

Under Sampling Adaboosting Shapelet Transformation for Time Series Feature Extraction

Yohan Joo and Jongpil Jeong(✉)

Department of Smart Factory Convergence, Sungkyunkwan University, Suwon, Gyeonggi-do 16419, Republic of Korea
{yahoo4903,jpjeong}@skku.edu

Abstract. To predict for machine defects, a classifier is required to classify the time series data collected from the sensors into the fault state and the normal state. In many cases, the data collected by sensors is time series data collected at various frequencies. Excessive computer load is required to handle this as it is. Therefore, there has been a lot of research being done on the process of extracting features that are highly classified from time series data. In particular, data generated at real-world is unbalanced and noisy, requiring time series classifiers to minimize their impact. Shapelet transformation is generally effectively known for classifying time series data. This paper proposes a process of feature extraction that is strong for noise and over-fitting to be applicable in practice. We can extract the feature from the time series data through the proposed algorithm and expect it to be used in various fields such as smart factory.

Keywords: Shapelet · Adaboosting · Under Sampling · Time series classification

1 Introduction

There has been a variety of studies on the use of sensors and big data analysis for the purpose of preserving the prediction of machine defects. Studies have been conducted to effectively classify time series datasets sampled at certain frequencies in normal and fault conditions on sensors in order to maintain the machine's preventive. Specially, a variety of methods have been proposed to extract features that best represent the characteristics of a time series in order to effectively handle a variety of sampling data. The simplest method, most commonly used to extract characteristics from time series data, is a statistical technique that extracts means, standard deviations, curvature, etc. from time series data. In the framework proposed by Helwig et al. [1], high dimensional time series data of 100 Hz were simplified by calculating means, variances, and curvature. Zhu et al. [2] also introduced the Shapelet transform based classification framework to determine short term voltage stability, which is a highly categorised transformation technique that uses information acquisition to record high titled decision boundaries.

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11624, pp. 69–80, 2019.
https://doi.org/10.1007/978-3-030-24311-1_5

Shapelet transformation is a technique that converts data into distance data so that it can generate decision boundaries that are highly classified based on information gain. It has a high classification performance through the process of extracting partial time series that can best represent it from a single time series. Recently, a method was proposed for converting multivariate time series data into distance data by applying a shapelet separately to each set.

This paper proposes an effective feature approach to binary classification of time series datasets. The proposed feature extraction technique is an ensemble combining Random Under Sampling and Adaboosting techniques. The results of each classification shall be evaluated so that the results are reflected in the next sampling. Finally, adaptive learning is repeated and converted into distance data that can produce high classification results. The transformed distance data is classified through the classification phase and can be recorded for high classification results.

This paper is organized as follows. In Sect. 1, the necessity of the algorithm proposed in this paper is explained in connection with the industrial field. Section 2 explains the methodology of the proposed algorithm, and Sect. 3 explains the structure of the proposed algorithm. Section 4 experiments the performance evaluation of the proposed algorithm. Finally, conclusion and future works are presented in Sect. 5.

2 Related Work

2.1 Time Series Shapelet Classification

Shapelet means the part sequence that best represents this for the Time series data [3]. The Time Series Shapelet Classification method extracts a partial sequence from a time series, obtains the Euclidean distance from sub sequence and the existing time series data and converts the time series data into distance data. It is then a method of classifying the data by finding a sub sequence that provides a clear decision boundary with high information gain for converted distance data. Instead of requiring complex computations and therefore having to spend a lot of time on them, it is a way of recording high classification performance. For a more detailed description of the Shapelet, see [4] of Ye et al. and [5] of Zhu et al. Figure 1 shows the method of Shapelet Transformation.

