Improving Cascade Classifier Precision by Instance Selection and Outlier Generation

Judith Neugebauer, Oliver Kramer and Michael Sonnenschein
Department of Computing Science, Carl von Ossietzky University Oldenburg, Oldenburg, Germany

Keywords: Time Series Classification, High-dimensional Classification, Imbalanced Learning, Data Preprocessing.

Abstract: Beside the curse of dimensionality and imbalanced classes, unfavorable data distributions can hamper classification accuracy. This is particularly problematic with increasing dimensionality of the classification task. A classifier that can handle high-dimensional and imbalanced data sets is the cascade classification method for time series. The cascade classifier can compound unfavorable data distributions by projecting the high-dimensional data set onto low-dimensional subsets. A classifier is trained for each of the low-dimensional data subsets and their predictions are aggregated to an overall result. For the cascade classifier, the errors of each classifier accumulate in the overall result and therefore small improvements in each small classifier can improve the classification accuracy. Therefore we propose two methods for data preprocessing to improve the cascade classifier. The first method is instance selection, a technique to select representative examples for the classification task. Furthermore, artificial infeasible examples can improve classification performance. Even if high-dimensional infeasible examples are available, their projection to low-dimensional space is not possible due to projection errors. We propose a second data preprocessing method for generating artificial infeasible examples in low-dimensional space. We show for micro Combined Heat and Power plant power production time series and an artificial and complex data set that the proposed data preprocessing methods increase the performance of the cascade classifier by increasing the selectivity of the learned decision boundaries.

1 INTRODUCTION

Classification of high-dimensional data sets with imbalanced or even severely imbalanced classes is influenced by the curse of dimensionality. This is also true for time series classification tasks, where the ordering of the features (time steps) is important, (Bagnall et al., 2012). Such tasks can be e.g., energy time series, where neighboring time steps are correlated. For these high dimensional time series classification tasks with imbalanced classes we have proposed the cascade classification model (Neugebauer et al., 2015). This model employs a cascade of classifiers based on features of overlapping time series steps. Therefore the high-dimensional feasible time series are projected on all neighboring pairs of time steps. In the low-dimensional space of the data subsets, the curse of dimensionality is no longer a problem.

Classification performance depends strongly on the distribution of the underlying data set, (Lin and Chen, 2013). Therefore, an improvement of the data distribution could improve classification performance. Time series classification tasks with a cascade classifier have mainly two reason for unfavorable data distributions. Beside the original often not homogeneous distribution of the time series in feature space, the projection of feasible time series leads to an inhomogeneous distribution in low-dimensional space. A selection of more homogeneously distributed feasible examples (instances) would lead to an improvement in classification performance for a constant number of training examples or decrease the number of training examples, that are necessary to achieve a certain classification performance. In this paper we propose to resample feasible low-dimensional examples based on the distance to their nearest neighbor. If the distance is greater than a certain threshold, the respective example is part of the new more homogeneous set.

Additionally, infeasible examples can further improve the classification performance by increasing the selectivity of the decision boundaries, (Zhuang and Dai, 2006). If there are enough infeasible examples, binary classification can be applied and yield better results than one-class classification, see (Bellinger et al., 2012). But even if there are infeasible examples available in high-dimensional space, they can not

Neugebauer, J., Kramer, O. and Sonnenschein, M.
Improving Cascade Classifier Precision by Instance Selection and Outlier Generation.
DOI: 10.5220/0005702100960104
In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence (ICAART 2016) - Volume 2*, pages 96-104
ISBN: 978-989-758-172-4

be used for training of the low-dimensional classifiers. Energy time series e.g., are only feasible, if all time steps are feasible. Due to this property infeasible power production time series projected to low-dimensional space can be located in the region of feasible ones.

Since projection of high-dimensional infeasible examples does not work, we propose a sampling procedure for artificial infeasible examples for the low-dimensional data subsets. Sampling of artificial infeasible examples is based on minimal distances to the nearest feasible neighbor. The infeasible examples are generated near the class boundary to improve the selectivity of the classifiers.

This paper is structured as follows. In Sect. 2, we provide an overview on related work, on instance selection and on generation of artificial infeasible examples (outliers). In Sect. 3 we describe the cascade classification approach and in Sect. 4 we introduce our data preprocessing methods to improve the cascade classifier. In Sect. 5, we compare the classification performance of the cascade approach with and without data preprocessing in an experimental study. This study is conducted on simulated micro combined heat and power plant (μCHP) data and an artificial complex data set. In Sect. 6, we summarize and draw conclusions.

