# Graph-based Feature Selection Filter Utilizing Maximal Cliques

Daniel Thilo Schroeder*†, Kevin Styp-Rekowski†, Florian Schmidt†, Alexander Acker† and Odej Kao†
*SimulaMet, Oslo, Norway
Email: daniels@simula.no
†Complex and Distributed IT-Systems Group, TU Berlin, Berlin, Germany
Email: {firstname}.{lastname}@tu-berlin.de

***Abstract*— Huge amounts of data are collected every millisecond all around the world. This ranges from images and videos to an increasing amount of sensor data. Thus, it gets difficult for humans to decide on the most important features anymore. But reducing the feature vector is an important and necessary task to achieve higher precision in classification tasks. Detecting anomalies and classifying data points is crucial for a variety of objectives in many domains. Therefore, this work focuses on feature selection for binary decision problems (e.g. anomaly detection, binary classification). We propose a novel graph-based feature selection filter, which takes into account both the importance and correlation of features at the same time. The graph-based feature selection filter recommends a subset by applying a rating function onto the maximal cliques of the graph. The evaluation is based on a comparison of the accuracy of multiple machine learning algorithms and datasets between different baseline feature selection approaches and the proposed approach. Results show that the proposed approach delivers the highest accuracy in about 69% of the cases compared to existing approaches, while reducing the number of features.**

***Index Terms*—feature selection; filter method; machine learning**

## I. Introduction

Classification problems are ubiquitous in many different domains such as medicine[1], anomaly detection [2], or resource management [3]. In practice, it is often difficult to choose a combination of features to describe an underlying problem in such a way that classifiers deliver highly accurate results. Therefore, in many cases all features that appear relevant are taken into account. This can lead to the existence of irrelevant or redundant features. Irrelevant features are by definition not related with the target concept but affect the classification process [4] and redundant features have intrinsically equal meaning, making it difficult to deliver high accurate results for a given classification algorithm. If datasets contain irrelevant or redundant features, performance, training time, precision and memory usage may be affected and over-fitting can be caused. For these reasons, it is important to eliminate irrelevant or redundant features. Such a task is called feature selection. More precisely feature selection is the process of selecting a subset of features with significant impact on a prediction result. As the number of all possible combinations of features within a subset is exponentially high, it is obvious that brute force is not an appropriate approach to solve this problem.

Furthermore, feature selection is suited for supervised [5] or unsupervised [6] learning. In this work, we consider feature selection for supervised learning, which can be further distinguished into three main categories [7], [8]: Wrapper methods, filter methods and embedded models. Wrapper methods employ a classification algorithm results to determine suitable features with most accurate prediction results [7]. Here in many cases, the classification algorithm is similar to the aimed classifier. Consequently, results are often optimized for this particular classifier. Filter methods examine a dataset's intrinsic properties prior to the classification [9]. This means feature selection and classifiers are separated and therefore, feature selection bias is reduced towards specific classification algorithms. Embedded models combine wrapper and filter methods. They interact with a classifier but also filter and are therefore less complex to compute than wrapper methods [4].

In summary, filter methods can be applied within a wide range of different fields and problem spaces, while wrapper methods focus on an individual machine learning approach. In this paper, we present a framework for developing filter methods for feature selection on binary classification problems based on graph structures. The key idea is to represent the importance of individual features as well as the redundancy of pair-wise features within a single graph structure. Based on this data structure, we provide a graph-based filtering approach, selecting a subset of features. The proposed approach combines feature selection and graph theory and thus enables a new perspective on feature selection in general.

Our paper makes the following key contributions:

- Definition of a generic graph-based feature selection framework.
- Description of a reference implementation and hyperparameter selection.
- Evaluation on four different datasets using four different classification approaches to validate the accuracy of the recommended feature sets. Furthermore, we evaluate against two reference approaches from the machine learning framework WEKA [10] and additionally brute forced optimal feature sets.

The rest of the paper is organized as follows. Section II provides information about related work. Afterwards, we continue to present our generic graph-based feature selection

framework in Section III. Section IV provides information about the specific configurations for the previously introduced framework which is evaluated and discussed in Section V. Finally, Section VI concludes this paper.