2.2 Sampling Method

Under sampling technique is a technique that samples only part of a sample used from data. In contrast, the technique of synthesize new data or creating samples that are larger than traditional datasets by randomly add traditional data is called over sampling technology. These techniques are an effective way to reduce the impact of noisy data and overfitting or underfitting in the data. Randomly reducing or duplicating data on existing datasets is called random sampling. It is known that this is simple but can be more effective in many

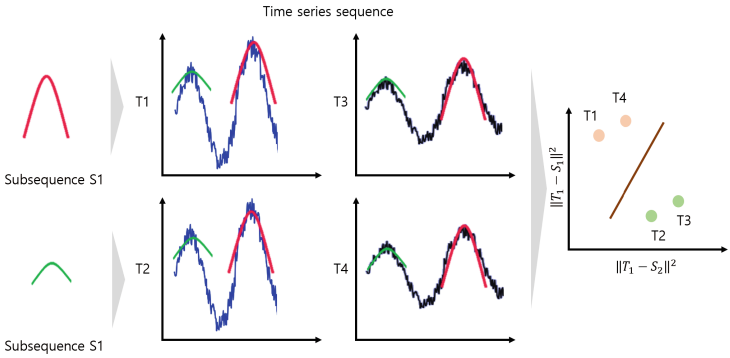

Fig. 1. Shapelet transformation

cases compared to other complex algorithms [14]. However, overfitting can occur if the same data are duplicated or under-fitting can occur if critical data are removed. A representative method of oversampling proposed to address this is the SMOTE (Synthetic Minority Oversampling Technique) algorithm proposed by Chawla et al. [6]. SMOTE algorithms have reduced the risk of overfitting by synthesizing existing data and are commonly used in many studies. Han et al. [7] proposed Borderline-SMOTE, which clarifies the classification boundary by applying the SMOTE algorithm only to the decision boundary portion of the class. Under sampling techniques are representative of NCL (Neighborhood Cleaning Rule) algorithms that remove data based on ENN (Edited Nearest Neighbor) algorithms [9]. Figure 2 show the under sampling which is the representative sampling method.

2.3 Ensemble Method

Ensemble in classification method means the way for reducing the risk of noise and overfitting and recording high classification performance by several iterative processes. A simple form of ensemble technique is the Bagging algorithm. The Bagging algorithm is a method of generating multiple samples of boot straps, and then applying the weak classifier to each sample to aggregate the results obtained to produce the final classification results according to the principle of majority decision. As with bagging, the Boosting algorithm generates multiple bootstrap samples and applies weak classifier to each one. And then, it evaluates the performance of each classification's results to provide further feedback to next sampling so that better results can be obtained. The Boosting method representatively includes the adaboost algorithm [11], SMOTEBoosting [12], and the Random Forest algorithm [13]. Figure 3 show the random forest algorithm which is the representative ensemble method.

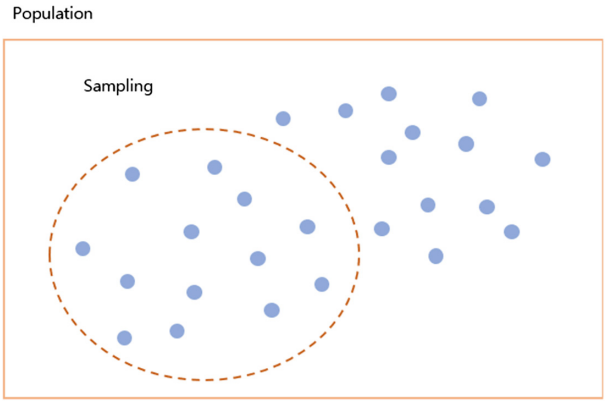

Fig. 2. Example of sampling method: under sampling

3 Under Sampling Adaboosting Feature Extraction

3.1 Overall Process

This paper proposes a feature extraction method that minimizes the risk of noisy data and overfitting. The process is expressed:

1. Initialize the sample's weights.
2. Based on the weight of the sample, sample it through under sampling.
3. After applying the shapelet classification to the generated data set, evaluate the performance and apply the shapelet transformation. Multiply the weight of the model by the weight of the corresponding distance data according to performance.
4. The sample weights are adjusted so that data with poor performance can be reflected in the next sampling by adding the weight of the sample.
5. After repeating the steps from 2 to 4 add all the distance data that was finally obtained and normalize it.

