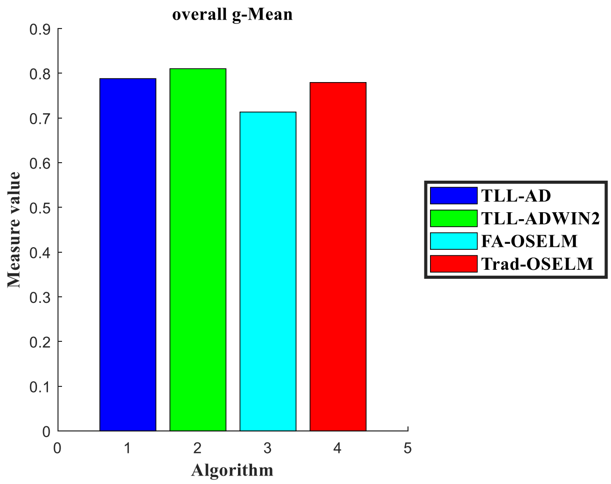

FIGURE 57. Overall G-Mean result comparison for default dataset.

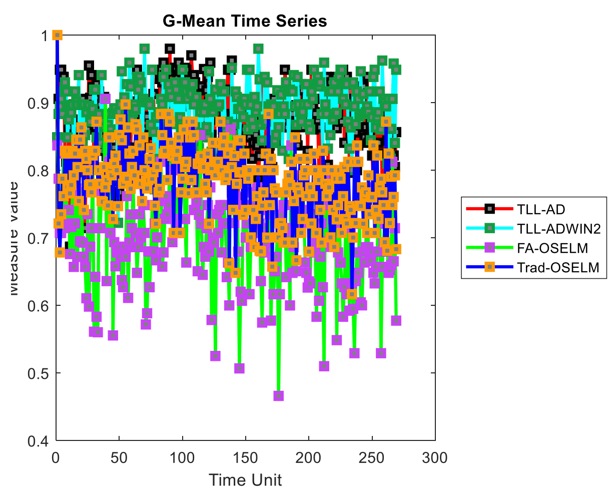

FIGURE 58. G-Mean time series comparison for lending club dataset.

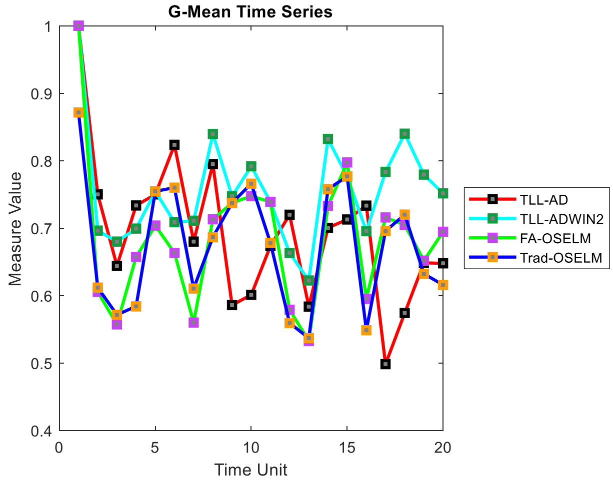

FIGURE 59. G-Mean time series comparison for german dataset.

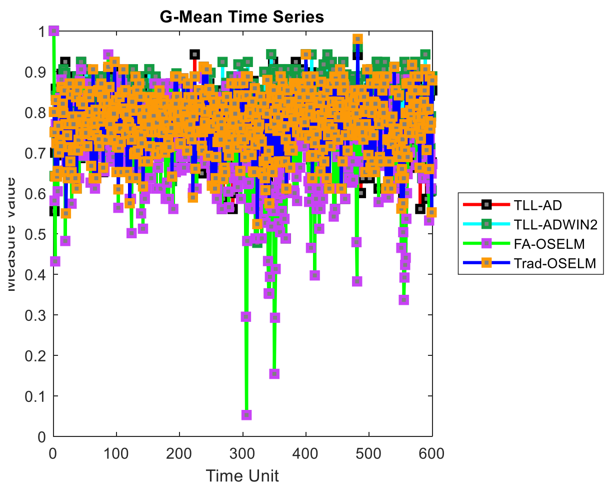

FIGURE 60. G-Mean time series comparison for default dataset.