2 RELATED WORK

In classification tasks, a lot of problems often arise due to not optimally distributed data, like not representative data samples or inhomogeneously distributed samples.

For the cascade classifier, (Neugebauer et al., 2015), the projection of the feasible examples from high to low-dimensional space leads to additional inhomogeneity in the distribution of feasible examples. Unfavorable data distributions hamper classification, (Lin and Chen, 2013). But data preprocessing methods that select representative examples from the data set and maintain the integrity of the original data set while reducing the data set can help to overcome the classification problems. Depending on the data distribution and the application several instance selection (also called record reduction / numerosity reduction / prototype selection) approaches have been developed. Beside data compression and classification performance improvement instance selection also works as noise filter and prototype selector, (Tsai et al., 2013; Blachnik, 2014; Wilson and Martinez, 2000). In the last years, several instance selection approaches have been proposed and an overview can be found e.g., in (Jankowski and Grochowski, 2004), (Liu et al., 2001), (Garcia et al., 2012). Based on these algorithms advanced instance selection algorithms e.g based on ensembles, (Blachnik, 2014), genetic algorithms, (Tsai et al., 2013) or instance selection for time series classification with hubs, (Tomašev et al., 2015) were developed. But all these instance selection approaches have more or less high computational complexity, because they are developed for d-dimensional data sets, while the cascade classifier has several similar structured data subsets in low-dimensional space. Therefore, we propose a simple and fast instance selection method for low-dimensional space.

As far as infeasible examples (outliers, counter examples) can improve (one-class) classification, (Zhuang and Dai, 2006), algorithms to sample infeasible examples have been proposed. One such algorithm generates counter examples around the feasible class based on points near the class boundary, (Bánhalmi et al., 2007). Another algorithm by (Tax and Duin, 2002) can sample outliers from a hyperbox or a hypersphere, that cover the target object (feasible class). The artificial infeasible examples of these algorithms have either high computational complexity or contain some feasible examples. But the cascade classifier requires a fast and simple sampling approach for all low-dimensional data subsets, where the generated infeasible examples are located in the region of the infeasible class. Thus we propose an artificial outlier generation method for the data subsets of the cascade classifier.

3 CASCADE OF OVERLAPPING FEATURE CLASSIFIERS

In this section, we introduce the cascade approach for time series classification (Neugebauer et al., 2015). As the classification of the high-dimensional time series is difficult, a step-wise classifier has been proposed. The cascade classification model is developed for high-dimensional binary time series classification tasks with (severely) imbalanced classes. The small interesting class is surrounded by the other class. Both classes fill together a hypervolume, e.g. a hypercube. Furthermore the cascade classifier requires data sets with clearly separable classes, where the small interesting class has a strong correlation between neighboring features (time steps). The low-dimensional data subsets of the small class should preferably employ only one concept (cluster) and a shape, that can be easily learned.

The model consists of a cascade of classifiers,

each based on two neighboring time series steps (features) with a feature overlap between classifiers. The cascade approach works as follows. Let $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$ be a training set of N time series $\mathbf{x}_i = (x_i^1, x_i^2, \ldots, x_i^d)^T \in \mathcal{R}^d$ of d time steps and $y_i \in \{+1, -1\}$ the information about their feasibility. For each 2-dimensional training set

$$((x_1^j, x_1^{j+1}), y_1), \ldots, ((x_N^j, x_N^{j+1}), y_N) \quad (1)$$

a classifier is trained. All $d-1$ classification tasks can be solved with arbitrary baseline classifiers, depending on the given data. Single classifiers employ similarly structured data spaces and thus less effort is needed for parameter tuning. Most of the times only feasible low-dimensional examples are available and in this case baseline classifiers from one-class classification are suitable. The predictions $f_1, \ldots, f_{d-1}$ of all $d-1$ classifiers are aggregated to a final result $F(\cdot)$ for a time series $\mathbf{x}$. A new time series $\mathbf{x}$ is feasible, only if all classifiers in the cascade predict each time step as feasible:

$$F(\mathbf{x}) = \begin{cases} +1 & \text{if } f_i \neq -1 \ \forall i = 1, \ldots, d-1 \\ -1 & \text{else} \end{cases} \quad (2)$$

The cascade classification approach can be modified and extended, e.g., concerning the length of the time series intervals, respectively the dimensionality of the low-dimensional data subsets.