3.2 Feature Extraction

The feature extraction technique proposed in this paper is based on the adaboost algorithm and combined with the Shapelet Transformation. The adaboost algorithm is one of the ensemble techniques, which increases accuracy by repeatedly learning samples that are misclassified by allowing classification results to be reflected in the next sampling process. For under sampled dataset, apply the shapelet classification and shapelet transformation respectively, and evaluate the classification results to set the weights for the model and each sample. The weight of the sample is reflected in the next sampling process, and repeated learning gradually produces a highly accurate classifier. Finally, aggregate the

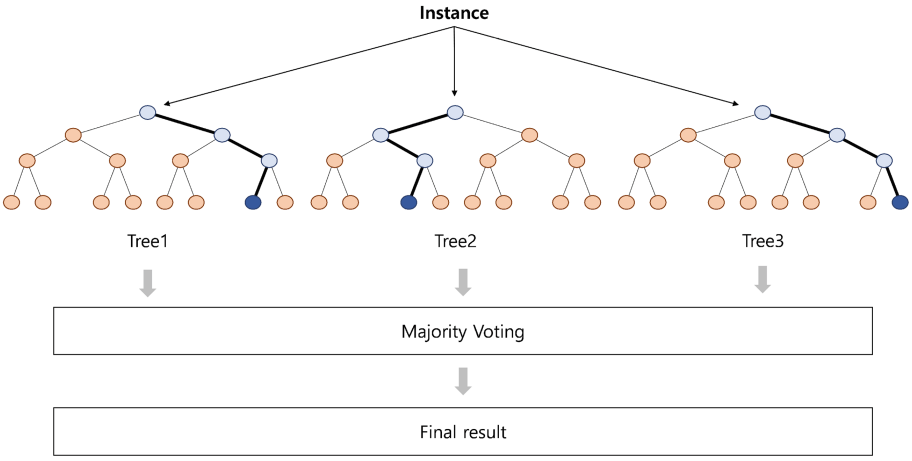

Fig. 3. Example of ensemble method: random forest

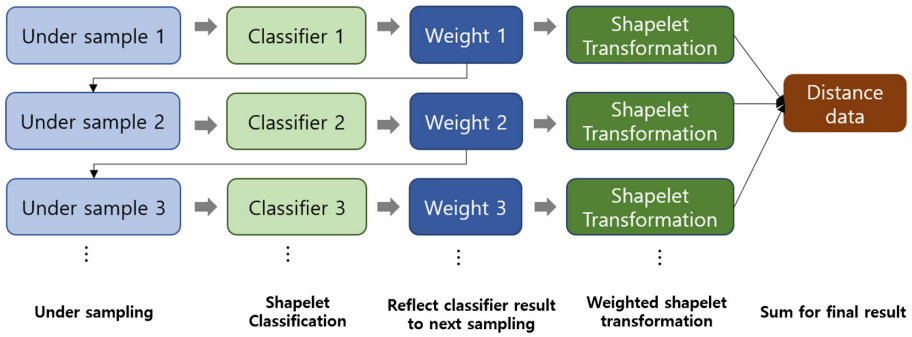

Fig. 4. Overall schematic of under sampling AdaBoosting algorithm

distances converted data according to the majority decision principle reflecting the weight of the model. The algorithm and its schematic are shown in Fig. 4.

The structure of the algorithm can be found in the following Algorithm 1:

The algorithm first receives training data, training data labels, test data, and parameter values as input. The parameter n means the number of repetitions of the ensemble, and l means the length of the shapelet. In this algorithm, training is performed on train data, and the transformation results is output to the test data. As a result, this experimental algorithm can compare the characteristics of the output test data with the actual *testLabel*.

The start of the algorithm begins by operating the for statement for the number of number of input parameter n. It execute the following command for each number of times of the for statement. First, under sampling is performed by inputting *trainTimeSeries* data and *trainLabel* to the random under sampling function. Second, we design the shapelet classification classifier by learn-

