A Pruning Optimized Fast Learn++NSE Algorithm

YONG CHEN1,4, YUQUAN ZHU1, HAIFENG CHEN2, YAN SHEN3, AND ZHAO XU4

1School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, Jiangsu 212013, China
2Unicom Internet of Things Co., Ltd., Nanjing, Jiangsu 210014, China
3Department of Information Management and Information System, Jiangsu University, Zhenjiang, Jiangsu 212013, China
4Nanjing Longyuan Microelectronics Technology Co., Ltd., Nanjing, Jiangsu 211106, China

Corresponding author: Yuquan Zhu (yqzhu@ujs.edu.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant 61702229, in part by the National Statistical Science Research Project of China under Grant 2016LY17, in part by the Nature Science Foundation of Jiangsu Province of China under Grant BK20150531, in part by the Key R&D Project of Jiangsu Province of China under Grant SBE2018310252, in part by the National Key New Product Plan of the Ministry of Science and Technology of China under Grant 2014GRC11603, in part by the SME Innovation Fund of the Ministry of Science and Technology of China under Grant 14C26213201047, in part by the Technology Support Plan of Jiangsu Province of China under Grant BE20140424, and in part by the Information Industry Transformation and Upgrading Fund Project of Jiangsu Province of China under Grant 2014GRC11602.

ABSTRACT Due to the large number of typical applications, it is very important and urgent to study the fast classification learning of accumulated big data in nonstationary environments. The newly proposed algorithm, named Learn++.NSE, is one of the important research results in this research field. And a pruning version, named Learn++.NSE-Error-based, was given for accumulated big data to improve the learning efficiency. However, the studies have found that the Learn++.NSE-Error-based algorithm often encounters a situation that the newly generated base classifier is pruned in the next integration, which reduces the accuracy of the ensemble classifier. The newly generated base classifier is very important in the next ensemble learning and should be retained. Therefore, the two latest base classifiers are retained without being pruned, and a new pruning algorithm named NewLearn++.NSE-Error-based was proposed. The experimental results on the generated dataset and the real-world dataset show that NewLearn++.NSE-Error-based can further improve the accuracy of the ensemble classifier under the premise of obtaining the same time complexity as Learn++.NSE algorithm. It is suitable for fast classification learning of long-term accumulated big data.

INDEX TERMS Ensemble learning, nonstationary environment, classification algorithm, big data mining

I. INTRODUCTION

With the popularity of mobile network, smart devices, Internet of Things and the development of cross-platform technology, various application systems built on many different platforms have accumulated large-scale data sets in the form of data streams. The big data is gradually accumulated in the form of time-efficient data streams, which is characterized by the fact that the probability distribution of data generation is generally changing and unstable. This produces lots of important research problems...
about efficient classification learning of big data that is gradually accumulated in the nonstationary environments.

Due to the urgent needs of many typical applications, the method of efficient classification learning of big data, which is continuously accumulated in nonstationary environments, has become one of the focuses and hotspots in the research field of data mining [1-7]. For example, in the user comment analysis system of the e-commerce platform, the classification model can be established according to the positive and negative comment texts, so as to understand the advantages and disadvantages of the enterprise’s products, and then continuously improve the products and maintain the competitiveness of the enterprise. User comments are affected by the product's life cycle, trend direction and the maturity, which is a typical big data classification learning problem in nonstationary environments. And for example, in the smart grid control system, smart meters distributed around the site gradually collect users’ electricity consumption information and return them to the data processing center to further mine the classification model, which can be used to predict the electricity consumption in advance, formulate energy-saving electricity generation and transmission plans, and smooth the electricity load. Obviously, users’ electricity consumption data is affected by different seasons, weather and the time, and the smart meter itself also would have aging, failure and attenuation. Therefore, the smart control system of the smart grid also includes the demand for classification learning of electricity consumption big data accumulated in nonstationary environments. Similar application scenarios are very typical in real life and production practice, and the researches in this field have important practical background and theoretical significance.

Because there are many typical applications of classification learning in nonstationary environments and they have close relationship with social practice, a lot of new and effective classification learning algorithms have emerged in this research area. According to the amount of data processed at one time, it can be divided into online classification algorithms and batch processing classification algorithms. Many algorithms can also be divided into single-classifier algorithms and multi-classifier ensemble algorithms based on the construction of the classification model. According to the detection mechanism to decide whether the generation environment of accumulated big data has changed, they can be divided into active detection algorithms and passive detection algorithms [8-10].

Among the numerous algorithms, it is worth paying attention to the recently proposed Learn++.NSE algorithm [7][11][12]. Learn++.NSE is a passive, multi-classifier integrated batch classification algorithm. The experimental results on the real-world dataset and the generated dataset show that in the process of classification learning for the accumulated big data in nonstationary environments, the Learn++.NSE algorithm has achieved more accurate and stable classification results than the single-classifier algorithm. The Literature [11] also points out that if not pruning, Learn++.NSE will achieve higher classification accuracy. Especially in the environment of periodic recurrence, Learn++.NSE will better show the advantage of retaining all the base classifiers. However, if all base classifiers are retained, as the number of base classifiers increases, the ensemble time of Learn++.NSE will increase exponentially. Therefore, Learn++.NSE also offers a pruning version when considering execution efficiency, which includes two different pruning strategies: pruning according to the accuracy of the base classifier and pruning according to the generation time of the base classifier.

