Computers, Materials & Continua
DOI: 10.32604/cmc.2022.025196
Article

Tech Science Press

Improving Association Rules Accuracy in Noisy Domains Using Instance Reduction Techniques

Mousa Al-Akhras[1,2,*], Zainab Darwish[2], Samer Atawneh[1] and Mohamed Habib[1,3]

[1]College of Computing and Informatics, Saudi Electronic University, Riyadh, 11673, Saudi Arabia
[2]King Abdullah II School for Information Technology, The University of Jordan, Amman, 11942, Jordan
[3]Faculty of Engineering, Port Said University, Port Said, 42523, Egypt
*Corresponding Author: Mousa Al-Akhras. Email: m.akhras@seu.edu.sa

Received: 16 November 2021; Accepted: 18 February 2022

Abstract: Association rules' learning is a machine learning method used in finding underlying associations in large datasets. Whether intentionally or unintentionally present, noise in training instances causes overfitting while building the classifier and negatively impacts classification accuracy. This paper uses instance reduction techniques for the datasets before mining the association rules and building the classifier. Instance reduction techniques were originally developed to reduce memory requirements in instance-based learning. This paper utilizes them to remove noise from the dataset before training the association rules classifier. Extensive experiments were conducted to assess the accuracy of association rules with different instance reduction techniques, namely: Decremental Reduction Optimization Procedure (DROP) 3, DROP5, ALL K-Nearest Neighbors (ALLKNN), Edited Nearest Neighbor (ENN), and Repeated Edited Nearest Neighbor (RENN) in different noise ratios. Experiments show that instance reduction techniques substantially improved the average classification accuracy on three different noise levels: 0%, 5%, and 10%. The RENN algorithm achieved the highest levels of accuracy with a significant improvement on seven out of eight used datasets from the University of California Irvine (UCI) machine learning repository. The improvements were more apparent in the 5% and the 10% noise cases. When RENN was applied, the average classification accuracy for the eight datasets in the zero-noise test enhanced from 70.47% to 76.65% compared to the original test. The average accuracy was improved from 66.08% to 77.47% for the 5%-noise case and from 59.89% to 77.59% in the 10%-noise case. Higher confidence was also reported in building the association rules when RENN was used. The above results indicate that RENN is a good solution in removing noise and avoiding overfitting during the construction of the association rules classifier, especially in noisy domains.

Keywords: Association rules classification; instance reduction techniques; classification overfitting; noise; data cleansing

1 Introduction

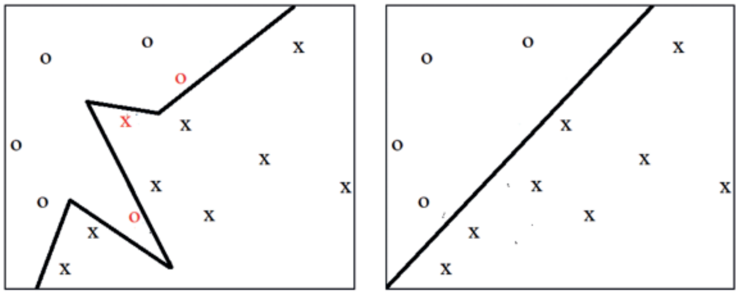
Data mining deals with the discovery of interesting unknown relationships in big data. It is the main technique used for knowledge discovery. Knowledge discovery is the extraction of hidden and possibly remarkable knowledge and underlying relations in raw data [1]. The aim of data mining is detecting the implicit and meaningful knowledge within raw data, which mainly has the following functions: Automatic prediction of trends and behavior, automatic data mining in large databases to find predictive information, the need for a large number of manual analyses of the problem can be done quickly and directly from the data itself to draw conclusions [2]. Association rule mining is a knowledge discovery technique that discovers remarkable patterns in big datasets; this is considered a crucial task during data mining [3,4]. A program learns from training samples in machine learning if it enhances its performance for a specified task with experience [5]. Learning techniques can be divided into two main groups [6]:

a. Supervised Learning: The training dataset has input vectors and matching target value(s).
b. Unsupervised Learning: The training dataset has input vectors but no output values related to them. Thus, the learning procedure determines distinct clusters or groups inside the datasets.

For classical supervised machine learning cases, the number of training instances is usually large. It could be enormous for certain types of problems. Each training instance consists of a vector of attribute values that are highly likely to uniquely correspond to a specific target class. Association rules algorithms are applied to these training examples to discover frequent patterns later used to classify unseen instances. Improving the quality of mined association rules is a complicated task involving different methods such as prevention, process control, and post evaluation, which utilize appropriate mechanisms. Users' active contribution in mining is the key to solving the problem [2]. The accuracy of machine learning algorithms is affected by the overfitting problem that occurs due to closely inseparable classes or more frequently due to noisy data. Overfitting means that the precision of the classification of current training examples is high, whereas the precision of classifying unseen test examples is much smaller [7].

Fig. 1a illustrates the case when noisy instances, which cause overfitting, are present. The constructed decision boundary between class o and class x overfits the training data, but it does not generalize well, which may cause the misclassification of unseen instances. In Fig. 1b, border (possibly noisy) instances are eliminated, simplifying finding a borderline that separates the two classes and is more likely to achieve a generalization accuracy. To reduce the effect of overfitting on the accuracy, noisy data should be eliminated. If the training dataset is free from noise, removing duplicates and invalid data; could improve the classification accuracy. However, most real-world data is not clean, and error rates are frequent in the range between 0.5% and 30%, with 1%–5% being very frequent [8].