## II. Related Work

The graph-based feature selection approach, discussed within this work assumes labeled training data for any binary classification problem. As described above, we focus on filter based methods also within the related work. In the past, several different filter based feature selection approaches were proposed. We highlight some within this section, mostly related and relevant for this work. For further reading, we recommend to look at the extensive review of Tang et al. [4].

In the area of statistical subset selection for binary decision problems, Christ et al. introduced FRESH, an algorithm for time series classification and regression including feature selection [11]. The significance of each feature vector for classification is individually determined. This results in a vector of p-values of the Kolmogorov-Smirnov test (KS-test) [12], which is evaluated on basis of the Benjamini-Yekutieli procedure [13] to decide on the most important features.

Besides the KS-test, Biesiada and Duch [14] showed the applicability to combine the KS-test with correlation measurements. Thus, the correlation is used to describe the symmetrical uncertainty as a first stage filter, followed by a KS-test filter to remove redundancy and to determine the relevance of features. Our method also combines the positive properties of correlation and statistical relevance tests like KS-test, while extending the technique to use graph-based filtering as a single filtering step. Nie et al. [15] proposed a graph-based feature selection technique which provides an automatic solution to generate a redundancy matrix. They also showed the applicability of graph-based methods in the field of feature selection. Thus, we aim to combine those methods.

Furthermore, feature selection approaches are nowadays highly relevant for e.g. the area of AutoML, where automatic optimization of machine learning algorithms are performed and feature selection is a relevant part of it [16]–[18]. Therefore, we focus on applied techniques from the machine learning framework WEKA [10], which is used within the WEKA-AutoML [16] and includes widely used feature select techniques, which are used as baseline methods to evaluate our proposed approach against.

## III. Graph-based Feature Filtering

The central idea of the proposed approach is to choose a representation that allows to combine the importance of individual features along with the redundancy of feature pairs. As shown in Figure 1, the algorithm is organized in the three phases: *Graph Building*, *Graph Filtering* and *Graph Selection*. Furthermore, Figure 2 depicts an example of the procedure described in the following paragraph.

### A. Graph Building

During the *Graph Building* phase, a complete weighted undirected graph $G = (V, E)$ is created, where $V$ is the set of vertices and $E$ denotes the set of edges. Here, each vertex $v \in V$ represents a feature along with its importance for an arbitrary binary classification problem which is defined by a weighting function $g : V \rightarrow \mathbb{R}$. In addition, each edge $e \in E$ connects two vertices $\{v_i, v_j\}$ , where $v_i \neq v_j, v_i, v_j \in V$. Here, we define an edge weight, representing the redundancy of the connected feature vertices, by a similarity function $s : E \rightarrow \mathbb{R}$.

### B. Graph Filtering

Throughout the *Graph Filtering* a function $f_G = (f_E \circ f_V)$ reduces the graph size by applying a filter criteria to the edges $f_E$ and vertices $f_V$ of $G$. This separation allows independent removal of nodes that are less important or removal of edges that connect redundant nodes.

### C. Graph Selection

The final *Graph Selection* consists of three steps. First, the Bron-Kerbosch-Algorithm is applied to find all cliques within the filtered graph. Thereby, a clique is defined as a complete subgraph $C \subseteq G$ [19]. Second, a set $C'$ of maximal cliques $C_i^{max}$ is determined, whereby a maximal clique is a complete subgraph $C_i \subseteq G$ which is not a proper subset of another complete subgraph $C_j \subseteq G$ [19].

Third, whenever there is $|C_i^{max}| > 1$, where $C_i^{max} \in C'$, a rating function $r : C_i^{max} \rightarrow \mathbb{R}$, $\forall C_i^{max} \in C'$ is applied and the maximal clique with the highest score is selected.

## IV. Reference Implementation and Hyperparameter Configurations

Next, the definition and functionality of the graph-based feature selection framework is explained.

### A. Graph Building

Here, an optimized version [20], [21] of the KS-distance that returns the maximum distance between the cumulative distributions for each class was chosen for the weighting function $g$.