In our previous research, it can be found that there is a way to further optimize the pruning mechanism of the Learn++.NSE-Error-based algorithm, which perform pruning based on the accuracy of the base classifier. Therefore, a pruning optimized fast Learn++.NSE algorithm was proposed, referred to as NewLearn++NSE-Error-based. Theoretical analysis and experiments show that the NewLearn++.NSE-Error-based algorithm can further improve the classification accuracy of the ensemble classification model, which is suitable for efficient classification learning of accumulated big data in nonstationary environments.
The main contributions of this paper include the following three aspects:

1) The shortcomings of the pruning mechanism of the original Learn++.NSE-Error-based algorithm were found, and a new pruning mechanism was designed to retain the latest two base classifiers without pruning, which built a foundation for the NewLearn++.NSE-Error-based algorithm.

2) Based on the new pruning mechanism, a new pruning algorithm for Learn++.NSE is proposed. The pruning algorithm further improves the accuracy of the integrated classification model for accumulated big data.

3) Comparison and analysis were made between the proposed NewLearn++.NSE-error-based algorithm and the original Learn++.NSE-Error-based algorithm on the data set generated by the program and the data set in the real world. The experimental results verify that under the same learning scenario, the NewLearn++.NSE-error-based algorithm can further improve the accuracy of the integrated classification model compared with the Learn++.NSE-error-based algorithm, which is suitable for the rapid classification learning for the accumulated big data.

II. The Related Research Work and Algorithms

The algorithms as far as we know for leaning in nonstationary environments can be summarized as a mind map in Figure 1. In the algorithms of classification learning for big data accumulated in nonstationary environments, the online classification algorithm processes one instance data at a time, which can adapt to changes of data generation environment more quickly, but with poor stability. In addition, the online classification algorithm is susceptible to interference from the order of learning data and noise data[12].

The typical online classification algorithms include TSVDD[13], eAdaBoost[14], KB-IELM[15], OS-ELM[16], etc. The batch processing classification algorithm processes a batch of data at a time, which can rely on the more data to overcome the interference of noise data, and the classification result is relatively stable. However, if the batch data is multi-source and the distribution probability is different, the algorithm is more difficult to process. The typical batch processing classification algorithms include IGMM-CD[17], EDTU[18], Learn++.NSE[11], etc. In order to improve the execution efficiency, the batch processing algorithm could make use of the parallel computation mechanism. Under this mechanism, different data batches are analyzed by different computers or different CPU cores. The parallel computer mechanism accelerates the data processing process, which is very suitable for big data. Take the PRLearn++.NSE [19] for example, which is proposed in our previous research to improve classification learning efficiency for big data, it uses the old base-classifiers as a supplement to the new base-classifier. It constructs a fast and parallel ensemble mechanism to accelerate the execution of original Learn++.NSE.

The single classifier algorithm generally has a low amount of calculation, but the classification accuracy is also low. And in order to take into account the retention of the original classification information, the adjustment is slow when processing new data. Typical algorithms include VFDT[20], ODTC[21], iCaRL[22], and the online limit learning algorithm WOS-ELM[23], which using neural networks for learning, etc. The multi-classifier ensemble algorithm is more popular for the classification learning of accumulated big data where the data generation environment may change. Because the multi-classifier ensemble algorithm generally has a lower classification error rate than the single-classifier algorithm, it is easier to adjust to the new data generation environment by adding and removing the base classifier, and the outdated classification information can be eliminated by deleting the base classifier. By flexibly adding and removing base classifiers and adjusting the weight of each base classifier, the multi-classifier ensemble algorithm can generally obtain a lower classification error rate and a more stable classification result than the single-classifier algorithm. Typical multi-classifier ensemble algorithms include SEA[24], ONSBoost[25], DWM[26], Learn++.NSE, etc.

The passive detection algorithm assumes that the probability distribution of data generation may change over time. Therefore, regardless of the change, the classification model must be updated whenever a new dataset arrives.
However, with the increase of base classifiers, the update efficiency decreases rapidly. Pruning the base classifier can improve the update efficiency, but useful classification information may be lost, which needs to be comprehensively considered according to the actual situation. Typical passive detection algorithms include OLIN\cite{27}, Learn++.NSE, etc. The active detection algorithms attempt to determine whether the probability distribution of data generation has changed, and updates the classification model only when the probability distribution has changed, so as to avoid invalid update and can improve the execution efficiency. However, there are risks of false positive and false negative in the judgment of the active detection algorithm, especially when disturbed by noise data. Typical active detection algorithms include CUSUM\cite{28}, JIT\cite{29}, ICI\cite{30}, etc.