4 DATA PREPROCESSING METHODS

In this section the selection of feasible examples and sampling of artificial infeasible examples is presented. These data preprocessing methods for the low-dimensional (2-dimensional) data subsets of the cascade classifier require data with clearly separable classes and 2-dimensional feasible data subsets in the same value ranges. If the feasible 2-dimensional examples employ different value ranges they have to be scaled. Preferably the high-dimensional data set is scaled to values between 0 and 1. For some data sets, where the 2-dimensional subsets are very different in shape and size, each subset has to be scaled individually. Achieving the same value range for all low-dimensional data subsets is necessary for the application of the same parameters on all subsets. Just like the dimensionality of the low-dimensional subsets of the cascade approach could be changed, the proposed data preprocessing methods could be also applied to data subsets of other dimensionality.

4.1 Selection of Feasible Examples

Selection of feasible examples leads to more homogeneously distributed feasible examples as in the original distribution of the low-dimensional data subsets, see Fig. 2. Here the selection of feasible examples increases the point density in the upper right corner and decreases the point density in the lower left corner, see Fig. 2(b) in comparison to the original distribution shown in Fig. 2(a). We propose a sampling Algorithm 1, based on a minimal distance δ of feasible examples to their nearest feasible neighbors.

Algorithm 1: Selection of feasible examples.

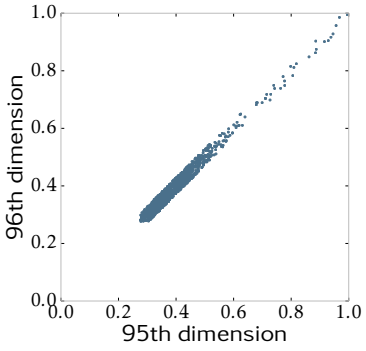
Require: 2-dimensional data set $\mathbf{X}$ with n feasible examples
choose t start examples S from $\mathbf{X}$
repeat
  choose t new examples E from $\mathbf{X}$
  calculate euclidean distance δ of E to their nearest neighbors in S
  if $\delta \geq \varepsilon$ **then**
    append respective examples to S
  end if
until all n examples are processed
shuffle S

Figure 1: Pseudocode for the selection of feasible examples. The minimal distance ε between feasible nearest neighbors depends on the data set.

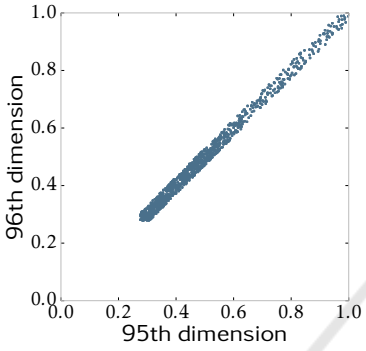
The distribution of the feasible examples can differ a little in homogeneity and shape among the 2-dimensional data subset despite previous scaling. Therefore the parameters of the procedure have to be adapted carefully, especially the minimum distance ε of new examples (E) to the nearest selected neighbors (S). Preferably, ε is selected in such a way, that the selection of feasible examples yields round about the number of examples required for training and validation. Such ε values yielded in pre-tests good classification results, because the resampled data sets maintain the integrity of the original data subsets best for the desired number of training and validation examples.

4.2 Sampling of Infeasible Examples Near the Class Boundaries

The sampling procedure of artificial infeasible 2-dimensional examples near the class boundaries, see Fig. 3, requires more or less homogeneously distributed feasible 2-dimensional examples that represent the whole feasible region.



(a) Initial distribution



(b) Resampled features

Figure 2: 1000 examples of the 95*th* and 96*th* dimensions of the feasible class of the μCHP data set (initial and resampled).

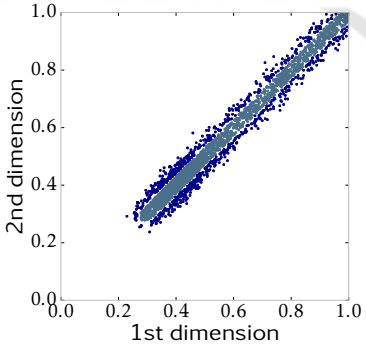


Figure 3: Resampled examples of the 1*st* and 2*nd* dimension of the feasible class of the μCHP data set with artificial infeasible examples. The feasible class shown as gray points is surrounded by artificial infeasible examples (blue points).

The better the distribution of the feasible examples, the better will be the distribution of the artificially generated infeasible examples. But if the projection of the high-dimensional data set to the 2-dimensional data sets exhibits a projection error, like e.g., a hypersphere data set, see (Neugebauer et al., 2015), then the artificial infeasible examples are not located near the true class boundary, but near the decision boundary learned by the cascade classifier. We propose to sample low-dimensional artificial infeasible examples by disturbing 2-dimensional feasible examples and identifying new infeasible instances (Γ) with the help of a certain minimal distance (δ_b) to their nearest feasible neighbors, see Algorithm 2.