Furthermore, the Pearson correlation

$$\rho = \frac{cov(v_1, v_2)}{\sigma_{v_1} \sigma_{v_2}} \tag{1}$$

is used as redundancy function $s$. For a pair of vertices $v_1, v_2 \in V$ the covariance is determined and divided by the product of their standard deviations. This means that the resulting Pearson coefficient $\rho \in \mathbb{R}$ lies in the interval of $[-1, 1]$. Here, zero indicates that the features represented by $v_1, v_2$ are not related to each other. The larger the absolute value of the correlation, the higher the correlation between those two features. Thus, we assume that a Pearson coefficient close to zero is an indicator for good combinations of features or contrary a measurement for the redundancy of features.

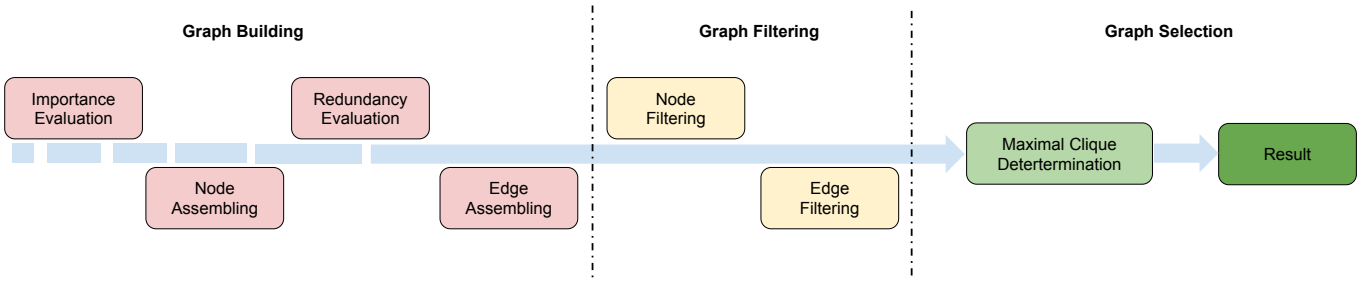


Fig. 1. Life cycle of a graph-based feature selection. The Selection is divided into the three main phases, *Graph Building*, *Graph Filtering* and *Graph Selection*. In the *Graph Building phase*, first, the importance of each individual feature is evaluated (*Importance Evaluation*) to assemble the graph nodes (*Node Assembling*). The same is done analogously for edges and redundancy (*Redundancy Evaluation, Edge Assembling*). In *Graph Filtering* the remaining nodes and edges are selected before the maximal cliques are passed to the selection function in the last step.

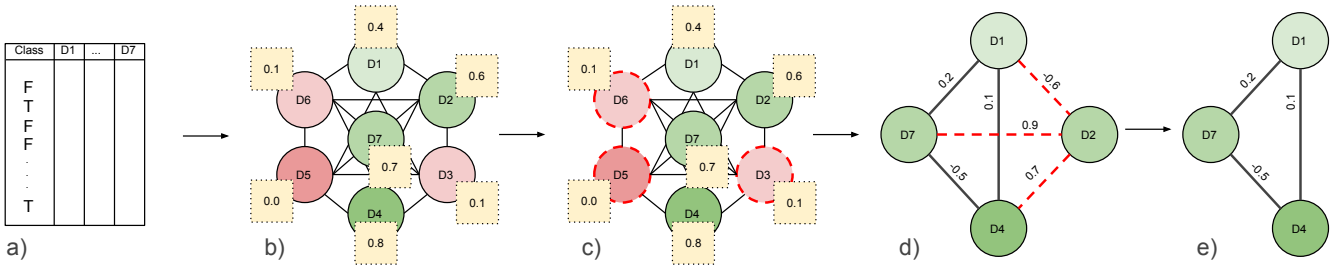


Fig. 2. Shows an example of a graph-based feature selection life cycle (see Figure 1). Here, $a$ shows the dataset followed by the composite graph after the *Graph Building* phase ($b$). Figures $c$ and $d$ depict the node and edge filter. In this example only one maximal clique is found ($e$).

### B. Graph Filtering

As stated in Section III-B the filter functions $f_E$ and $f_V$ have to be defined as parameters.

One possibility is to evaluate the chosen parameters on a part of the dataset, the validation dataset, but this data needs to be collected first and held back for validation. This comes with the cost of a bias due to the fact that the classification algorithm influences hyperparameters to deliver better results and therefore leads to a different set of hyperparameters for each application. Therefore, a lightweight alternative in form of a heuristic was introduced. The heuristic filters the vertices of the graph with the resulting computational complexity as a goal. Therefore, all vertices with an assigned KS-distance $k \leq 0.1$ are filtered out in order to only retrieve features with meaningful additions to the binary classification problem. Afterwards, the heuristic ensures a limit of 19 vertices with the highest KS-distances. This limit ensures fast computation as well as a remaining set of meaningful features regarding to the corresponding classification problem.