Algorithm 1 . Under Sampling Adaboosting Shapelet Transformation Algorithm

Input: Single time series training data *trainTimeseries*, *trainLabel*, *testTimeseries*, *parameter*$\{n, l\}$
Output: Distance data *trainDistance*, *testDistance*

```
for all i ← 1 to n do
    under_sample, under_sample_label ← RandomUnderSampling(trainTimeseries,
trainLabel, sample_weight)
    clf ← ShapeletClassification(under_sample, under_sample_label, l)
    valid ← clf.prediction(trainTimeSeries)
    error ← count(valid != trainLabel)
    error_rate ← error / len(trainLabel)
    Model_Weight[i] = log((1-error_rate) / error_rate)
    trainDistance ← clf.transform(trainTimeSeries)*Model_Weight[i]
    testDistance ← clf.transform(testTimeSeries)*Model_Weight[i]
    for all j ← 1 to len(trainTimeseries) do
        sample_weight[j] *= exp((-1)*Model_weight[i]*trainLabel[j]*clf[j])
        sample_weight[j] = sample_weight[j] / Z (Z : normalization factor)
    end for
end for
trainDistance ← trainDistance / Model_Weight.sum()
testDistance ← testDistance / Model_Weight.sum()
return trainDistance, testDistance
```

ing the sampled data. Third, *trainTimeSeries* data is input to the classifier to derive classification results. Fourth, the classification results is compared with the correct answer to calculated the error rate. Fifth, the weight of the model is calculated based on the error rate. The model weight are calculated as follows:

$$Model_Weight = \frac{1}{2} \ln \frac{1 - \epsilon_m}{\epsilon_m}$$

The ϵ means error rate. Sixth, the feature of the sample is extracted by multiplying the result of *trainDistance* and model shapelet transformation based on the weight of the model and the model learned by the corresponding sample. In the seventh, the weight of the sample to be reflected in the next sampling is calculated. The equation for the sample weights follows:

$$sample_weight = \frac{sample_weight * exp((-1) * MW * trainLabel * clf)}{Z}$$

where MW is Model_Weight and Z is normalization factor. The weights of the calculated samples are reflected in the random under sampling process during the ensemble process, and the learning is repeated. It is possible to continuously repeat the learning of samples whose classification performance is lacking due to lack of learning thereby enabling more complicated and precise learning. As a result, the final feature is returned by synthesizing repeated features.

This process is based on the principle of the adaboost algorithm. The adaboost algorithm evaluates the samples that were learned before in the ensemble process and reflects them in the next sampling process. Through this, it is possible to perform repeated learning for samples whose classification performance is poor due to lack of learning. Therefore, as the number of iterations increases, it is possible to design a precise classifier by repeating learning on data with a low learning rate. The finally designed classifier capable of recording high classification performance through repeated learning. Figure 5 show how adaboosting method can create effective classifier.

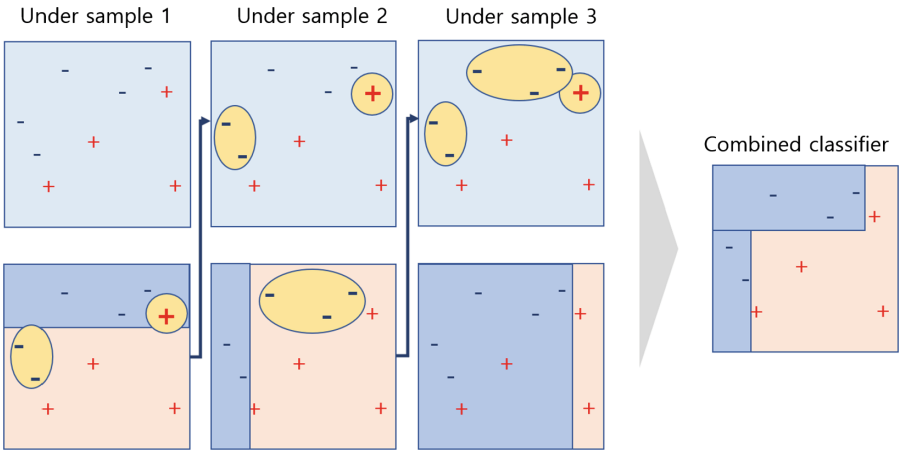

Fig. 5. Working process of adaboosting algorithm

In summary, this algorithm performs a shapelet classification on a partial sample made by Random Under Sampling. The performance of the classified results is then assessed to obtain the weight of the model according to its performance. Depending on the weight, time series data is converted to distance data through shapelet transformation. Depending on the results of the classification, the weight of the sample is reset and the weight is reflected in the next sample. This process is repeated n times so that the results are combined to return the final distance data.