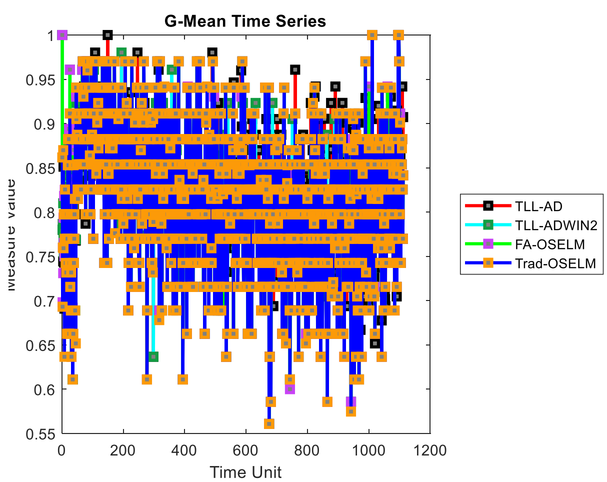

FIGURE 61. G-Mean time series comparison for PPDai dataset.

OSELM. It indicates that FA and OSELM have relatively poor performance concerning NPV. Figures 42-45 present the time series for all methods and datasets to depict overall NPV performance.

TABLE 5. The numerical results of all metrics with AUC.

	ALGORITHM	Accuracy	Recall	Precision	G-Mean	F-Measure	Specificity	NPV	AUC
LENDING CLUB	TLL-AD	**0.7676**	0.7528	0.5261	0.5410	0.4877	0.7528	0.5261	**0.73**
	TLL-ADWIN	**0.8147**	0.7523	0.5307	0.5591	0.5108	0.7523	0.5307	**0.75**
	FA-OSELM	0.5133	0.5266	0.5020	0.4086	0.3575	0.5266	0.5020	0.52
	TRAD-OSELM	0.6426	0.5358	0.5030	0.4482	0.4120	0.5358	0.5030	0.51
PPDAI DATA	ALGORITHM	Accuracy	Recall	Precision	G-Mean	F-Measure	Specificity	NPV	AUC
	TLL-AD	0.8681	0.5120	0.6257	0.5204	0.4940	0.5120	0.6257	**0.51**
	TLL-ADWIN	0.**8717**	0.5065	0.7662	0.5149	0.4794	0.5065	0.7662	**0.51**
	FA-OSELM	0.8703	0.5036	0.6415	0.4969	0.4741	0.5036	0.6415	0.50
	TRAD-OSELM	0.8694	0.5019	0.5687	0.4865	0.4712	0.5019	0.5687	0.50
DEFAULT DATA	ALGORITHM	Accuracy	recall	precision	G-mean	F-measure	Specificity	NPV	AUC
	TLL-AD	0.8065	0.5917	0.7649	0.6379	0.6039	0.5917	0.7649	0.59
	TLL-ADWIN	**0.8171**	0.6248	0.7749	0.6725	0.6478	0.6248	0.7649	**0.62**
	FA-OSELM	0.7623	0.5290	0.5817	0.5341	0.5159	0.5290	0.5817	0.52
	TRAD-OSELM	0.8064	0.6065	0.7433	0.6479	0.6238	0.6065	0.7433	**0.60**
GERMAN DATA	ALGORITHM	Accuracy	Recall	Precision	G-Mean	F-Measure	Specificity	NPV	AUC
	TLL-AD	0.6390	0.5802	0.5773	0.5786	0.5784	0.5802	0.5773	0.58
	TLL-ADWIN	**0.7570**	0.6579	0.7163	0.6790	0.6711	0.6579	0.7163	**0.65**
	FA-OSELM	0.7310	0.6345	0.6733	0.6488	0.6437	0.6345	0.6733	0.63
	TRAD-OSELM	0.7110	0.6450	0.6520	0.6483	0.6480	0.6450	0.6520	0.64

G. 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 47-49. 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.

H. OVERALL G-MEAN

G-Mean measures the balance between classification performance for both majority and minority classes. The results are illustrated in Figures 54-57. It is evident that the proposed TLL approach outperformed both FA-OSELM and OSLEM in terms of the reached G-mean. It also indicates the poor performance of both FA and OSELM compared to TLL. Figures 58-61 present the time series of all methods and datasets to elaborate overall specificity performance.

In order to summarize the accomplished performance, we present the numerical results of all metrics with AUC in Table 5 and the improvement percentage with respect to accuracy. The table compares datasets, namely, German, Lending Club, default, and PPDai. As depicted in the table, our approach has improved overall benchmarks, with 53% and 65% being the best improvements over OSELM and FA-OSELM.