This study aims to avoid overfitting problems in association rules learning using instance reduction techniques as noise filters. The investigated solution is a dual goal approach. It uses instance reduction techniques, which are proposed for minimizing memory requirements in instance-based learning [9], to minimize overfitting without degrading the classification accuracy of the association rules. As a result, it reduces the complexity of the classifier. Implementing noise-filtering techniques aims to decrease the number of misclassified instances caused by noise. Previous preliminary work investigated the effect of applying some limited instance reduction techniques on chosen datasets. Preceding association rules mining and building classifier with instance reduction as a pre-cleansing step, aiming to minimize the effect of noise on association rules-based classification [10]. The results showed good improvement in classification accuracy after applying the ALL K-Nearest Neighbors (ALLKNN) algorithm, particularly with higher noise ratios. This study examines through extensive empirical experiments the effect of applying five instance reduction techniques, namely Decremental Reduction Optimization Procedure (DROP) 3, DROP5, ALLKNN, Edited Nearest Neighbor (ENN), and Repeated Edited Nearest Neighbor (RENN) techniques on the accuracy of association rules classifier. These techniques are covered in Section 2.4.



(a) (b)

Figure 1: Decision boundaries with and without noisy instances. a) overfitting due to noise b) eliminating overfitting by eliminating noisy instances

The rest of this paper is organized as follows: Section 2 presents an overview of association rules learning and its related performance concepts, then it describes instance reduction techniques that will be utilized in the research, related literature is also discussed. Section 3 covers the research methodology; it includes experiments description, performance metrics, the used datasets, and the experiment settings. The conducted experiments are described, and the obtained results are discussed in Section 4 with illustrations and comparisons. Section 5 summarizes the results and presents avenues for future work.

2 Literature Review

This section gives an overview of association rules learning and its related performance concepts, then it describes utilized instance reduction techniques, related literature is also discussed.

2.1 Association Rules

Association rules classifier is a technique implemented to discover new knowledge from hidden relations among data items in a dataset. An association rule is presented as a relationship between two sides, A→ B, where A and B represent a variable or a group of variables. A represents the ancestor, while B represents the consequent. A commonly represents attributes illustrate a specific data record that governs the other part B, representing the objective class (output). Association rules learning can be applied to:

- Datasets with transactions on them: a collection of transaction records for specific data items, such as transactions on a supermarket's items.
- Datasets have no transactions: such as medical records for patients.
- Data with no timestamp is persistent through time, such as DNA sequencing.

Many association rules algorithms have been presented, for example, Apriori, Apriori Transaction Identifier (AprioriTID), Frequent Pattern (FP) Growth (FP-Growth), and others [11,12]. Apriori is the most widely used algorithm for mining association rules [11]. In Apriori, a rule-based classifier is built from the extracted association rules, which are mined from a large dataset. It calculates the occurrences of each item combination, and then combinations below a specific minimum threshold are excluded. Once the items sets are excluded, the support for each subset of items is computed using Breadth-First Search and a Hash-Tree structure. Support is described in Section 2.2.

FP-Growth is another algorithm for extracting association rules. Similarly, the support of each itemset is computed; individual items within an itemset are ordered in descending way according to the support values. Any itemsets under the minimum specified support are eliminated. The remaining itemsets are then used to build a Frequent Pattern Tree (FP-Tree) [13]. The FP-Tree illustrates a prefix tree presenting the transactions [13], where an individual path represents a group of transactions sharing similar prefix items. Then, each item is represented by a node. All nodes referring to the same item are linked in a list. Iteratively, a new path is constructed for each unique transaction, and if it shares a common prefix itemset, nodes are added as needed. The processing order of examples does not matter in association rules learning, nor each variable's order within the example.

2.2 Support and Confidence

The biggest problem in mining association rules from big datasets is the vast number of rules. The number of rules increases exponentially as the dataset size increases. Thus, most algorithms limit the discovered rules to specific quality measures. Those measures are usually support and confidence for each discovered rule. The following equation calculates the support for each itemset:

$$\mathrm{Support}\ (\mathrm{I}_0) = \frac{|\{\tau \in \mathrm{R} | \mathrm{I}_0 \subseteq \tau\}|}{\mathrm{R}} \tag{1}$$

I_0 is an instance from the items' set I, R is a transaction set on I, and τ (tau) is a transaction on I_0 [1].

The confidence for generated association rule is computed using the following equation:

$$\mathrm{Confidence}\ (\mathrm{A} \rightarrow \mathrm{C}) = \frac{\mathrm{support}\ \ (\mathrm{A} \cup \mathrm{C})}{\mathrm{support}\ \ (\mathrm{A})} \tag{2}$$

$\mathrm{A} \rightarrow \mathrm{C}$ is the examined rule for calculating its accuracy [1]. Typical algorithms for association rules mining find all itemsets meeting the minimum specified support. Association rules are then generated from them [14], resulting in an excessively large number of very specific association rules or rule sets with low predictive power.

2.3 Overfitting and Association Rules Pruning

Overfitting the training results is considered the most serious problem for discovering association rules. Overfitting is when very accurate classification accuracy is obtained for the training examples, while it is much worse for unseen examples. The pruning method is used to prevent the overfitting issue. Association rules pruning is divided into two main approaches [15]:

a. Pre-Pruning: Sample from the generated rules are terminated. It is commonly applied when the algorithm implemented for generating association rules uses a decision tree in intermediate form.
b. Post-Pruning: Some of the rules are eliminated after generating all rules. Two main methods for post-pruning can be applied depending on their error rates. One of the approaches divides the dataset into three parts: training, validation, and testing datasets. The training dataset is used to generate association rules, and then pruning is applied based on the association rules performance on the validation set by eliminating rules under a minimum stated threshold. The

other method is pessimistic error pruning; where the training part is used for both training and validation, and the pessimistic error rate is calculated per rule. Rules that have a pessimistic error greater than the corresponding sub-rules will be removed.