![FIGURE 1. The mind map of learning in nonstationary environments](image)

It is worth noting that many algorithms provide the pruning strategies in consideration of the classification learning efficiency of accumulated big data. For example, the SEA algorithm fixes the capacity of the ensemble classifier. When the ensemble capacity is exceeded, it would be pruned according to the generation time or accuracy of the base classifier. In addition, the ensemble capacity of the DWM algorithm is not fixed but dynamically adjusted, but once the set capacity is exceeded, the base classifier will also be pruned according to the preset criteria. Learn++.NSE also provides two versions: Learn++.NSE-Error-based that preforms pruning based on the accuracy of the base classifier and Learn++.NSE-Age-based that preforms pruning based on the generation time of the base classifier. There is no definite advantage or disadvantage between the two methods, which needs to be selected according to the data generation situation in actual applications. Considering the research results in the literature \cite{11}, under the same test conditions, Learn++.NSE can achieve a lower classification error rate compared to SEA algorithm and DWM algorithm. In addition, the Learn++.NSE algorithm can also deal with the case of variable rates and periodic data generation that the above algorithms cannot handle well. Therefore, taking Learn++.NSE-Error-based as an optimization research object, the classification accuracy of the classification model is further improved.

III. Learn++.NSE-Error-based Algorithm

Learn++.NSE algorithm is a passive, batch-type multi-classifier ensemble learning algorithm. The Learn++.NSE algorithm uses the time-adjusted loss function named sigmoid to calculate the weighted average error rate of base classifiers generated in different periods. Each base classifier calculates the voting weight according to its own weighted average error rate, so that the recent base classifier with a lower weighted average classification error rate can obtain a higher voting weight. It is found that the Learn++.NSE algorithm without pruning can achieve more stable and accurate classification results. However, if the base classifier is not pruned, the time for the Learn++.NSE algorithm to construct the ensemble classifier will increase exponentially, which will greatly affect the efficiency of the algorithm in processing accumulated big data. Therefore, the Learn++.NSE algorithm also provides a pruning version. When considering the efficiency of processing big data, the pruning version of Learn++.NSE can be selected. The Learn++.NSE algorithm can remove the oldest base classifier based on the generation time of the base classifier (Learn++.NSE-Age-based).

The implementation details of Learn++.NSE-Error-based are given below. Define the meaning of operator \([\ ]\) as\([A] = \{1, A = True\}
\(0, A = False\).

**Input:**
- Training dataset \(d^t\), \(t=1,2,3,\ldots\), \(d^t\) is \(\{x^t(i) \in X; y^t(i) \in Y = \{1,\ldots,c\}\}, i = 1,\ldots,m^t\), \(x^t(i)\) represents the data point at time \(t\), \(y^t(i)\) represents the category in
\( \{1, \ldots, c\} \) which the data point belongs to at time \( t \), \( n^t \) represents the number of data points, \( d^t \) represents batch data gradually accumulating to form big data.

\( \circ \) A classification algorithm for forming base classifiers, strong classification algorithm is recommended.

\( \circ \) The falling morphological parameter \( a \) of the Sigmoid function, the half value intersection parameter \( b \).

\( \circ \) The capacity of the base classifier \( \text{ensembleSize} \).

**Output:** The Ensemble classification model \( H^t \).

Perform the following steps for each batch of dataset \( d^t \) as Fig. 2.

**FIGURE 2.** The flow diagram of Learn++-NSE-Error-based algorithm

In the step 5 of Learn++-NSE-Error-based algorithm, when calculating the error rate of the base classifier, the \( D^t \) obtained by the steps 2) and 3) were used for weighting. The purpose of doing this is that when calculating the error rate of the base classifier, if the base classifier cannot correctly classify the data that the current ensemble classifier misclassified, the base classifier will get a higher penalty and get a lower voting weight. In other words, the base classifier that can correctly classify data that the current ensemble classifier cannot correctly classify will get a higher voting weight. In this way, the overall ensemble classifier tends to be positively optimized and strives to identify data that is not
easy to classify, which can reduce the classification error rate of the ensemble classifier.

Additionally, when the ensemble classifier gets a lower error rate, $E^t$ is smaller, and the base classifier of the misclassified instance is punished more heavily. This is because the low classification error rate obtained by the ensemble classifier means that the environment has not changed significantly, and these instances should have been learned before and should be correctly classified. If this data is misclassified by a base classifier at this time, the voting weight should be greatly reduced and heavy penalty should be given. On the contrary, $E^t$ is larger, it means that the environment has greatly changed. These instances have not been analyzed before, and the penalty for misclassification should be lighter, and more base classifiers should be encouraged to cooperate to perform ensemble classification and reduce the classification error rate.

The research results show that the Learn++.NSE algorithm significantly improves the classification accuracy compared to the single-classifier algorithm, SEA algorithm and DWM algorithm\[11\]. In addition, its pruning version Learn++.NSE-Error-based algorithm was given a Java implementation in the latest version of the massive online learning platform Massive online analysis (MOA).