Algorithm 2: Sampling of infeasible examples.

Require: 2-dimensional data set $\mathbf{X}$ with n feasible examples, where the distance between infeasibles and their feasible nearest neighbors is $\leq \varepsilon_b$ in about 95% of cases

1: $Y = \mathbf{X} + \mathcal{N}(\mu, \sigma)) \cdot \alpha$
2: calculate euclidean distance δ_b of all examples in Y to their nearest neighbors in X
3: **if** $\delta_b \geq \varepsilon_b$ **then**
4: examples are infeasible examples (Γ)
5: **end if**
6: **repeat**
7: $Y = \Gamma + \mathcal{N}(\mu, \sigma) \cdot \alpha$
8: calculate euclidean distance δ_b of all examples in Y to their nearest neighbors in X
9: **if** $\delta_b \geq \varepsilon_b$ **then**
10: append example to Γ
11: **end if**
12: **until** number of examples in Γ is sufficient
13: shuffle Γ

Figure 4: Pseudocode for sampling of artificial infeasible examples in 2-dimensional space. The factor α for the standard normal distribution $\mathcal{N}(\mu, \sigma)$ and the minimal distance between feasible examples and their nearest infeasible neighbors ε_b depend on the data set.

This procedure turned out to be parameter-sensitive. The minimal distance between feasible and infeasible examples ε_b has to be larger than the minimal distance ε between the selected feasible examples and preferably also larger than the longest distance between feasible nearest neighbors. The closer the infeasible examples are located to the class boundary, the greater is the improvement of classification specificity. But the closer the infeasible examples are located to the class boundary, the higher is the probability, that these artificial infeasible examples could be located in the region of the feasible class. This phenomenon can hamper classification improvement by artificial infeasible examples. Therefore a very careful parametrization of the algorithm is necessary.

5 EXPERIMENTAL STUDY

In this section, the effect of the proposed data preprocessing methods on the performance of the cascade classification approach is evaluated on two data sets. The first data set is an energy time series data set micro combined heat and power plant (μCHP) power production time series. The second data set is an artificial complex data set where the small interesting class has a *Hyperbanana* shape. *Banana* and *Hyperbanana* data sets are often used to test new classifiers, because they are considered as difficult classification tasks. Therefore we take the test with the *Hyperbanana* data set as a meaningful result.

The experimental study is done with cascade classifiers on each data set. Altogether three classification experiments are conducted on both data sets. The first experiment is done without preprocessing (no prepro.), the second with selected feasible examples (fs) and the third with selected feasibles and artificial infeasible examples (fs + infs). For all experiments a one-class baseline classifier is used. The third experiment is also done with binary baseline classifiers.

The experimental study is divided into a description of the data sets, the experimental setup and the results.

5.1 Data Sets

The experiments are conducted with simulated μCHP power output time series and an artificial *Hyperbanana* data set. Both data sets have 96 dimensions (time steps, resp. features).

5.1.1 μCHP

A μCHP is a small decentralized power and heat generation unit. The μCHP power production time series are simulated with a μCHP simulation model[1]. The μCHP simulation model includes a μCHP model, a thermal buffer and the thermal demand of a building. A μCHP can be operated in different modes, where its technical constraints, the constraints of the thermal buffer and the conditions of the thermal demand of the building are complied. Power output time series can be either feasible or infeasible depending on these constraints. The μCHP simulation model calculates the power production time series for feasible operation modes, but also infeasible power output time series can be generated, where at least one constraint is violated. Due to the different constraints the class of feasible power production time series consists of several clusters. For convenience only such feasible power output time series are chosen, where the power production is greater than 0 at each time step. Infeasible power output time series are sampled from the whole volume of the infeasible class. In data space the class of infeasible power output time series occupies a much larger volume than the class of feasible ones, (Bremer et al., 2010). The classes are severely imbalanced, but the experiments are conducted with equal numbers of examples from both classes.

The feasible and infeasible μCHP power output time series are scaled according to the maximal electrical power production to values between 0 and 1.

5.1.2 Hyperbanana

As far as there is now 96-dimensional *Hyperbanana* data set, we have generated a data set from the extended d-dimensional Rosenbrock function, (Shang and Qiu, 2006).

$$f(x) = \sum_{i=1}^{d-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2] \qquad (3)$$

The small and interesting class, or here also called feasible class is sampled from the Rosenbrock valley with $f(x) < 100$ and the infeasible class with $f(x) >= 100$ is sampled only near the class boundary to test the sensitivity of the decision boundaries of the classifiers.