2.4 Instance Reduction Algorithms

For big datasets, excessive storage and time complexity are needed to process the large number of instances in a dataset. Moreover, some of these instances may cause the classifier to be very specific and generate an overfitted classifier that produces unreliable classification results on unseen data. Instance reduction algorithms are mainly used to minimize the vast memory requirements for huge datasets [9]. Most instance reduction algorithms have been designed and combined with the nearest neighbor classifier. As a lazy learning algorithm, it classifies unseen instances to the class to which the majority of their k nearest neighbors in the training set belong based on a certain distance or similarity measure. Thus, reducing the training set allows us to reduce computations complexity and alleviate the high storage requirement of this classifier [16].

Instance reduction algorithms are categorized into two groups:

- Incremental reduction algorithms: the process starts with an empty set, and in each iteration, essential items from the original dataset are added incrementally, e.g., Encoding Length (ELGrow) [17].
- Decremental reduction algorithms: begin with the complete dataset, then progressively eliminate irrelevant items, e.g., Decremental Reduction Optimization Procedures (DROPs) [9].

In this research, instance reduction algorithms are used as filtering techniques to keep the nearest instances based on a specified distance function and discard other distant instances that could cause overfitting.

The following is a brief description of the instance reduction algorithms tested in this work:

- ALLKNN algorithm [18]: ALLKNN extended ENN. This algorithm works as follows: for $i = 1$ to k, any instance that is not classified correctly by its neighbors is flagged as a bad instance. After completing the loop all k times, instances flagged as bad are removed from S.
- ENN algorithm [19]: A Decremental reduction algorithm; it starts with the whole dataset S and then removes each instance that does not conform to any of its K nearest neighbors (with $K = 3$, typically) according to the applied distance function. This process smooths decision boundaries by removing noisy cases and close-border instances. ENN algorithm has been used in various condensation methods as a pre-processing filter to exclude the noisy instances [20].
- RENN Algorithm [19]: RENN applies ENN repeatedly until the majority of the remaining instances' neighbors have the same class, leaving clear classes with smooth decision boundaries.
- DROP3 algorithm [19]: This algorithm starts with a noise filtering pass similar to ENN. Any instance misclassified by its k nearest neighbors is removed from S. Then instances are sorted according to the distance between them and their nearest enemy (nearest neighbor in a different class). Instances with higher distances are removed first.
- DROP5 algorithm [9]: The removal criterion for an instance is: "Remove instance p if at least as many of its associates in T would be classified correctly without p." The removal process starts with instances that are nearest to their nearest enemy.

The mentioned instance reduction algorithms keep the patterns that have a higher contribution for pattern classification and remove the large number of inner patterns and all outlier patterns [20].

2.5 Related Work

Many of the algorithms for discovering association rules work on structured data. However, nowadays, with the widespread sensor-rich environments and the massive volume of flexible and extensible data (e.g., JavaScript Object Notation (JSON)), these algorithms are not designed for unstructured or semi-structured datasets. Jury on [21] presented a data analytics method for data models in a tree-based format to support the discovery of positive and negative association rules. His method evaluates fractions of Extensible Markup Language (XML) based data to determine if they could present informative negative rules or not, even if their support and confidence values were not enough to the given conditions. The work in this paper can also be applied to unstructured data. However, in this case, datasets have to be transformed using an information retrieval scheme to apply filtering techniques before generating association rules.

Dong et al. [22] referenced the main problem in mining Positive and Negative Association Rules (PNAR), which represent a huge count of discovered rules. Indeed, the number of negative rules is more than three times of the positive rules discovered. This makes it very difficult for users to build a decision from these rules. A novel methodology is presented to prune redundant negative association rules by applying logical reasoning. They merged the correlation coefficient with multiple minimum confidences to assure that the discovered PNARs are related, this proposed model supports the control of the count of all rule's types, and it prunes weak correlation rules.

An analysis of the crucial risk factors and treatment mechanisms, following the integrated Bayesian network, then applying association rule algorithms has been confronted by Du et al. [23]. They applied their study to analyze the methods to minimize the risk of postpartum hemorrhage after cesarean section. The probability of risk factors influencing the main factors that cause postpartum hemorrhage after the cesarean section has been computed by a Bayesian network model depending on regression analysis. The discovered rules confronted solutions to different causes of postpartum hemorrhage and offered suggestions for medical institutions to amend the efficiency of variance treatment.

Yang et al. [24] focused on the process of finding and pruning time series association rules from sensor data. Regular association rules algorithms produce a huge number of rules; this makes it very hard to interpret or use; thus, they presented a two-step pruning approach to decrease the redundancy in a huge result set of time series rules. The first step targets determining rules that can correspond to other rules or carry more information than other rules. The second step summarizes the leftover rules using the bipartite graph association rules analysis method, which is appropriate for demonstrating the distribution of the rules and summarizing the interesting clusters of rules.

Najafabadi et al. [25] applied a modification step to the pre-processing phase prior to the association rules mining to discover similar patterns. Also, they used the clustering method in the proposed algorithm to minimize the data size and dimensionality. The results indicated that this algorithm improved the performance over traditional collaborative filtering techniques measured by precision and recall metrics.