IV. NewLearn++.NSE-Error-based Algorithm

A careful analysis about the implementation details of the Learn++.NSE-Error-based algorithm shows that in order to realize efficient classification learning of accumulated big data, the Learn++.NSE-Error-based algorithm sets an upper limit, that is \textit{ensembleSize}, on the number of base classifiers. When the number of base classifiers exceeds the limit value, according to the classification error rate of each base classifier on the current dataset, the base classifier with the highest classification error rate is deleted, and the final ensemble classifier is generated from remaining base classifiers.

Because the pruning algorithms focus on the execution efficiency, they would set an upper limit on the base-classifiers. When the number of base-classifiers exceeds the upper limit, some pruning strategy would remove the base-classifier that is unsuitable. The pruning strategy could be that removing the oldest base-classifier based on generation time. The rationale for this is that the generation of data could change, the oldest base-classifier should be the one that is most unsuitable and should be removed. The pruning strategy also could be that removing the one of highest error rate after evaluating all the base-classifiers. The rationale for this is that since the one has the highest error rate, it should be most unsuitable for current data set and should be removed.

There is no fixed good or bad among the two different pruning strategies when dealing with data set. The final learning result is relative to the distribution of data and the change of the distribution. The choose of the pruning strategy should accord to the actual application scenarios. Without losing generality, the Learn++.NSE-Error-based algorithm based on error rate is chosen for comparative study.

The research shows that under the premise of considering the efficiency of ensemble classification learning for accumulated big data, Learn++.NSE-Error-based is an efficient ensemble learning algorithm.

However, during the research process, it can be found that during the integration process of Learn++.NSE-Error-based, the base classifier $h_{t-1}$ that was just generated last time tends to encounter $e_{t-1}^t > 1/2$ in the current classification error rate evaluation process and will be removed. It is considered that the newly generated base classifier should be the most suitable for the current classification learning environment and should not be pruned. In addition, since the currently generated base classifier $h_t$ has the lowest weighted classification error rate and the highest voting weight, the base classifier $h_{t-1}$ will be corrected by $h_t$ and the subsequent base classifier ensemble, even if the data generation environment abruptly changes, and it does not have to be pruned. Therefore, when pruning the base classifier, the proposed NewLearn++.NSE-Error-based algorithm retains the current base classifier $h_t$ and the previous base classifier of the current base classifier $h_{t-1}$ without being pruned, which can further improve the
accuracy of classification learning for accumulated big data. That is the main differences of the two algorithms. The implementation details of the NewLearn++.NSE-Error-based algorithm are described below.

**Input:**

1. Training dataset $d^t$, $t=1,2,3,...$, $d^t$ is $\{x^t(i) \in X; y^t(i) \in Y = \{1,...,c\}\}$, $i = 1,...,m^t$, $x^t(i)$ represents the data point at time $t$, $y^t(i)$ represents the category in $\{1,...,c\}$ which the data point belongs to at time $t$, $m^t$ represents the number of data points, $d^t$ represents batch data gradually accumulating to form big data.

2. A classification algorithm for forming base classifiers, strong classification algorithm is recommended.

3. The falling morphological parameter $a$ of the Sigmoid function, the half value intersection parameter $b$.

4. The capacity of the base classifier $ensembleSize$.

**Output:** Ensemble classification model $H'$.

Perform the following steps for each batch of dataset $d^t$ as Fig. 3

The proposed NewLearn++.NSE-Error-based algorithm analyzes the integration process of base classifiers, proposes to retain the two current base classifiers that are very important for the current classification learning, and selects the base classifier with the highest classification error rate from earlier base classifiers to delete. In the classification learning experiments of generated and accumulated big data, it is proved that this method can further significantly improve the accuracy of the Learn++.NSE-Error-based ensemble classification algorithm.

**V. The Experimental Results and Analysis**

Using the dataset described in the literature [11] (http://users.rowan.edu/~polikar/research/nse/) to analyze the classification learning effect of the proposed NewLearn++.NSE-Error-based algorithm.

Analyzing the time complexity, the time consumption of the two algorithms is mainly concentrated in the evaluation phrase of the base-classifiers when learning the new data set. Assuming the number of the base-classifiers is $N$, the Learn++.NSE-Error-based should evaluate $N$ base-classifiers and the NewLearn++.NSE-Error-based should evaluate $N-2$ base-classifiers. It can be considered that the time complexity of the two algorithms in this stage are both $T(N)$.

In the case that the time complexity of classification learning is almost the same, the classification error rate is compared with the Learn++.NSE-Error-based algorithm.

The experiment environment is ThinkPad T460P, CPU is Core i7-6700, dominant frequency is 2.6GHz, memory is 24GB, Win10 64-bit operating system, the experimental development environment is MATLAB R2016a.