Sampling of the banana shaped valley is done by disturbing the minimum of the extended 96-dimensional Rosenbrock function with normally distributed values ($\mathcal{N}(0,1) \cdot \beta$ with $\beta \in \{40, 50, 60, 70\}$). The minima of the Rosenbrock function are presented in (Shang and Qiu, 2006) for different dimensionalities, but the minimum for 96 dimensions is missing. Therefore we approximated the minimum with regard to the other minima with -0.99 for the first dimension and 0.99 for all other dimensions. The procedure of disturbing and selecting values from the Rosenbrock valley is repeated with the sampled values until enough data points are found. As far as it is difficult to sample the banana "arms" all at the same time, we sampled them separately by generating points that are $<$ or $>$ than a certain value and continued sampling by repeating disturbance and selection with these values.

Values from all these repetitions were aggregated to one data set and shuffled. Finally all dimensions (features) x_i of the data set are scaled to values between 0 and 1 by $x_i = [x_i + (\min(x_i)+\text{offset})]/[\max(x_i)+\text{offset}-\min(x_i)+\text{offset}]$ with offset $= 0.2$.

[1]Data are available for download on our department website http://www.uni-oldenburg.de/informatik/ui/forschung/themen/cascade/.

The samples generated by this procedure are not homogeneously distributed in the Rosenbrock valley and they do not represent all *Hyperbanana* "arms" equally.

The 96-dimensional infeasible examples near the class boundary are sampled in the same way as the feasible ones but starting with the feasible *Hyperbanana* samples and with $100 \leq f(x) \leq 500$.

5.2 Experimental Setting

The experimental setting is divided into two parts: data preprocessing and classification. All calculations are done in Python. The first part, data preprocessing (selection of feasible examples and generation of infeasible examples) is done according to Sect. 4.1 and Sect. 4.2.

Selection of feasible examples is parametrized differently for both data sets as a result of pre-studies. The pre-studies were conducted with different minimal distances ε and ε_b and evaluated according to the number of resulting examples and their distribution in the 2-dimensional data subset. For the μCHP data set instance selection is parametrized as follows, the minimal distance between feasible examples is set to $\varepsilon = 0.001$ and the number of new examples used for each iteration t is set to $t = 1000$. Generation of artificial infeasible examples is parameterized with $n = 15000$ initially feasible examples disturbance $= \mathcal{N}(0, 0.01) \cdot \alpha$ with $\alpha = 1$ and minimal distance between infeasible examples and their nearest feasible neighbors $\varepsilon_b = 0.025$. For the *Hyperbanana* data set the instance selection parameters are set to $\varepsilon = 0.002$ and $t = 1000$ and parameters for generating artificial infeasible examples are set to $n = 20000$, disturbance $= \mathcal{N}(0, 0.02) \cdot \alpha$ with $\alpha = 1$ and $\varepsilon_b = 0.002$.

The second part of the experimental study, the three classification experiments, are done with the cascade classifier, see Sect. 3, with different baseline classifiers from SCIKIT-LEARN, (Pedregosa et al., 2011), a One-Class SVM (OCSVM) and two binary classifiers, k-nearest neighbors (kNN) and Support Vector Machines (SVMs). The OCSVM baseline classifier is used for all three experiments. The two binary classifiers kNN and binary SVM are used for the third experiment with both preprocessing methods (fs + infs).

All experiments are conducted identically on both data sets except for the parametrization. For all experiments the number of feasible training examples N is varied in the range of $N = \{1000, 2000, \ldots, 5000\}$ for the μCHP data set and $N = \{1000, 2000, \ldots, 10000\}$ for the *Hyperbanana* data set. For binary classification N infeasible examples are added to the N feasible training examples.

Parameter optimization is done with grid-search on separate validation sets with the same number of feasible examples N as the training sets and also N artificial infeasible examples for the third experiment. For the first experiment (no prepro.) and the second experiment (fs) the parameters are optimized according to true positive rates (TP rate or only TP), (TP rate = (true positives) / (number of feasible examples)).