Most business enterprises aim to anticipate their client potential to help business decisions and determine possible business intelligence operations to acquire dependable forecasting results. Yang et al. [26] modified the Naïve Bayes classifier in association rules mining to determine the relations for marketing data in the banking system. In the first step, a classifier was implemented to classify dataset items. Then, the Apriori algorithm was employed to merge interrelated attributes to minimize the dataset's features.

Nguyen et al. [27] used collaborative filtering for quantitative association rules to build a recommendation system. A solution has been presented to discover association rules on binary data and to support quantitative data. The algorithm was applied on Microsoft Web (MSWEB) and MovieLens datasets, binary and quantitative datasets. The results indicate that the proposed collaborative filtering model to discover implication rules is more efficient than the traditional model measured by accuracy, performance, and the required time for building a recommender system.

Zhang et al. [28] proposed a novel multi-objective evolutionary algorithm to discover positive and negative association rules. This algorithm aims to enhance the process of multi-objective optimization by applying a reference point that depends on a non-dominated sorting method. The genetic crossover technique is applied to extend the process for crossover, and the rules mutation has been improved. In addition, the algorithm can deal with all attribute types in the datasets. The results show that this improved algorithm performs much more effectively in quality measurements.

3 Research Methodology

3.1 Experiments Description

The proposed approach includes performing a set of experiments to demonstrate results compared to other approaches. Experiments are applied as follows:

a. Read a dataset from the UCI machine learning repository.
b. Produce new datasets by injecting noise with 0%, 5%, and 10% to the original datasets. Noise is injected by randomly changing the class attribute of the itemsets.
c. For each of the above noise ratios, generate association rules in each of the following cases:
    - Neither noise filtering nor pruning techniques are applied.
    - Built-in pruning is used without implementing noise filtering techniques.
    - Noise filtering techniques are used without implementing built-in pruning.
    - Noise filtering techniques and built-in pruning are both applied.

Experiment (a) aims to study the impact of noise on classification accuracy, while experiment (b) is intended to show whether applying only pruning will improve the classification accuracy. Experiment (c) aims to discover the efficiency of only implementing noisy filtering techniques. Finally, experiment (d) tests the efficiency of combining noise filtering and pruning algorithms in succession. The previous experiments aim to illustrate the impact of noise on classification accuracy and examine the effect of built-in pruning and noise filtering techniques on the classification accuracy in noisy instances. To conduct the above experiments and apply association rules classification, the Waikato Environment for Knowledge Analysis (WEKA) data mining tool [29] is employed to build an association rules classifier from datasets. WEKA implements several algorithms to build association rules. The Apriori algorithm is used in this research since it is adequate with all the chosen datasets. The research methodology is illustrated in Fig. 2.

The pruning function implemented in WEKA applies a pessimistic error rate-based pruning in C4.5. After deleting one of their conditions, it keeps the rules that have an estimated error rate lower than those of the same rules. The pessimistic error rate is a top-down greedy pruning approach that eliminates conditions successively from the Apriori-Tree if this reduces the estimated error. The problem is that some rules could be discarded totally, which means that some cases will not be covered, and this will affect the prediction of unseen instances. This will be confirmed empirically in this research.

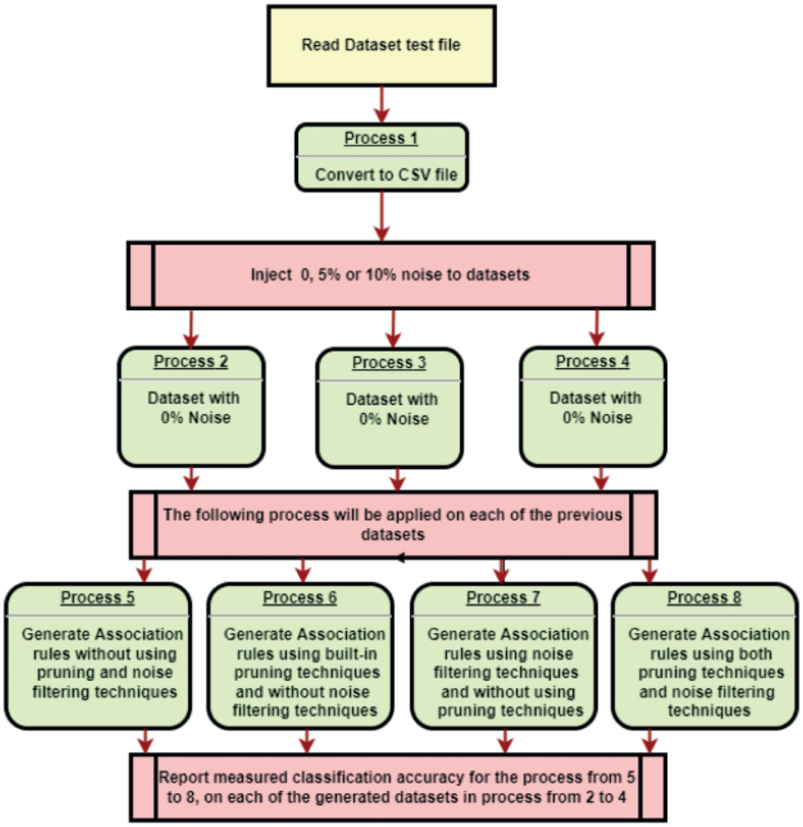

Read Dataset test file
Process 1
Convert to CSV file
Inject 0, 5% or 10% noise to datasets
Process 2
Dataset with 0% Noise
Process 3
Dataset with 0% Noise
Process 4
Dataset with 0% Noise
The following process will be applied on each of the previous datasets
Process 5
Generate Association rules without using pruning and noise filtering techniques
Process 6
Generate Association rules using built-in pruning techniques and without noise filtering techniques
Process 7
Generate Association rules using noise filtering techniques and without using pruning techniques
Process 8
Generate Association rules using both pruning techniques and noise filtering techniques
Report measured classification accuracy for the process from 5 to 8, on each of the generated datasets in process from 2 to 4