For the fair comparative analysis, the experiments follow these settings: (1) In the process of experiment, the accumulated big data is decomposed to multiple batches of dataset for training of the ensemble classifier. The accumulated big data is generated by nonstationary environments. The data generation distribution, generation rules are not known by the two classification learning algorithms in advance. (2) During the experiment, both classification algorithms use the same base classifier. That is the CART decision tree. The parameters of CART are both mergeleaves=on, minleaf=1, prune=on, surrogate=off. The NewLearn++.NSE-Error-based and Learn++.NSE-Error-based algorithm use the same sigmoid parameters, $a=0.5$, $b=10$, and the pruning threshold $ensembleSize$ is set to 15. (3) Each algorithm ran 20 times, and the values of classification error rate and g-mean were recorded each time.

The experimental results and analysis of the NewLearn++.NSE-Error-based and Learn++.NSE-Error-based algorithm on the artificially generated SEA dataset, the rotating spiral dataset and the real weather prediction dataset NOAA are given below.

**A. The SEA dataset**

The SEA dataset is a data set introduced when the SEA algorithm was proposed, and it is currently a benchmark dataset for testing a classification learning algorithm for accumulated big data in nonstationary environments. The dataset consists of three numeric fields and a class label. To increase the difficulty of classification learning, only two of the three numeric fields are related to classification labels,
and another one is interference data. The classification label is a two-valued attribute. When the sum of the two relevant numeric fields is less than the preset threshold $\sigma^t$, the class label is 2, otherwise 1. In addition, in order to increase the complexity of ensemble classification learning, 10% noise data was added to the training dataset during the experiment. At a given moment, the preset threshold $\sigma^t$ will change to simulate the sudden change in the data generation environment.

The experiment was divided into 200 batches of data, and each batch contains 2000 records, that were 400,000 records. The threshold $\sigma^t$ changed three times at time $t = 50$, $t = 100$, and $t = 150$, that is $4 \to 7 \to 4 \to 7$. While simulating the abrupt change of environment, it also simulates the situation where the data generation environment periodically recurs. Independent test dataset was used to evaluate the classification accuracy of different classification algorithms. In each batch of data, 2,000 test

![Flow Diagram](image-url)

**FIGURE 3. The flow diagram of NewLearn++.NSE-Error-based algorithm**

The experiment was divided into 200 batches of data, and each batch contains 2000 records, that were 400,000 records. The threshold $\sigma^t$ changed three times at time $t = 50$, $t = 100$, and $t = 150$, that is $4 \to 7 \to 4 \to 7$. While simulating the abrupt change of environment, it also simulates the situation where the data generation environment periodically recurs. Independent test dataset was used to evaluate the classification accuracy of different classification algorithms. In each batch of data, 2,000 test
data sets were extracted independently from the same data
generation environment, a total of 200 test data sets were
extracted, and the error rates of different classification
algorithms were calculated.

The experiment recorded the classification error rate of
NewLearn++.NSE-Error-based and Learn++.NSE-Error-
based algorithm for 200 batches of data sets, as shown in
Figure 4. The categories of this dataset are uneven, and the
classification accuracy evaluation index has poor ability to
evaluate the classification model when the categories are
uneven. It is better to use g-mean for evaluation, $g - \text{mean} = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}}$, as shown in Figure 5.

![FIGURE 4. Classification error rate of two algorithms on SEA](image)

![FIGURE 5. G-mean value of two algorithms on SEA](image)

It can be seen from the Figure 4 that, when the data
generation environment is relatively stable, the
NewLearn++.NSE-Error-based algorithm achieves a lower
classification error rate than the Learn++.NSE-Error-based
algorithm. When the data generation environment changes
suddenly, the NewLearn++.NSE-Error-based algorithm has
a slightly higher classification error rate than the
Learn++.NSE-Error-based algorithm, which is due to the not
pruning strategy of $h^L$ and $h^L_1$. However, the
NewLearn++.NSE-Error-based algorithm would quickly
achieve a lower classification error rate than the
Learn++.NSE-Error-based algorithm. In addition, it can be
seen from Figure 5 that if the data imbalance is considered,
the g-mean value of the NewLearn++.NSE-Error-based
algorithm is always better than the Learn++.NSE-Error-
based algorithm.

Because the classification evaluation index in the
experiment is not the normal distribution, the rank sum test
was used to determine whether the classification learning
effect of the two different classification algorithms is
significantly different. The rank sum test was performed for
the average classification accuracy and average g-mean of
each batch data of NewLearn++.NSE-Error-based and
Learn++.NSE-Error-based algorithm, and the results are
shown in the Table 1. The experimental results in the Table
1 show that the classification learning effect of
NewLearn++.NSE-Error-based and Learn++.NSE-Error-
based algorithm is significantly different, and
NewLearn++.NSE-Error-based is better than Learn++.NSE-
Error-based, which reflects the effectiveness of the
optimized pruning strategy.