In order to filter the edges, which correspond to the correlation between features, a second heuristic has been implemented. This heuristic sorts the edge set according to their corresponding correlation and rejects the 10% with the lowest correlation value. In addition to that, only edges with correlation values in $[-0.9, 0.9]$ remain in the edge set, ensuring all edges representing highly correlating features are removed.

Thus, the heuristic is able to reduce the size of the graph by applying at least a proportional reduction. If necessary, the heuristic is capable of reducing the size of the graph further by checking for very high correlating features. In a dataset, where a lot of features correlate, this can lead to a strong reduction.

### C. Graph Selection

After reducing the graph size during the *Graph Filtering*, a set of maximal cliques $C_{max}$ is calculated. Throughout *Graph Selection*, the rating function $r$ is applied to select the most valuable maximal clique $c_{max} \in C_{max}$ and thus the most valuable set of features.

Figure 4 depicts the Pearson correlation, KS-distance and accuracy for the classifier Naive Bayes on the shuttle dataset. Here, the coloration corresponds to the precision, whereby red indicated more accuracy and blue less. It is obvious that both Pearson correlation as well as KS-distance influence precision, but also the combination induces higher accurate results.

Since we received similar outcomes for the remaining datasets and classifiers, we assume that correlation and distance in general influence the accuracy. Therefore, we choose the selection function $s$ to be

$$s(C_{max}) := 1 - \overline{|c|} + \overline{k}, \tag{2}$$

where $\overline{|c|} \in [0, 1]$ is the absolute correlation average and $\overline{k} \in [0, 1]$ the distance average. Values towards 2 indicate more recommendable results. Finally, the maximal clique $c$ with the highest value for $s(c)$ is selected.

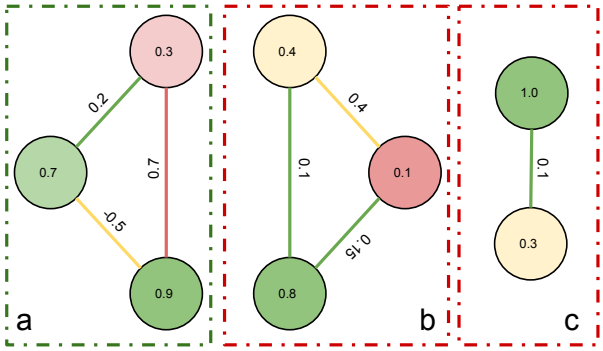


Fig. 3. gives an example for the *Graph Selection*. The three graphs are the return of the *Graph Filtering* phase. Graph $c$ is not a maximal clique and is therefore not considered. For $s(a) \approx 1.17$ while $s(b) \approx 1.21$ meaning that a will be the selected maximal clique and therefore the selected set of features

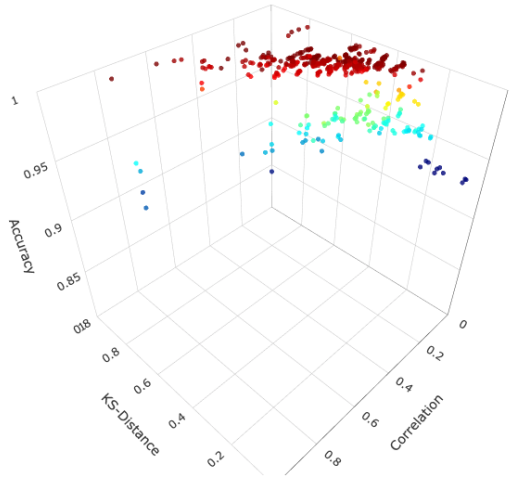


Fig. 4. Correlation and KS-distance to accuracy for Naive Bayes on Shuttle data. It can be seen that both correlation and distance influence precision. This observation is the reason to choose a $s(C_{max})$ in which both correlation and distance play a role.

## V. Evaluation

In this section, the evaluation setup, the parameter selection as well as the results of the evaluation will be discussed.