Figure 2: The research methodology followed in the paper

In WEKA, JCBAPrunning is a class implementing the pruning step of the classification Based on Association (CBA) algorithm using a Cache-conscious Rectangle Tree (CR-Tree). The CR-tree is a prefix tree structure to explore the sharing among rules, which achieves substantial compactness. CR-tree itself is also an index structure for rules and serves rule retrieval efficiently. Valid options for JCBAPruning are:

- C the confidence value: The confidence value for the optional pessimistic-error-rate-based pruning step (default: 0.25).
- N: If set, no pessimistic-error-rate-based pruning is performed.

3.2 Performance Metrics

The accuracy of association rules classification in different experiments is compared using several performance metrics: precision, recall, and percentage of correctly classified instances. Here is a brief explanation for each measure:

$$\text{Precision (true positives)} = \frac{\text{Number of the instances which truly have class x}}{\text{Total number of instances classified as class x}} \qquad (3)$$

$$\text{Recall} \quad = \frac{\text{True Positive}}{\text{True positive} + \text{false negative}} \tag{4}$$

where False negatives: percentage of incorrectly classified examples as not belonging to class x while they truly belong to class x.

$$\text{Correctly classified instances percent} = \frac{\text{Number of correctly classified instances}}{Number\ of\ all\ instances} \tag{5}$$

3.3 Datasets

To evaluate the proposed approach, eight benchmark datasets were selected from the University of California Irvine (UCI) machine learning repository [30] to conduct the experiments. These datasets vary in size, the number of attributes, and data type. These datasets are convenient to supervised learning as they include only one class, as the Apriori algorithm accepts only single target attribute datasets. Tab. 1 shows the details for the used datasets.

Table 1: UCI eight benchmark datasets

Name	Instances	Input attributes	Output attributes	Classes
Iris	150	4	1	3
Glass	214	9	1	7
Voting	435	16	1	2
Breast cancer-diagnostic	699	10	1	2
Liver	345	6	1	2
Pima-indians-Diabetes	768	8	1	2
Banknote authentication	1372	4	1	2
Heart-Cleveland	303	13	1	2

3.4 Experiments Settings

WEKA, a data mining tool from the University of Waikato, New Zealand, was used to conduct the pre-described experiments and build associate rules classifiers from the above datasets. The version that was used to apply association rule-based classification is WEKA 3.8.4. This version of WEKA contains a package manager that enables the user to install learning schemes (in our case, we will install Java association rule-based classifier) since it is not embedded implicitly. This classifier implements Apriori and predictive Apriori algorithms. In the conducted experiments, Apriori is applied since it fits the chosen datasets. An optional pruning parameter is included in the Apriori algorithm to enable/disable built-in pruning. The algorithm works as a decision list classifier and includes mandatory and optional pruning phases. The optional pruning parameters are deactivated when no pruning is implemented. A pessimistic error rate-based pruning is applied in the pruning function such as in C4.5. Thus, the rules are pruned with an approximate error, which is greater than the corresponding rule after deletion of one of its conditions. Elective pessimistic error rate-based pruning has a confidence level that ranges from zero to 1.0. The conducted experiments, set confidence value to its default value of 0.25. To prepare the datasets for the experiments, attribute values must be discretized as required by the classifier.

In WEKA, Discretize is an instance filter that discretizes a range of numeric attributes in a dataset into nominal attributes. Discretization is a simple binning. Skips the class attribute if set. Continuous ranges are divided into sub-ranges by the user-specified parameter, such as equal width (specifying range of values) and equal frequency (number of instances in each interval).

Valid options for the discretizing filter are:

- unset-class-temporarily:	Unsets the class index temporarily before the filter is applied to the data. (default: no)
- B<num>:	Specifies the (maximum) number of bins to divide numeric attributes into. (default = 10)
-M<num>:	Specifies the desired weight of instances per bin for equal-frequency binning. If this is set to a positive number, then the-B option will be ignored. (default = −1)
- F:	Use equal-frequency instead of equal-width discretization.
- O:	Optimize the number of bins using leave-one-out estimate of estimated entropy (for equal-width discretization). If this is set, then the-B option will be ignored.
- R <col1, col2-col4,... >:	Specifies list of columns to Discretize. First and last are valid indexes. (default: first-last)
- V:	Invert matching sense of column indexes.
- D:	Output binary attributes for discretized attributes.
- Y:	Use bin numbers rather than ranges for discretized attributes.
- precision<integer>:	Precision for bin boundary labels. (default = 6 decimal places).
- spread-attribute-weight:	When generating binary attributes, spread weight of old attribute across new attributes. Do not give each new attribute the old weight.

4 Experiments and Results

The proposed approach aims to reduce the effect of overfitting in noisy domains by applying instance reduction techniques, which will act as noise filters before generating association rules and conducting association rules classification. Classification accuracy will be compared with the case when only built-in pessimistic error pruning is applied. Further comparisons will explore the effect of combining both instance reduction (noise filtering) techniques and built-in pruning to improve classification accuracy.