Such under sampling and adaboosting techniques are known to have the effect of minimizing the influence of noise and overfitting according to the prior research [11]. In addition, it has been used in various studies because it can design strong classifiers through repeated learning and has excellent classification performance. Shapelet Transformation technique is also known to record excellent performance in classification of time series data. The algorithm proposed in this paper proposes feature extraction technique of time series data by combining algorithms that have been verified with excellent performance according to these previous studies. Therefore, it is expected that the performance of

the proposed algorithm will be improved to better performance than the existing methods.

4 Experimental Design and Implementation

In this paper, we focus the experimental design of our proposed algorithm rather than performance evaluation. To do this, we construct an experiment to test the performance of the proposed algorithm. The purpose of this experiment is to test whether the performance of this algorithm is superior to that of other algorithms for real time series data.

We design an experiment to verify the performance of the proposed algorithm. First, control groups are set up to identify performance advantages. As controls group, statistical techniques (average, standard deviation, maximum, minimum, etc.) and common shapelet transformation techniques, which are used in time series data feature extraction can be used. We apply various machine learning classification techniques (ex. Decision Tree, Random Forest, Support Vector Machine etc.) for the proposed algorithm and the two comparators and compare the classification results. The experiment process is as follows Fig. 6.

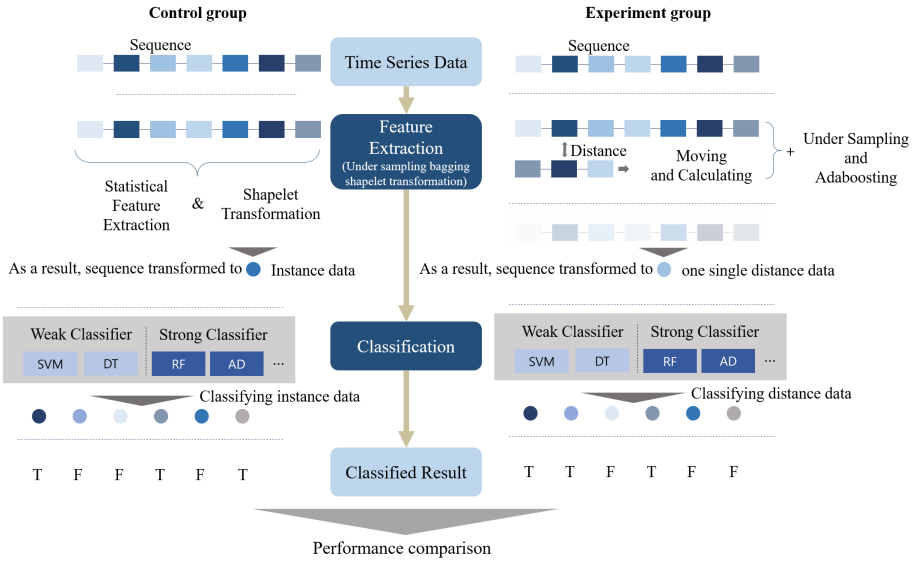

Fig. 6. Experiment process

4.1 Experiment Environment

We can use various time series data from open source to test our experiment. In this paper, we design an experiment using hydraulic system data including various time series data among several open source data. In this data, various

time series data collected from various kinds of sensors are included and classification experiments can be performed. By using this, time series data collected at various sampling rates can be used. Among them, 5 kinds of data (temperature, vibration, cooling power, etc.) are collected at 1 Hz. All data are labeled with stability 0 or 1, 0 means steady state, and 1 means non-steady state. The PC to be used for the experiment will be i7-8700K, and 16 GB of DDR4 RAM.