### A. Evaluation Setup

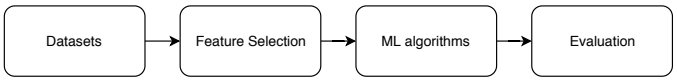


Fig. 5. Flow diagram of evaluation setup.

The evaluation setup is a classic machine learning pipeline as can be seen in Figure 5. Several datasets will be investigated, on which our feature selection approach will be applied in comparison with other approaches. The focus of this work lies on the feature selection. Thus, neither the machine learning algorithms nor their parameters have been modified or optimized. In addition to that, no preprocessing was applied. The main measurement for the quality of the chosen feature subset will be the accuracy of the resulting machine learning models.

In order to achieve meaningful comparison results, several approaches have been investigated. For each dataset the baseline contains the anomaly rate of the dataset, the full set of features as well as the optimal solution (identified through exhausted search). The anomaly rate of the baseline corresponds to a classifier, which labels every data point as normal, the full feature set uses all features when evaluating the ML algorithms. The optimal solution could only be computed on smaller datasets, where all possible subsets of features were investigated in terms of their resulting accuracy. For larger datasets, Monte Carlo methods were used to estimate the optimal solution. In addition to that, we compare our approach against two established feature selection methods from the widely used WEKA machine learning framework, namely the Information Gain Attribute Ranking and a Correlation-Based Feature Selection with a Greedy Stepwise Search [22].

For each measurement a 10-fold cross-validation has been conducted. Therefore, the average accuracy among all runs has been calculated and an overfitting could be prevented. Additionally, the number of features within the selected feature set was measured. The smaller the number of features in the resulting subset is, the faster machine learning algorithms can train and predict as their runtime depends on the input data.

### B. Datasets

Overall, four datasets from the UCI machine learning repository have been used, which contain labeled data divided into two classes using the transformations from the Stonybrook's Outlier Detection Datasets [23], [24]. The datasets were chosen with respect to different properties like number of data points, number of features and complexity of the classification task. In the following, a short description for each of the datasets will be given.

*1) Glass Identification:* This dataset is the result of the chemical examination of different kinds of glass and contains data of their chemical ingredients [25]. There are 9 metrics for 214 data points of which 9 are outliers (4.2%).

*2) Wine:* This dataset has arisen from the comparison of chemical components within several wines from 3 different cultivars from which one is chosen as the outlier. There are 13 metrics for 178 data points of which 10 are outliers (7.7%). The glass and wine dataset have been chosen to show that the approach works generally.

*3) Shuttle:* This dataset contains metrics recorded by a shuttle. There are 9 metrics for 49097 data points of which 3511 are outliers (7%). Here, we can see how the approaches behave, when investigating a dataset containing larger sets of data points.

*4) Ionosphere:* This dataset contains radar data about the ionosphere [26]. There are 34 metrics for 351 data points of which 126 are outliers (36%). By investigating this dataset more insights on the behavior of the different algorithms can be gained, when larger sets of metrics are present.

### C. ML algorithms

Four machine learning algorithms have been evaluated in order to receive an accuracy result from a variety of different machine learning approaches. Therefore, Naive Bayes (NB), K-Nearest-Neighbors (KNN), Support Vector Machines (SVM) and Logistic Regression (LR) have been chosen, as they represent some of the major kinds of classification algorithms. As stated previously, standard implementations of these algorithms from the WEKA library (version 3.8.3) were used without further optimization [10].

### D. Results

The resulting tables show the number of features (#F) in the resulting feature subset together with the accuracy (Acc) for the different machine learning algorithms. For each accuracy value, the Root Mean Squared Error (RMSE) is given as well, arising from the 10-fold cross validation. In addition to that, for each ML algorithm the best accuracy values are marked boldly in order to show which feature selection algorithm performed best.

Table I and Table II show the results of the evaluation for the Glass Identifiaction and the Wine dataset. These datasets have small sizes of data points and metrics and were chosen in order to show how the approach compares to other approaches.

It can be seen that the proposed approach outperforms or performs equivalent on these two datasets for almost all ML algorithms, the only exception being the LR algorithm applied to the Wine dataset. Remarkably, the WEKA-based Greedy-stepwise search failed for the glass dataset as no feature after the first one could directly add value to the accuracy. Only considering multiple features at once leads to improvements in the accuracy for this dataset. Here, the graph-based approach could outperform the existing approach as it considers all features at the same time.