In addition to the noise-free base case, the noise was also introduced to the datasets in two ratios: 5% and 10% by changing classes of the itemsets. In the classification task, 10-fold cross-validation is used to test the learning algorithm. Also, filtering techniques, when applied, use 10-folds. The results reported in this section show the average for the 10-folds. The used performance metrics are the percentage of correctly classified instances, precision, and recall.

4.1 Investigating the Effect of Noise

In this experiment, both noise filtering and built-in pruning were not applied. This experiment aims to study the effect of noise and construct a baseline that can be compared with the results in subsequent experiments. Tab. 2 shows the performance of associate rules using different datasets under 0%, 5%, and 10% noise ratios. Fig. 3 compares the performance of the association rules classifier using different noise ratios when both noise filtering and built-in pruning were not applied. It can intuitively be noticed that all three performance metrics degrade with the increased noise level. The above set of experiments will be referred to as baseline when compared with the results obtained in subsequent sections.

Table 2: Association rules classifier results with no pruning, no filtering using different noise ratios

	0% noise (A1)			5% noise (B1)			10% noise (C1)		
Dataset name	Correctly classified %	Precision %	Recall %	Correctly classified %	Precision %	Recall %	Correctly classified %	Precision %	Recall %
Iris	94.00	94.00	94.00	80.00	84.00	80.00	63.33	68.00	63.00
Glass	47.20	48.00	47.00	38.32	59.00	38.00	37.38	41.00	37.00
Voting	92.41	92.00	92.00	92.41	92.00	92.00	80.23	80.00	80.00
Breast cancer-diagnostic	92.99	93.00	93.00	89.56	90.00	90.00	73.82	75.00	74.00
Liver	57.97	34.00	58.00	57.10	33.00	57.00	57.97	34.00	58.00
Pima-indians-diabetes	67.19	72.00	67.00	63.54	40.00	64.00	62.11	39.00	62.00
Banknote authentication	55.54	31.00	56.00	55.03	30.00	55.00	54.45	30.00	54.00
Heart-cleveland	56.44	50.00	56.00	52.68	28.00	53.00	49.84	25.00	50.00
Average	70.47	64.25	70.38	66.08	57.00	66.13	59.89	49.00	59.75

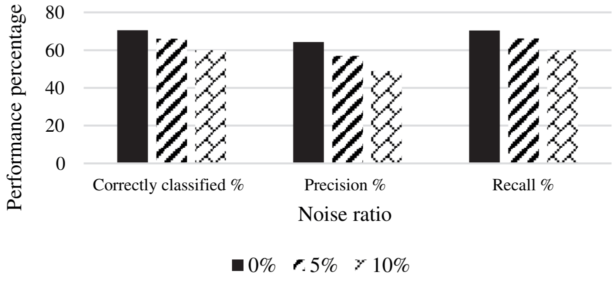


Figure 3: Comparison of performance of association rules under different noise ratios with no noise filtering or built-in pruning

4.2 The Effect of Applying Filtering and Pruning

In this section, a series of experiments are conducted to study the effect of implementing noise filtering and built-in pruning in different combinations on the datasets with 0%, 5%, and 10% noise ratios inserted.

4.2.1 The Effect of Applying Filtering and Pruning with 0% Noise

Experiments set (A1-A4) are applied on the original datasets without noise injection.

- A1. Baseline results were obtained in Section 4.1 without applying noise filtering or pruning.
- A2. Results were obtained by applying built-in pruning methods with no noise filtering method.
- A3. Results were obtained by applying noise filtering methods without built-in pruning.
- A4. Results were obtained by applying both noise filtering and built-in pruning.

Tab. 3 compares the accuracy for cases: A1, A2, and A3 for the chosen datasets using different instance reduction techniques. Tab. 4 compares the accuracy for cases: A1, A2, and A4. Results highlighted in Bold indicate classification accuracy enhancements compared with the baseline case A1. In all Subsequent tables, CC implies Correctly Classified instances, P implies precision, and R implies recall.

It can be clearly noticed that applying only pruning to the datasets in case A2 was insufficient to increase the classification accuracy. It unexpectedly reduced the overall precision compared with the base case A1. This was a probable result due to the behavior of pessimistic error rate-based pruning methodology as a greedy pruning approach that consecutively eliminates conditions from the Apriori-Tree if this minimizes the estimated error. Some potential important rules could be discarded; consequently, the prediction accuracy for unseen instances could be affected. A3 cases show the impact of using noise filtering techniques only without applying pruning. ENN showed good improvement, and better results were achieved when RENN was applied as the classification accuracy improved in five out of eight datasets.

The last set of tests, A4, investigates the impact of implementing both noise filtering and built-in pruning algorithms. When results are compared with A3 cases, classification accuracy is reduced heavily, even when compared to the benchmark case A1. Therefore, implementing both filtering and pruning algorithms concurrently would not certainly result in better classification accuracy. It must be noted that the behavior of this combination of algorithms still needs to be studied in noisy domains, as explored in the subsequent experiments. As the best results in the current set of experiments were achieved when only noise filtering was applied before building association rules (A3), the impact of applying noise filtering techniques on the quality of association rules will be examined next.

Tabs. 5 and 6 compare the resulting rules without applying filtering or built-in pruning with the resulting rules after applying RENN (the best noise filter) for the Breast-Cancer dataset. Tab. 5 shows the first ten rules and their confidence value produced in the zero-noise case without applying filtering or pruning. The confidence range for the rules produced after applying RENN is shown in Tab. 6.

It can be noticed that applying RENN affected the confidence of the generated association rules. When neither noise filtering nor pruning were applied, the confidence ranged from 1 to 0.94, while it improved with RENN from 1 to 0.99.