B. The Rotating Spiral Dataset

The rotating spiral dataset consists of 4 data point sets
representing four different spirals, of which 2 spirals
represent the majority class and the other 2 spirals represent
the minority class, as shown in Figure 6. In the experiment,
each spiral rotates $2\pi$ radians in 300 time points along the
spiral center at a constant speed, and each time point
produces 500 data points, which constitute the accumulated
big data. In the dataset generated at each time point, the
minority class accounted for 5%. Since the distribution scene
of data points will reappear once every $\pi$ radians, there will
be two repeating scenes of classification training in the
experiment to simulate the periodic imbalanced
classification training scenarios in nonstationary
environments. Because this classification scene is very
difficult to learn, no noise data was added. At each time point,
an independent test dataset was generated according to the
same method to evaluate the classification learning effect of
different classification algorithms.
The experiment recorded the classification error rate of NewLearn++.NSE-Error-based and Learn++.NSE-Error-based algorithm for 300 batches of dataset, as shown in Figure 7. The rotating spiral dataset is a dataset of imbalanced categories. In order to better evaluate the classification learning effect of different algorithms, the g-mean evaluation value of each classification result was recorded, as shown in Figure 8.

C. The NOAA Weather Dataset

The NOAA dataset is weather big data collected by the U.S. National Oceanic and Atmospheric Administration relying on sensors from hundreds of regions around the world. The dataset contains long-term weather information since 1930 and characterizes the changing trends of weather, including wind speed, visibility, temperature, air pressure and other characteristic attributes. In addition, it contains a classification label that indicates whether it is raining. Different from the previous SEA and rotating spiral dataset, which are artificially generated dataset, this dataset is a real-world dataset that can be used to verify the classification learning effect of new classification algorithm in real scenarios. This experiment selected a subset of the dataset, a 50-year weather data of the Offutt Air Base in Nebraska from 1949 to 1999, for classification learning, which is a long-term and periodic accumulated big data set. According to the availability of data, the characteristic attributes with a miss rate of more than 15% were removed, and the remaining 8 attributes were selected as the features of the training set. The dataset contains a total of 18159 records, and 69% of the data were labeled rain=no, 31% were labeled rain=yes. During the experiment, 10% of noise data was randomly generated. The accumulated big data was divided according to the year, and every 365 records constitute a batch of data for classifier training, which was divided into different pruning strategies of the two algorithms. It can be seen from the Figure 7 that in terms of classification error rate, the NewLearn++.NSE-Error-based algorithm is always lower than Learn++.NSE-Error-based. Considering the category imbalance of training dataset, it can be found that the NewLearn++.NSE-Error-based algorithm is always better than Learn++.NSE-Error-based by analyzing the g-mean value, which is shown in Figure 8. It can be seen from the rank sum test of classification learning result indicators shown in Table 1, the classification learning effect of NewLearn++.NSE-Error-based and Learn++.NSE-Error-based algorithm is significantly different, and the NewLearn++.NSE-Error-based algorithm is better than Learn++.NSE-Error-based.
50 batches of dataset. The experiment was performed in the form of training then testing, that is, the first batch of dataset was used for training, the second batch of dataset was used for testing, then the second batch of dataset was trained, and so on. A total of 49 classification prediction results were obtained. The average classification error rate and average g-mean of each prediction were recorded, as shown in Figure 9 and Figure 10.

It can be seen from the Figure 9 that, the classification learning effect of NewLearn++.NSE-Error-based and Learn++.NSE-Error-based algorithm is equivalent at the initial stage. Subsequently, when the size of the ensemble classifier exceeds the preset value, both algorithms started to prune the base classifier. Since the two algorithms adopt different pruning strategies, there is difference in the classification learning effect. As Figure 9, in terms of classification accuracy, the NewLearn++.NSE-Error-based algorithm is generally superior to Learn++.NSE-Error-based. Similarly, considering the g-mean evaluation value in the case of imbalanced training data categories, as shown in Figure 10, it can be concluded that the NewLearn++.NSE-Error-based algorithm is better than Learn++.NSE-Error-based. It can also be seen from the rank sum test of the classification learning result evaluation in Table 1 that there is a significant difference in the classification learning effect of these two algorithms. And the classification learning effect of the newly proposed NewLearn++.NSE-Error-based algorithm is better than that of Learn++.NSE-Error-based algorithm, which reflects the effectiveness of the optimized pruning strategy.

The execution process of the proposed NewLearn++.NSE-Error-based algorithm was carefully analyzed and the reason why the algorithm works was explored. It can be found that during the execution of the Learn++.NSE-Error-based algorithm, the weighted error rate of the newly generated base classifier $h_k$ on the latest data set often appears that $e_k > 1/2$, which makes the final voting weight $W_k$ of the base classifier is very close to 0 and
is very likely to be pruned. This is inconsistent with daily experience. The base classifier that just produced should be very suitable for the current data set and should be used seriously. After analysis, it is found that this is caused by the weighting mechanism of the Learn++ .NSE-Error-based algorithm. This weighting mechanism pays more attention to the data that cannot be recognized by the current ensemble classifier, and the data that cannot be recognized by the current ensemble classifier often cannot be recognized by the recent base classifier, which makes the weighted error rate $\epsilon^*_k$ easily greater than 1/2. The algorithm is optimized by keeping the latest two base classifiers without pruning. The experimental results verified our expected hypothesis and the optimization does help to improve the accuracy of the ensemble classifier.