Table III visualizes the result for the shuttle dataset. Here, the optimal solution was marked as it was estimated. This was needed as the dataset was too large to be fully investigated. The graph-based approach achieved similar results to the existing approaches. It was not able to outperform the other approaches but also achieved high accuracy results while maintaining a low number of features. It can be noted that no approach for itself was able to achieve dominant results for this dataset.

The result for the 4th dataset can be seen in Table IV. Because this dataset had a very large feature vector, there is a large search space for feature subsets. Hence, the optimal solution was also estimated using Monte Carlo methods. For this dataset the proposed approach shows very good results as it outperforms or performs similarly as the existing approaches for almost every ML algorithm. In addition to that, the feature subset utilizes only 12 features which is the smallest value of all the approaches and favorable.

All experiments were performed on a 2.3 GHz Intel Core i5 processor (7360U) with the computation time of the approach depicted in Figure 6. Here, the complete approach from receiving all features until receiving the selected feature subset has been evaluated. The parameter which limits the maximum number of vertices in the Node Filtering phase has been used to evaluate the capability of the approach. As expected, computation time grows exponentially due to the fact that the maximal clique problem is NP-hard. However, the approach is applicable with low computational costs for datasets with a dimension vector of size 19 or lower without any further optimizations.

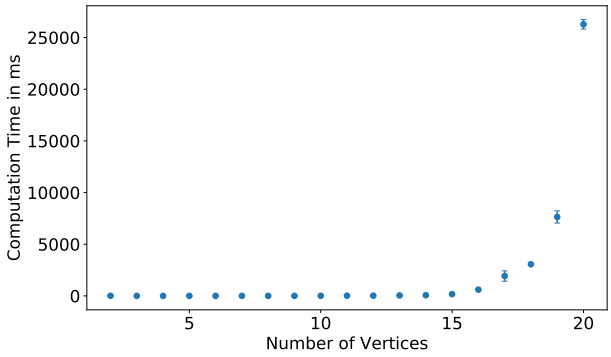


Fig. 6. Mean Computation times of 10 runs with their standard deviation.

Overall, it could be shown that the proposed approach delivers strong results, which can oftentimes outperform existing approaches while reducing the number of features in the feature subset. For the ML algorithms NB, KNN and SVM the algorithm performed best in 3 out of 4 datasets, for the algorithm LR best in 2 out of 4. In summary, the approach performed best in $11/16 \approx 69\%$ of the examined cases. This shows that this approach is a valid alternative to existing approaches. Compared to the results of the complete feature set, it can be seen that the evaluation supports the assumption that reducing the feature subset consistently leads to better results in terms of accuracy. Compared to the optimal solutions, the graph-based approach achieves almost as high accuracy as the global optimum, but never less than $2.81\%$ for all ML algorithms and datasets.

TABLE I
GLASS DATASET RESULTS

| Algorithm | #F | NB | KNN | SVM | LR |
|---|---|---|---|---|---|
| | | Acc. \| RMSE | Acc. \| RMSE | Acc. \| RMSE | Acc. \| RMSE |
| Baseline | | 95.79 | | | |
| All-Features | 9 | 96.26 \| 0.19 | 97.20 \| 0.17 | 95.79 \| 0.21 | 98.13 \| 0.13 |
| Optimal | div. | 98.13 \| 0.14 | 99.07 \| 0.10 | 95.79 \| 0.21 | 99.53 \| 0.08 |
| Weka-Ranker | 3 | 87.85 \| 0.30 | 96.26 \| 0.19 | **95.79** \| 0.21 | 95.33 \| 0.17 |
| Weka-Greed | **1** | 88.32 \| 0.21 | 95.79 \| 0.18 | **95.79** \| 0.21 | 95.79 \| 0.17 |
| Graph-Based | 7 | **96.26** \| 0.19 | **98.13** \| 0.14 | **95.79** \| 0.21 | **98.60** \| 0.10 |