For the third experiment, where the validation is done with N additional infeasible examples, parameters are optimized according to accuracy (acc = (true positives + true negatives)/(number of positive examples + number of negative examples)). The OCSVM parameters are optimized in the ranges $\nu \in \{0.0001, 0.0005, 0.001, 0.002, \ldots, 0.009, 0.01\}$, $\gamma \in \{50, 60, \ldots, 200\}$, the SVM parameters in $C \in \{1, 10, 50, 100, 500, 1000, 2000\}$, $\gamma \in \{1, 5, 10, 15, 20\}$ and the kNN parameter in $k \in \{1, 2, \ldots, 26\}$.

Evaluation of the trained classifiers is done on a separate independent data set with 10000 feasible and 10000 real infeasible 96-dimensional examples according to TP and TN rates for varying numbers of training examples N. The classification results could be evaluated with more advanced measures, see e.g. (He and Garcia, 2009; Japkowicz, 2013). For better comparability of the results on both data sets and the option to distinguish effects on the classification of feasible and infeasible examples we use the simple TP and TN rates. TN rates on both data sets are difficult to compare, because the infeasible μCHP power output time series are distributed in the whole region of infeasible examples, while the infeasible *Hyperbanana* examples are distributed only near the class boundary. As far as most classification errors occur near the class boundary, the TN rates of the *Hyperbanana* set are expected to be lower than the TN rates on the μCHP data set.

5.3 Results

The proposed data preprocessing methods, selection of feasible examples and generation of artificial infeasible examples show an increase in classification performance of the cascade classifier in the experiments.

On both data sets (μCHP and *Hyperbanana*) data preprocessing leads to more precise decision boundaries than without data preprocessing, see Fig. 5 and Fig. 7. This can be also seen in the TP and TN rates of the classification results, see Fig. 6 and Fig. 8.

For the μCHP data set, all three experiments lead to TN rates of 1, therefore only the TP rates are plotted in Fig. 6. But high TN rates for the μCHP data set

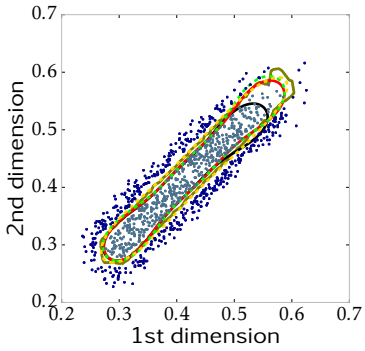


Figure 5: Decision boundaries on the 1st and 2nd dimension of the μCHP trained with $N = 1000$ feasible (+ $N = 1000$ infeasible) training examples, no prepro. (dashed black), fs (dashed green), OCSVM(fs + infs) (red), kNN(fs + infs) (olive) and SVM(fs + infs) (yellow). The gray points indicate 500 of the selected feasible training examples and the blue points 500 of the artificial infeasible examples.

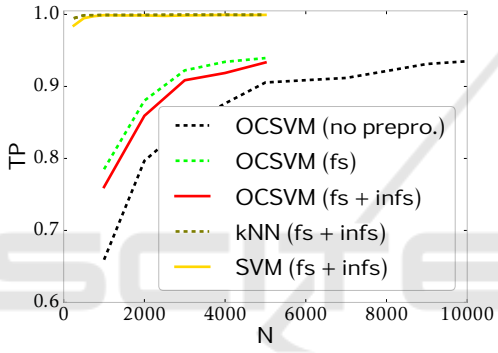
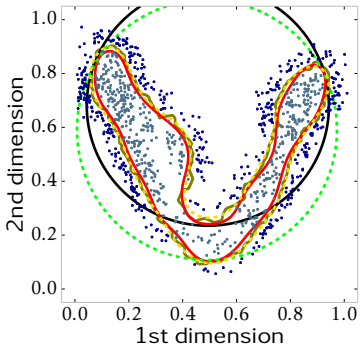


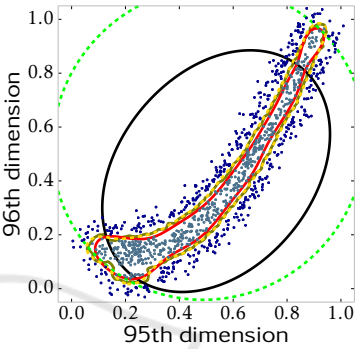
Figure 6: TP rates on the μCHP data set for different preprocessing steps and different baseline classifiers.

do not necessarily mean, that further infeasible time series are classified correctly. The applied infeasible test examples are taken from the whole volume of the large infeasible class and therefore most of the examples are not located near the class boundary. The first experiment without data preprocessing (no prepro.) yields the lowest TP rates of all experiments for all numbers of training values N and the second experiment with selection of feasible examples (fs) leads already to higher TP rates. The third experiment with selection of feasible examples and artificial infeasible examples (fs + infs) leads to different results with the OCSVM baseline classifier and the binary SVM and kNN baseline classifiers. While the OCSVM(fs + infs) achieves slightly lower TP rates than OCSVM(fs) in the second experiment, the binary baseline classifiers SVM(fs + infs) and kNN(fs + infs) achieve TP rates near 1.