More experiments are needed in noisy domains to check the effectiveness of instance reduction techniques as noise filters.

Table 3: Comparing classification accuracy results for cases A1, A2, and A3

Dataset	A1			A2			A3														
							DROP3			DROP5			ALLKNN			ENN			RENN		
	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R
Iris	94.00	94.00	94.00	94.00	95.00	94.00	76.58	86.00	77.00	69.80	74.00	70.00	80.14	83.00	80.00	81.69	85.00	82.00	81.21	84.00	81.00
Glass	47.20	48.00	47.00	35.51	13.00	36.00	36.87	34.00	45.00	40.10	29.00	40.00	50.80	49.00	51.00	47.21	43.00	47.00	62.91	61.00	63.00
Voting	92.41	92.00	92.00	92.41	92.00	92.00	72.50	75.00	73.00	64.70	62.00	65.00	63.85	71.00	64.00	63.30	71.00	63.00	63.30	71.00	63.00
Breast cancer	92.99	93.00	93.00	89.70	90.00	90.00	71.86	75.00	72.00	57.00	66.00	62.00	96.42	96.00	96.00	95.64	96.00	96.00	95.65	96.00	96.00
Liver	57.97	34.00	58.00	57.97	34.00	58.00	54.22	56.00	54.00	54.00	45.00	54.00	81.23	81.00	82.00	80.27	80.00	80.00	82.63	82.00	83.00
Diabetes	67.19	72.00	67.00	67.19	72.00	67.00	63.93	72.00	64.00	59.70	64.00	60.00	58.38	78.00	65.00	65.36	77.00	65.00	65.43	78.00	65.00
Banknote	55.54	31.00	56.00	55.54	31.00	56.00	59.40	65.00	64.00	76.80	77.00	77.00	92.87	93.00	93.00	93.68	94.00	94.00	94.10	94.00	94.00
Heart	56.44	50.00	56.00	56.77	48.00	57.00	64.21	64.00	64.00	57.50	52.00	58.00	65.33	49.00	65.00	65.82	48.00	66.00	67.99	52.00	68.00
Average	70.47	64.25	70.38	68.64	59.38	68.75	62.45	65.90	64.10	60.00	58.60	60.80	73.63	75.00	74.50	74.12	74.30	74.10	76.65	77.25	76.60

Table 4: Comparing classification accuracy results for cases A1, A2, and A4

Dataset	A1			A2			A4														
							DROP3			DROP5			ALLKNN			ENN			RENN		
	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R
Iris	94.00	94.00	94.00	94.00	**95.00**	94.00	54.49	55.00	54.00	63.80	64.00	64.00	59.42	62.00	59.00	62.64	69.00	63.00	62.69	69.00	63.00
Glass	47.20	48.00	47.00	35.51	13.00	36.00	38.47	18.00	39.00	39.14	20.00	39.00	46.41	38.00	47.00	44.08	37.00	44.00	**56.28**	**56.00**	**56.00**
Voting	92.41	92.00	92.00	92.41	92.00	92.00	68.88	67.00	73.00	65.64	56.00	66.00	63.85	71.00	64.00	43.03	38.00	44.00	56.28	56.00	56.00
Breast cancer	92.99	93.00	93.00	89.70	90.00	90.00	61.76	59.00	62.00	67.09	67.00	67.00	**96.37**	**96.00**	**96.00**	**95.59**	**96.00**	**96.00**	**95.60**	**96.00**	**96.00**
Liver	57.97	34.00	58.00	57.97	34.00	58.00	53.32	**49.00**	53.00	54.05	**42.00**	54.00	**81.90**	**81.00**	**82.00**	**80.40**	**80.00**	**81.00**	**80.88**	**80.00**	**81.00**
Diabetes	67.19	72.00	67.00	67.19	72.00	67.00	57.55	33.00	58.00	49.68	39.00	50.00	55.96	57.00	56.00	55.85	58.00	56.00	55.71	58.00	56.00
Banknote	55.54	31.00	56.00	55.54	31.00	56.00	51.70	**48.00**	52.00	**56.12**	**56.00**	56.00	**88.45**	**89.00**	**88.00**	**84.14**	**85.00**	**84.00**	**85.25**	**86.00**	**85.00**
Heart	56.44	50.00	56.00	**56.77**	48.00	**57.00**	**60.62**	**51.00**	**61.00**	**57.78**	**78.00**	**58.00**	**65.21**	**49-.00**	**65.00**	**65.82**	**48.00**	**66.00**	**67.50**	**51.00**	**68.00**
Average	**70.47**	**64.25**	**70.38**	**68.64**	**59.38**	**68.75**	**55.85**	**47.50**	**56.50**	**56.66**	**52.80**	**56.80**	**69.70**	**67.90**	**69.60**	**66.44**	**63.90**	**66.80**	**70.02**	**69.00**	**70.10**

Table 5: Confidence value for the first ten rules when neither noise filtering nor built-in pruning was applied at 0% noise ratio

Rule	Confidence
If uniformity of cell size = 1 AND normal nucleoli = 1 ==> Class = 2	1
If uniformity of cell shape = 1 ==> Class = 2	0.99
If uniformity of cell size = 1 AND mitoses = 1 ==> Class = 2	0.99
If bare nuclei = 1 AND normal nucleoli = 1 ==> Class = 2	0.99
If uniformity of cell size = 1 ==> Class = 2	0.99
If bare nuclei = 1 AND mitoses = 1 ==> Class = 2	0.98
If bare nuclei = 1 ==> Class = 2	0.96
If single epithelial cell size = 2 AND mitoses = 1 ==> Class = 2	0.95
If marginal adhesion = 1 AND mitoses = 1 ==> Class = 2	0.95
If single epithelial cell size = 2 ==> Class = 2	0.94