VI. CONCLUSION

The classic classification learning algorithm Learn++ .NSE in nonstationary environments was optimized, and the new version Learn++ .NSE-Error-based of the algorithm was redesigned to prune according to the accuracy of the base classifier, and a new optimized pruning algorithm NewLearn++ .NSE-Error-based was proposed. The NewLearn++ .NSE-Error-based algorithm retains the latest two base classifiers, which are very important for the current ensemble classification, to avoid the problem of the classification accuracy of the ensemble classifier being reduced due to the wrong pruning of the two base classifiers. Experimental results on real dataset and artificially generated standard test dataset showed that under the premise of the same time complexity, the NewLearn++ .NSE-Error-based algorithm can achieve higher classification accuracy than Learn++ .NSE-Error-based, which is an effective fast classification learning algorithm for accumulated big data. The proposed algorithm can be used in application scenarios such as e-commerce recommendation systems, social public opinion analysis, and intelligent internet of things to improve the accuracy of prediction.

Although this research has made some progress, there are still some shortcomings. For example, it is still dissatisfied that the pruning strategy of the algorithm proposed so far. The base classifier of current algorithms will no longer be selected to use after it is deleted, which may make the ensemble classifiers cannot well track the changes of data set and reduce the accuracy of it. The reuse of base-classifiers will be the key breakthrough in the future research.

REFERENCES

[1] Duda P, Rutkowski L, Jaworski M, et al. On the Parzen Kernel-Based Probability Density Function Learning Procedures Over Time-Varying Streaming Data With Applications to Pattern Classification[J]. IEEE Transactions on Cybernetics, 2018, PP(99):1-14.

[2] Cesare Alippi, Wen Qi, Manuel Roveri. Learning in Nonstationary Environments: A Hybrid Approach[C]// International Conference on Artificial Intelligence and Soft Computing. 2017.

[3] Sayedmouchaweh M. Learning from data streams in dynamic environments[M]// Learning from Data Streams in Dynamic Environments. Springer International Publishing, 2016.

[4] Bilal M , Lin Z , Liu N . Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift[J]. Neurocomputing, 2015, 149(Part A):316-329.

[5] Conca P , Timmis J , Rogério de Lemos, et al. An Adaptive Classification Framework for Unsupervised Model Updating in Nonstationary Environments[C]// International Workshop on Machine Learning, Optimization and Big Data. Springer International Publishing, 2015.

[6] Alzubi O A , Alzubi J A , Alweshah M , et al. An optimal pruning algorithm of classifier ensembles: dynamic programming approach[J]. Neural Computing and Applications, 2020, 32(5):1-17.

[7] Ali M A , Jan Z . A novel method for creating an optimized ensemble classifier by introducing cluster size reduction and diversity[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, (99):1-11.

[8] Cesare Alippi, Roveri, et al. Learning in Nonstationary Environments: A Survey[J]. IEEE computational intelligence magazine, 2015.

[9] Noda I. Optimality and Equilibrium of Exploration Ratio for Multiagent Learning in Nonstationary Environments[C]//International Workshop on Multi-Agent Systems and Agent-Based Simulation. Springer International Publishing, 2016.

[10] Shaker A. Recovery analysis for adaptive learning from non-
stationary data streams[11]. Elsevier Science Publishers B. V. 2015.

[11] ELWELL R, POLIKAR R. Incremental Learning of Concept Drift in Nonstationary Environments[J]. IEEE Transactions on Neural Networks, 2011, 22(10):1517-31.

[12] KULKARNI P, ADE R. Incremental Learning From Unbalanced Data with Concept Class, Concept Drift and Missing Features: A Review[J]. International Journal of Data Mining & Knowledge Management Process, 2014, 4(6):15-29.

[13] Lei B, Xiao H, Guo Y. Two kinds of targets on-line classification based on incremental SVDD[C]// 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC). IEEE, 2017.

[14] Dinakaran S, Thangaiah P R J. Ensemble Method of Effective AdaBoost Algorithm for Decision Tree Classifiers[J]. International Journal of Artificial Intelligence Tools, 2016, 26(3).

[15] Zhang W, Xu A, Ping D, et al. An improved kernel-based incremental extreme learning machine with fixed budget for nonstationary time series prediction[J]. Neural Computing and Applications, 2017, 31(6):1-16.

[16] Huang S, Wang B, Qiu J, et al. Parallel ensemble of online sequential extreme learning machine based on MapReduce[J]. Neurocomputing, 2016, 174(PA):352-367.

[17] Luan S O, Batista G E A P A. IGMM-CD: A Gaussian Mixture Classification Algorithm for Data Streams with Concept Drifts[C]// 2015 Brazilian Conference on Intelligent Systems (BRACIS). IEEE, 2015:55-61.