For the *Hyperbanana* data set with a more complex data structure, data preprocessing influences the TP rates, see Fig. 8(a) and the TN rates, Fig. 8(b) of the classification results. In the first experiment (no



(a) $2d$-boundaries on dim. 1/2

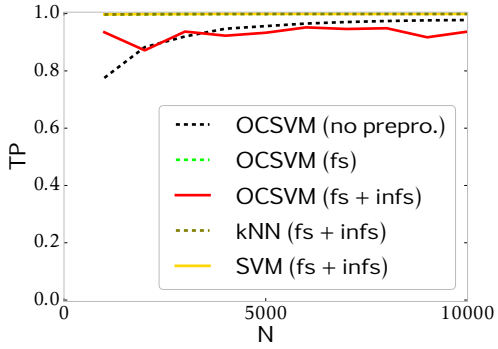


(b) $2d$-boundaries on dim. 95/96

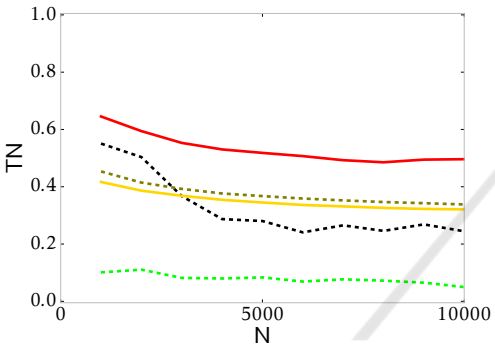
Figure 7: Decision boundaries on the *Hyperbanana* data set trained with $N = 1000$ feasible (+ $N = 1000$ infeasible) training examples, no prepro. (dashed black), fs (dashed green), OCSVM(fs + infs) (red), kNN(fs + infs) (olive) and SVM(fs + infs) (yellow). The gray points indicate 500 of the selected feasible training examples and the blue points 500 of the artificial infeasible examples.

prepro.) and second experiment (fs) the classification achieves relatively high TP rates and at the same time the lowest TN rates of all experiments due to too large decision boundaries, see Fig. 7. The third experiment (fs + infs) revealed an opposed behavior of the OCSVM baseline classifier and the SVM and kNN baseline classifiers. The OCSVM(fs + infs) achieves lower TP rates than the OCSVM in the previous experiments but also the highest TN rates of all experiments. SVM and kNN baseline classifiers with (fs + infs) achieve the highest TP rates of all experiments and at the same time lower TN rates than the OCSVM(fs + infs).

In summary, data preprocessing increases the classification performance of the cascade classifier on both data sets. While the selection of feasible examples increases the classification performance, artificial infeasible examples can lead to an even greater increase depending on the data set and the baseline classifier.



(a) TP rates on a differently preprocessed *Hyperbanana* set



(b) TN rates on a differently preprocessed *Hyperbanana* set

Figure 8: TP and TN rates on the *Hyperbanana* data set for different preprocessing steps and different baseline classifiers. The legend in Fig. 8(a) is also valid for Fig. 8(b). The green line of OCSVM(fs) in Fig. 8(a) is covered by the olive and the yellow lines.

6 CONCLUSIONS

In this paper, we proposed two data preprocessing methods to improve the performance of the cascade classification model (selection of feasible examples and generation of artificial infeasible examples). In the experimental study, we showed for a μCHP power output time series data set and an artificial and complex *Hyperbanana* data set, that data preprocessing increases the performance of the cascade classifier. Selection of feasible examples leads to more representative training data and artificial infeasible examples lead to more precise decision boundaries of the low-dimensional classifiers. Depending on the data set and the baseline classifier, the application of both data preprocessing methods yields the best classification performance. The application of only one data preprocessing method (selection of feasible examples) and no data preprocessing yielded always worse results, lower TP rates on the μCHP data set and especially very low TN rates on the *Hyperbanana* data set.

In summary, the proposed data preprocessing methods for the cascade classifier are very sensitive concerning the parametrization, but a careful parameter choice increases the classification performance.

We plan to generalize our cascade classification model in future work in such a way, that it can deal with data sets with more complex data structures, e.g., the small and interesting class consists of several clusters or the low-dimensional data subsets employ a data structure that can not be learned easily like a butterfly-like shape.