VI. CONCLUSION AND FUTURE WORK

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. The role of transfer learning is to transfer the weights that are associated with currently non-active features to memory and to restore the weights from the memory when a new feature is active. 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. For chunk balancing, 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. Future work is to extend the developed algorithm to include optimization of the random weights between the input-hidden layer algorithm and to incorporate dynamic feature selection.

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, Jul. 2018.

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

[3] A. Levy and R. Baha, "Credit risk assessment: A comparison of the performances of the linear discriminant analysis and the logistic regression," *Int. J. Entrepreneurship Small Bus.*, vol. 42, nos. 1–2, pp. 169–186, 2021.

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

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

[7] J. Nalić, G. Martinović, and D. Žagar, "New hybrid data mining model for credit scoring based on feature selection algorithm and ensemble classifiers," *Adv. Eng. Informat.*, vol. 45, Aug. 2020, Art. no. 101130.

[8] H. Sutrisno and S. Halim, "Credit scoring refinement using optimized logistic regression," in *Proc. Int. Conf. Soft Comput., Intell. Syst. Inf. Technol. (ICSIIT)*, Sep. 2017, pp. 26–31.

[9] G. Bennouna and M. Tkiouat, "Fuzzy logic approach applied to credit scoring for microfinance in Morocco," *Proc. Comput. Sci.*, vol. 127, pp. 274–283, Jan. 2018.

[10] C.-S. Ong, J.-J. Huang, and G.-H. Tzeng, "Building credit scoring models using genetic programming," *Expert Syst. Appl.*, vol. 29, pp. 41–47, Jul. 2005.

[11] K. Tran, T. Duong, and Q. Ho, "Credit scoring model: A combination of genetic programming and deep learning," in *Proc. Future Technol. Conf. (FTC)*, Dec. 2016, pp. 145–149.

[12] G. Teles, J. J. P. C. Rodrigues, R. A. L. Rabê, and S. A. Kozlov, "Artificial neural network and Bayesian network models for credit risk prediction," *J. Artif. Intell. Syst.*, vol. 2, no. 1, pp. 118–132, 2020.

[13] B. Zhu, W. Yang, H. Wang, and Y. Yuan, "A hybrid deep learning model for consumer credit scoring," in *Proc. Int. Conf. Artif. Intell. Big Data (ICAIBD)*, May 2018, pp. 205–208.

[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, May 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, Dec. 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, Apr. 2018.

[17] V.-S. Ha and H.-N. Nguyen, "Credit scoring with a feature selection approach based deep learning," in *Proc. MATEC Web Conf.*, vol. 54, 2016, p. 05004.

[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, Sep. 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, Jan. 2019.

[20] L. M. 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, Aug. 2020, Art. no. 113351.

[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, Nov. 2020, Art. no. 103980.

[22] P. Pławiak, M. Abdar, and U. R. Acharya, "Application of new deep genetic cascade ensemble of SVM classifiers to predict the Australian credit scoring," *Appl. Soft Comput.*, vol. 84, Nov. 2019, Art. no. 105740.

[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, May 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, May 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. Finance Econ.*, vol. 26, no. 3, pp. 4372–4385, Jul. 2021.

[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. Pławiak, M. Abdar, J. Pławiak, V. Makarenkov, and U. R. Acharya, "DGHNL: A new deep genetic hierarchical network of learners for prediction of credit scoring," *Inf. Sci.*, vol. 516, pp. 401–418, Apr. 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, Oct. 2020, Art. no. 113366.

[29] S. Maldonado, G. Peters, and R. Weber, "Credit scoring using three-way decisions with probabilistic rough sets," *Inf. Sci.*, vol. 507, pp. 700–714, Jan. 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, Nov. 2020, Art. no. 113489.

[31] J. Nalić and G. Martinovic, "Building a credit scoring model based on data mining approaches," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 30, no. 2, pp. 147–169, Feb. 2020.

[32] F. Shen, X. Zhao, G. Kou, and F. E. Alsaadi, "A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique," *Appl. Soft Comput.*, vol. 98, Jan. 2021, Art. no. 106852.

[33] Z. Li, Y. Tian, K. Li, F. Zhou, and W. Yang, "Reject inference in credit scoring using semi-supervised support vector machines," *Expert Syst. Appl.*, vol. 74, pp. 105–114, May 2017.

[34] M. Ala'raj and M. F. Abbod, "Classifiers consensus system approach for credit scoring," *Knowl.-Based Syst.*, vol. 104, pp. 89–105, Jul. 2016.