[18] Xing Z, Meili W, Yang Z, et al. Classifier Ensemble Algorithm for Data Stream with AttribuUncertainty[J]. Journal of Computational and Theoretical Nanoscience, 2016, 13(10):7519-7525.

[19] Yan Shen, Jianguo Du, Jingen Tong, et al. A Parallel and Reverse Learn++.NSE Classification Algorithm[J]. IEEE ACCESS, 2020, 8(1):64157-64168.

[20] Desai S, Roy S, Patel B, et al. Very Fast Decision Tree (VFDT) algorithm on Hadoop[C]// 2016 International Conference on Computing Communication Control and automation (ICCUBEA). IEEE, 2016.

[21] Das S, Dahiyi S, Bharadwaj A. An online software for decision tree classification and visualization using c4.5 algorithm (ODTC)[C]// International Conference on Computing for Sustainable Global Development. IEEE, 2014.

[22] Rebuffi S A, Kolesnikov A, Sperl G, et al. iCaRL: Incremental Classifier and Representation Learning[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017.

[23] Ding S, Mirza B, Lin Z, et al. Kernel based online learning for imbalance multiclass classification[J]. Neurocomputing, 2017, 277:139-148

[24] Nick Street W., Kim YongSeog. A streaming ensemble algorithm (SEA) for large-scale classification. [C] //Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2001: 377-382

[25] Pocock Adam, Yiapanis Paraskevas, Singer Jeremy, et al. Online non-stationary boosting [J] Lecture Notes in Computer Science, 2010, 5997: 205-214

[26] MEJRI D, LIMAM M, WEIHS C. A new dynamic weighted majority control chart for data streams[J]. Soft Computing, 2016:1-12.

[27] Cohen Lior, Avrahami Gil, Last Mark, et al. Info-fuzzy algorithms for mining dynamic data streams. [J] Applied Soft Computing Journal, 2008, 8(4): 1283-1294

[28] HUANG Y, TANG J, CHENG Y, et al. Real-Time Detection of False Data Injection in Smart Grid Networks: An Adaptive CUSUM Method and Analysis[J]. IEEE Systems Journal, 2016, 10(2):532-543.

[29] HUANG Y, TANG J, CHENG Y, et al. Real-Time Detection of False Data Injection in Smart Grid Networks: An Adaptive CUSUM Method and Analysis[J]. IEEE Systems Journal, 2016, 10(2):532-543.

[30] Alippi Cesare, Boracchi Giacomo, Roveri Manuel. A just-in-time adaptive classification system based on the intersection of confidence intervals rule. [J] Neural Networks, 2011, 24(8): 791-800.
YONG CHEN received a bachelor degree from Peking University, Beijing, China, in 2000, and a master degree from Shanghai Jiaotong University, Shanghai, China, in 2007. He is currently pursuing for the Ph.D. degree in Computer Science and Communication Engineering at Jiangsu University, Jiangsu, China. He is an expert enjoying the special government allowance of the State Council. His main research interests include the Internet of Things, computers and communications.

YUQUAN ZHU received Ph. D. degree in Engineering from Southeast University, Professor, Ph. D. supervisor, Research Interests include Data Mining, Complex Information System Integration, KDD, database systems & its application.

HAIFENG CHEN received the master degree from Beijing University of Posts and Telecommunications, Beijing, China, and the Ph.D. degree in Engineering from Tsinghua University, Beijing, China. He is currently the general Manager of CU IoT Co., Ltd., the president of CU IoT Research Institute. He was a visiting lecturer of BUPT, Professor of industry of NJUPT, Vice President of Guangdong Precision Medicine Association, Senior Planner of smart city of MIIT. His major research interests includes Industrial Internet, IDC, cloud computing, big data, IoT and technology applications, especially in e-government affairs, e-healthcare and Industrial Internet. He is proficient in 5G and IoT technology and received sci-tech prizes from 2014 to 2020, such as British GTB Innovation Award and DCD Internet Data Center Award.

YAN SHEN received the Ph.D. degree in School of Computer Science and Communication Engineering from Jiangsu University, Jiangsu, China, in 2013. He is currently an Associate Professor with the Department of Information Management and Information System in Jiangsu University of China. He was a Postdoctoral Research Associate with the Control Science and Engineering and Management Science and Engineering in Jiangsu University of China, respectively from 2013 to 2017 and 2018 to 2020. His main research interests include big data mining, machine learning and business intelligence.

ZHAO XU received the B.S. degree in information and computing science from Nanjing University of Science and Technology, Nanjing, China, in 2003, and the M.E. degree in computer science and technology from Jiangsu University of Science and Technology, Zhenjiang, in 2019. He is currently the deputy general manager and technical director of Nanjing Longyuan Microelectronics Technology Co., Ltd. His main research interests include IoT gateway architecture design and distributed microservice IoT back-end architecture design.