TABLE II
WINE DATASET RESULTS

| Algorithm | #F | NB | KNN | SVM | LR |
|---|---|---|---|---|---|
| | | Acc. \| RMSE | Acc. \| RMSE | Acc. \| RMSE | Acc. \| RMSE |
| Baseline | | 66.85 | | | |
| All-Features | 13 | 95.51 \| 0.17 | 97.75 \| 0.15 | 96.63 \| 0.18 | 97.19 \| 0.16 |
| Optimal | div. | 99.44 \| 0.10 | 98.88 \| 0.11 | 98.31 \| 0.13 | 100.00 \| 0.00 |
| Weka-Ranker | 13 | 95.51 \| 0.17 | **97.75** \| 0.15 | **96.63** \| 0.18 | 97.19 \| 0.16 |
| Weka-Greed | 10 | 97.75 \| 0.14 | 97.19 \| 0.17 | 95.51 \| 0.21 | **98.31** \| 0.13 |
| Graph-Based | **9** | **99.44** \| 0.09 | **97.75** \| 0.15 | **96.63** \| 0.18 | 97.19 \| 0.16 |

TABLE III
SHUTTLE DATASET RESULTS

| Algorithm | #F | NB | KNN | SVM | LR |
|---|---|---|---|---|---|
| | | Acc. \| RMSE | Acc. \| RMSE | Acc. \| RMSE | Acc. \| RMSE |
| Baseline | | 92.849 | | | |
| All-Features | 9 | 99.295 \| 0.08 | 99.943 \| 0.02 | 99.603 \| 0.06 | 99.648 \| 0.05 |
| Optimal** | div. | 99.666 \| 0.06 | 99.945 \| 0.02 | 99.631 \| 0.06 | 99.648 \| 0.05 |
| Weka-Ranker | 9 | 99.295 \| 0.08 | **99.943** \| 0.02 | 99.603 \| 0.06 | **99.648** \| 0.06 |
| Weka-Greed | **2** | **99.540** \| 0.07 | 99.884 \| 0.03 | **99.631** \| 0.06 | 99.627 \| 0.07 |
| Graph-Based | 5 | 99.401 \| 0.07 | 99.845 \| 0.04 | 99.605 \| 0.06 | **99.648** \| 0.06 |

TABLE IV
IONOSPHERE DATASET RESULTS

| Algorithm | #F | NB | KNN | SVM | LR |
|---|---|---|---|---|---|
| | | Acc. \| RMSE | Acc. \| RMSE | Acc. \| RMSE | Acc. \| RMSE |
| Baseline | | 64.10 | | | |
| All-Features | 34 | 82.62 \| 0.40 | 87.75 \| 0.35 | 86.61 \| 0.37 | 87.18 \| 0.32 |
| Optimal** | div. | 92.59 \| 0.27 | 92.88 \| 0.27 | 90.03 \| 0.32 | 90.02 \| 0.30 |
| Weka-Ranker | 33 | 82.62 \| 0.40 | 87.75 \| 0.35 | 86.61 \| 0.37 | 87.18 \| 0.32 |
| Weka-Greed | 14 | **91.45** \| 0.27 | 90.31 \| 0.31 | **87.75** \| 0.35 | **89.17** \| 0.29 |
| Graph-Based | **12** | **91.45** \| 0.27 | **90.60** \| 0.31 | **87.75** \| 0.35 | 88.89 \| 0.30 |

## VI. CONCLUSION

We proposed a novel generic graph-based feature selection filtering method for binary classification problems, which combines importance- and redundancy evaluations with maximal clique search in order to find a suitable subset of features. Based on the defined generic approach, we showed a reference implementation of all relevant parts including Pearson correlation and KS-distances as statistical functions in order to build the weighted graph. Furthermore, we evaluated the given method against open source available feature selection techniques from the WEKA machine learning framework and exhausted search to find the optimal solution on several different datasets. The results show that the proposed approach achieves high accuracy, while reducing the number of features on all given datasets. Compared to related methods, the graph-based feature selection outperforms on many datasets or shows similar qualitative results using less number of features. All in all, we showed the applicability of the graph-based filtering framework for feature selection.

As future work, we like to concentrate on unsupervised threshold identification for the graph-based filtering and the usage of further statistical information to enlarge the information diversity of the graph structure. Additionally, a greedy based implementations for finding maximal weighted cliques may avoid the problem of NP-complete runtime. Thus, the given approach can be applied to big data. Ongoing work are also an extension of the evaluation in both further datasets and related methods to provide practical guidance of selecting suitable feature selection methods in applied domains.