In general, evaluation results are confirmed by applying various evaluation indexes used in evaluation of classification algorithms. In the data classification problem, it is better to avoid using the accuracy because the imbalance between the data can cause the bias. In the designed experiments, performance is evaluated using G-mean and F-measure rather than accuracy. Here, the normal data labeled as "0" is referred to as "Negative", and the defect data labeled as "1" is referred to as "Positive". True Positive(TP) and True Negative(TN) are defined as when normal data is classified as normal and when the defect data are classified as defective. On the other hand, False Positive (FP) and False Negative (FN) are referred to when the defect data is classified as normal and when the normal data is classified as the defect, respectively. At this time, TP rate (TPR, Precision) and TN rate (TNR, Recall) can be calculated as follows.

$$TPR = TP/(TP + FN)$$

$$TNR = TP/(TP + FP)$$

G-mean is the geometric mean of TPR and TNR. Generally, it has a range of $[0, 1]$, and the higher the score, the higher the performance. F-measure means the harmonic mean of precision and recall. Likewise, the higher the score, the higher the performance Through TPR and TNR, G-mean and F-measure are defined as follows:

$$G\text{-}mean = \sqrt{TPR \times TNR}$$

$$F\text{-}measure = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

4.2 Configuration for Training and Test Data Set

We should distinguish five 1 Hz data from the hydraulic system mentioned above. 70% of the total data is classified as train, and 30% as test. At this time, cross-validation is performed to test the performance of the learning model. In general, there is a risk of overfitting in the case of classification and prediction models through machine learning techniques. While overfitting can result in high performance for a particular dataset, there is a risk that performance may drop dramatically if the test data changes. In order to prevent such a risk, we perform multiple cross validation by dividing the train data and the test data in the dataset. It is considered that the problem caused by overfitting is minimized when the same good performance is recorded in repeated cross validation. In this experiment, five(train, test) sets configurations are proposed by dividing each division criterion. This allows us to configure a set of data to apply multiple cross validation test. Figure 7 shows the data partitioning technique for cross validation.

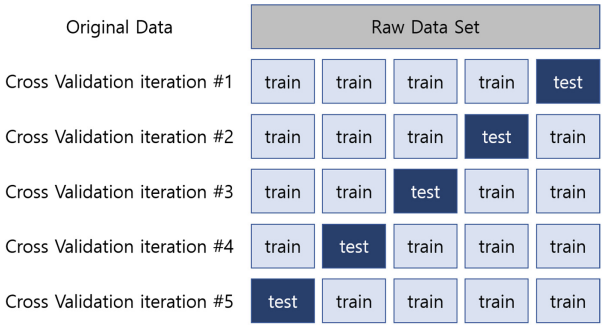

Fig. 7. Cross validation

4.3 Feature Extraction

The main contribution of this paper is the proposal of a new feature extraction algorithm. To verify the performance of the proposed algorithm, a control group and an experimental group are selected. As experimental group, the feature extracted through the algorithm described in this paper is specified. For the control group, feature extracted from 'statistical feature extraction' and 'shapelet transformation' are selected.

In this experiment, two techniques are used as a control group. The first is statistical feature extraction technique which is generally used in industrial field. The statistical feature extraction technique refers to a technique in which mean, standard deviation, and kurtosis are extracted as a feature of sampled signals. It is t he most commonly used technique because it is sample, computationally simple, and does not require high computing resources. As another control group, the shapelet transformation, which is the idea based on this paper, is selected. By comparing the performance with the shapelet transformation, we experiment to see how the proposed algorithm improves performance. Figure 8 shows the feature extraction technique of the control group and the feature extraction technique of the experimental group.