Furthermore, we intend to evaluate the proposed data preprocessing methods on such data sets.

ACKNOWLEDGEMENT

This work was funded by the Ministry for Science and Culture of Lower Saxony with the PhD program System Integration of Renewable Energy (SEE).

REFERENCES

Bagnall, A., Davis, L. M., Hills, J., and Lines, J. (2012). Transformation based ensembles for time series classification. In *Proceedings of the Twelfth SIAM International Conference on Data Mining, Anaheim, California, USA, April 26-28, 2012.*, pages 307–318.

Bánhalmi, A., Kocsor, A., and Busa-Fekete, R. (2007). Counter-example generation-based one-class classification. In Kok, J. N., Koronacki, J., Mantaras, R. L., Matwin, S., Mladenič, D., and Skowron, A., editors, *Machine Learning: ECML 2007*, volume 4701 of *Lecture Notes in Computer Science*, pages 543–550. Springer Berlin Heidelberg.

Bellinger, C., Sharma, S., and Japkowicz, N. (2012). One-class versus binary classification: Which and when? In *Machine Learning and Applications: ICMLA, 2012 11th International Conference on*, volume 2, pages 102–106.

Blachnik, M. (2014). Ensembles of instance selection methods based on feature subset. *Procedia Computer Science*, 35(0):388 – 396. Knowledge-Based and Intelligent Information & Engineering Systems 18th Annual Conference, KES-2014 Gdynia, Poland, September 2014 Proceedings.

Bremer, J., Rapp, B., and Sonnenschein, M. (2010). Support vector based encoding of distributed energy resources' feasible load spaces. In *Innovative Smart Grid Technologies Conference Europe IEEE PES*.

Garcia, S., Derrac, J., Cano, J., and Herrera, F. (2012). Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transac-*

tions on Pattern Analysis and Machine Intelligence, 34(3):417–435.

He, H. and Garcia, E. (2009). Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on*, 21(9):1263–1284.

Jankowski, N. and Grochowski, M. (2004). Comparison of instances seletion algorithms i. algorithms survey. In Rutkowski, L., Siekmann, J., Tadeusiewicz, R., and Zadeh, L., editors, *Artificial Intelligence and Soft Computing - ICAISC 2004*, volume 3070 of *Lecture Notes in Computer Science*, pages 598–603. Springer Berlin Heidelberg.

Japkowicz, N. (2013). *Assessment Metrics for Imbalanced Learning*, pages 187–206. John Wiley & Sons, Inc.

Lin, W.-J. and Chen, J. J. (2013). Class-imbalanced classifiers for high-dimensional data. *Briefings in Bioinformatics*, 14(1):13–26.

Liu, H., Motoda, H., Gu, B., Hu, F., Reeves, C. R., and Bush, D. R. (2001). *Instance Selection and Construction for Data Mining*, volume 608 of *The Springer International Series in Engineering and Computer Science*. Springer US, 1 edition.

Neugebauer, J., Kramer, O., and Sonnenschein, M. (2015). Classification cascades of overlapping feature ensembles for energy time series data. In Woon, W. L., Aung, Z., and Madnick, S., editors, *Data Analytics for Renewable Energy Integration*. Springer. in print.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Shang, Y.-W. and Qiu, Y.-H. (2006). A note on the extended rosenbrock function. *Evol. Comput.*, 14(1):119–126.

Tax, D. M. J. and Duin, R. P. W. (2002). Uniform object generation for optimizing one-class classifiers. *J. Mach. Learn. Res.*, 2:155–173.

Tomašev, N., Buza, K., Marussy, K., and Kis, P. B. (2015). Hubness-aware classification, instance selection and feature construction: Survey and extensions to time-series. In Stańczyk, U. and Jain, L. C., editors, *Feature Selection for Data and Pattern Recognition*, volume 584 of *Studies in Computational Intelligence*, pages 231–262. Springer Berlin Heidelberg.

Tsai, C.-F., Eberle, W., and Chu, C.-Y. (2013). Genetic algorithms in feature and instance selection. *Knowledge-Based Systems*, 39(0):240–247.

Wilson, D. and Martinez, T. (2000). Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286.

Zhuang, L. and Dai, H. (2006). Parameter optimization of kernel-based one-class classifier on imbalance text learning. In Yang, Q. and Webb, G., editors, *PRICAI 2006: Trends in Artificial Intelligence*, volume 4099 of *Lecture Notes in Computer Science*, pages 434–443. Springer Berlin Heidelberg.