## REFERENCES


[1] S. A. Hicks, K. Pogorelov, T. de Lange, M. Lux, M. Jeppsson, K. R. Randel, S. Eskeland, P. Halvorsen, and M. Riegler, "Comprehensible reasoning and automated reporting of medical examinations based on deep learning analysis," in *Proceedings of the 9th ACM Multimedia Systems Conference*, ser. MMSys '18. New York, NY, USA: ACM, 2018, pp. 490–493. [Online]. Available: http://doi.acm.org/10.1145/3204949.3208113

[2] F. Schmidt, A. Gulenko, M. Wallschläger, A. Acker, V. Hennig, F. Liu, and O. Kao, "Iftm-unsupervised anomaly detection for virtualized network function services," in *2018 IEEE International Conference on Web Services (ICWS)*. IEEE, 2018, pp. 187–194.

[3] L. Thamsen, B. Rabier, F. Schmidt, T. Renner, and O. Kao, "Scheduling recurring distributed dataflow jobs based on resource utilization and interference," in *2017 IEEE International Congress on Big Data (BigData Congress)*. IEEE, 2017, pp. 145–152.

[4] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," *Data classification: algorithms and applications*, p. 37, 2014.

[5] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping, "Use of the zero-norm with linear models and kernel methods," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1439–1461, 2003.

[6] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *Journal of machine learning research*, vol. 5, no. Aug, pp. 845–889, 2004.

[7] M. Gutlein, E. Frank, M. Hall, and A. Karwath, "Large-scale attribute selection using wrappers," in *2009 IEEE symposium on computational intelligence and data mining*. IEEE, 2009, pp. 332–339.

[8] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[9] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Advances in neural information processing systems*, 2006, pp. 507–514.

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[11] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package)," *Neurocomputing*, vol. 307, pp. 72–77, 2018.

[12] I. Chakravarti, R. Laha, and J. Roy, "Kolmogorov-smirnov (ks) test," *Handbook of methods of applied Statistics*, vol. 1, pp. 392–394, 1967.

[13] Y. Benjamini, D. Yekutieli *et al.*, "The control of the false discovery rate in multiple testing under dependency," *The annals of statistics*, vol. 29, no. 4, pp. 1165–1188, 2001.

[14] J. Biesiada and W. Duch, "Feature selection for high-dimensional data: A kolmogorov-smirnov correlation-based filter," in *Computer Recognition Systems*. Springer, 2005, pp. 95–103.

[15] F. Nie, W. Zhu, and X. Li, "Unsupervised feature selection with structured graph optimization," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[16] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-weka: Combined selection and hyperparameter optimization of classification algorithms," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 847–855.

[17] H. Jin, Q. Song, and X. Hu. (2018) Auto-keras: An efficient neural architecture search system.

[18] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Advances in neural information processing systems*, 2015, pp. 2962–2970.

[19] C. Bron and J. Kerbosch, "Algorithm 457: finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[20] G. Marsaglia, W. W. Tsang, and J. Wang, "Evaluating kolmogorov's distribution," *Journal of Statistical Software, Articles*, vol. 8, no. 18, pp. 1–4, 2003. [Online]. Available: https://www.jstatsoft.org/v008/i18

[21] R. Simard and P. L'Ecuyer, "Computing the two-sided kolmogorov-smirnov distribution," *Journal of Statistical Software, Articles*, vol. 39, no. 11, pp. 1–18, 2011. [Online]. Available: https://www.jstatsoft.org/v039/i11

[22] M. A. Hall and G. Holmes, "Benchmarking attribute selection techniques for discrete class data mining," 2002.

[23] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[24] S. Rayana, "ODDS library," 2016. [Online]. Available: http://odds.cs.stonybrook.edu

[25] I. W. Evett and E. J. Spiehler, "Rule induction in forensic science," Central Research Establishment, Home Office Forensic Science Service, Tech. Rep., 1987.

[26] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker, "Classification of radar returns from the ionosphere using neural networks," *Johns Hopkins APL Tech. Dig*, vol. vol. 10, pp. 262–266, 1989, in.