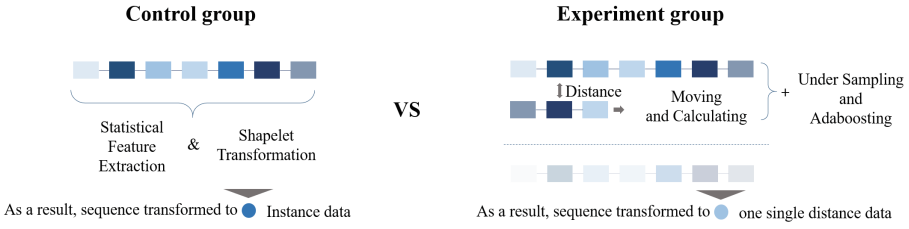

Fig. 8. Comparison of feature extraction methods

4.4 Classification

After applying various classification algorithms to the extracted results, the results are evaluated. This process is repeatedly applied to various classification algorithms, and it is confirmed whether the proposed algorithm has effective superiority. Classification algorithm can confirm performance by applying both weak classifier and strong classifier. We can use SVM (Supporter Vector Machine, RBF-SVM, Linear SVM), and DT (Decision Tree) which are commonly used in simple classification problem. The strong classifier is a classifier with a more sophisticated classification algorithm than the weak classifier. Typically, algorithms such as a random forest, adaboosting, and gradient boosting can be used, Each of the algorithms is evaluated as results of recording the optimum performance through parameter optimization. The classification results are evaluated according to the evaluation criteria described above, and it is judged whether the performance of the proposed algorithm is superior to the feature extraction technique selected as the control group.

This algorithm is thought to be able to prove its performance through experiments in *Python* environment. With the *tslearn* library provided by Python, we can implement the shapelet and the proposed algorithm, Under Sampling Adaboosting Shapelet Transformation. The various machine learning algorithms used for classification can also be used with the *sklearn* library.

5 Conclusion

Data from sensors such as mechanical equipment are far from ideal shapes. Factory data sets are basically very unbalanced because they are not prone to defects in a refined environment. Also, because of the variety of noise environments, the data includes a lot of noise. Therefore, using these acquired data requires time series processing techniques that are resistant to over-fitting and noise in the process. This paper proposes a feature extraction technique that has been developed by adding ensembles and sampling methods to a shapelet transformation that can produce a high classification effect for time series data. For the proposed Under Sampling Adaboosting Shapelet transformation, repeated ensemble techniques are expected to be added to the normal shapelet to produce more effective results for noise and overfitting. However, the adaboost technique has the disadvantage of spending a lot of time on the operation because parallel processing is not supported compared to the one that requires a lot of computations. In the following studies, to address this problem, a dataset is divided and algorithms are redesigned to allow partial parallelism.

Acknowledgments. This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-2018-0-01417) supervised by the IITP (Institute for Information & communications Technology Promotion). This work has supported by the Gyeonggi Techno Park grant funded by the Gyeonggi-Do government (No. Y181802).

References

1. Helwig, N., Pignanelli, E., Schutze, A.: Condition monitoring of a complex hydraulic system using multivariate statistics. In: Proceedings of IEEE Instrumentation and Measurement Technology Conference (I2MTC), pp. 210–215, July 2015
2. Zhu, L., Lu, C., Dong, Z.Y., Hong, C.: Imbalance learning machine based power system short term voltage stability assessment. IEEE Trans. Ind. Inform. **13**, 2533–2543 (2017)
3. Ye, L., Keogh, E.: Time series shapelets: a new primitive for data mining. In: Proceedings of 15th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, pp. 947–955, June 2009
4. Lines, J., Davis, L.M., Hills, J., et al.: A shapelet transform for time series classification. In: Proceedings of 18th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, pp. 289–297, August 2012
5. Zhu, L., Lu, C., Sun, Y.: Time series shapelet classification based online short term voltage stability assessment. IEEE Trans. Power Syst. **31**(2), 1430–1439 (2016)
6. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
7. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005). https://doi.org/10.1007/11538059_91
8. He, H., Yai, B., Garcia, E., Li, S.: ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of IEEE International Joint Conference on Neural Networks, pp. 1322–1328, June 2008
9. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. In: Quaglini, S., Barahona, P., Andreassen, S. (eds.) AIME 2001. LNCS (LNAI), vol. 2101, pp. 63–66. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-48229-6_9
10. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
11. Freund, Y., Schapire, R.E.: A decision theoretic generalization of online learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
12. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: SMOTEBoost: improving prediction of the minority class in boosting. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 107–119. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39804-2_12
13. Breiman, L.: Random forest. Mach. Learn. **45**(1), 5–32 (2001)
14. Lee, T., Lee, K., Kim, C.: Performance of machine learning algorithms for classimbalanced process fault detection problems. IEEE Trans. Seminocuctor Manuf. **29**(4), 436–445 (2016)