Table 6: Confidence value for the first ten rules when RENN noise filtering was applied without built-in pruning at 0% noise ratio

Rule	Confidence
If uniformity of cell size= (-inf-0.9] AND normal nucleoli= (-inf-0.9] ==> Class = 0	1
If uniformity of cell size= (-inf-0.9] AND bare nuclei= (-inf-0] ==> Class = 0	1
If uniformity of cell size= (-inf-0.9] AND normal nucleoli= (-inf-0.9] AND mitoses= (-inf-0.9] ==> Class = 0	1
If uniformity of cell size= (-inf-0.9] AND bare nuclei= (-inf-0] AND mitoses= (-inf-0.9] ==> Class = 0	1
If uniformity of cell shape= (-inf-0.9] ==> Class = 0	1
If uniformity of cell shape= (-inf-0.9] AND mitoses= (-inf-0.9] ==> Class = 0	1
If uniformity of cell Size= (-inf-0.9] AND mitoses= (-inf-0.9] ==> Class = 0	0.99
If bare nuclei= (-inf-0] AND normal nucleoli= (-inf-0.9] AND mitoses= (-inf-0.9] ==> Class = 0	0.99
If marginal adhesion= (-inf-0.9] AND bare nuclei= (-inf-0] AND mitoses= (-inf-0.9] ==> Class = 0	0.99
If marginal adhesion= (-inf-0.9] AND normal nucleoli= (-inf-0.9] AND mitoses= (-inf-0.9] ==> Class = 0	0.99

4.2.2 The Effect of Applying Filtering and Pruning with 5% Noise

Experiments set (B1-B4) 5% noise ratio was injected into the datasets by changing the class attribute. Four different tests are conducted as follows:

- B1. Baseline results were obtained in Section 4.1 without applying noise filtering or pruning.
- B2. Results were obtained by applying built-in pruning methods with no noise filtering method.
- B3. Results were obtained by applying noise filtering methods without built-in pruning.

- B4. Results were obtained by applying both noise filtering and built-in pruning.

Tab. 7 compares the accuracy for cases: B1, B2, and B3 for the chosen datasets using different instance reduction techniques. Tab. 8 compares the accuracy for cases: B1, B2, and B4. Results highlighted in Bold indicate accuracy enhancements compared with the baseline case B1.

It can be noticed that in B2 when only built-in pruning was applied, classification accuracy did not increase for the 5% noise injected datasets. It even reduced overall accuracy due to the pessimistic error rate-based pruning approach. From B3 experiments it appears that applying ALLKNN showed a great improvement, RENN achieved a significant improvement. It can be noticed from Tab. 7 that the difference between the results achieved by RENN (77.47%) in comparison to the baseline (66.08%) is greater than the same difference (76.65% to 70.47%) when the noise was 0% as shown in Tab. 3. This indicates that the improvement due to the noise filter would be more apparent with the increase in the noise ratio. This observation will later be thoroughly analyzed. The results for the last set B4 were worse than those in B3, applying ALLKNN and RENN with pruning introduced good results, but still the best results achieved by applying ALLKNN and RENN without pruning.

Tab. 9 shows the first ten rules produced for the Breast-Cancer dataset and their confidence without applying filtering or built-in pruning with a 5% noise ratio. Tab. 9 shows the first ten rules with their confidence produced after applying RENN. It can be noticed from Tab. 9 that the ten rules have a 0.94 confidence value, which is lower than the zero-noise case, and this is an expected result due to noise. Again, the confidence value for the constructed association rules improved when the RENN noise filter was applied; it ranges from 1 to 0.99, as shown in Tab. 10.

4.2.3 The Effect of Applying Filtering and Pruning with 10% Noise

Experiments set (C1-C4) 10% noise ratio was injected into the datasets by changing the class attribute. Four different tests are conducted as follows:

- C1. Baseline results were obtained in Section 4.1 without applying noise filtering or pruning.
- C2. Results were obtained by applying built-in pruning methods with no noise filtering method.
- C3. Results were obtained by applying noise filtering methods without built-in pruning.
- C4. Results were obtained by applying both noise filtering and built-in pruning.

Tab. 11 compares the accuracy for cases: C1, C2, and C3 for the chosen datasets using different instance reduction techniques. Tab. 12 compares the accuracy for cases: C1, C2, and C4. Results highlighted in Bold indicate accuracy enhancements compared with the baseline case C1.

In this experiment, it is hard to notice an improvement in C2 compared to C1. Accuracy enhanced significantly in C3 when ALLKNN and RENN were applied, which indicates an excellent performance for these techniques with higher noise ratios. Also, improvements are made in C4 when applying ALLKNN and RENN simultaneously with pruning, compared to C1.

The efficiency of applying only noise filtering, mainly ALLKN and RENN, is becoming more evident with the increased injected noise ratio. RENN achieved 77.59% classification accuracy compared to (59.89%) achieved by the baseline experiment with the same noise ratio. When the noise ratio was 0%, RENN achieved 76.65%, while the achieved accuracy with the baseline at the same noise ratio was 70.47%, as reported in Tab. 3. The difference now is more apparent, indicating the increasing importance of using noise filtering techniques in the presence of more noise.

Table 7: Comparing classification accuracy results for cases B1, B2, and B3

Dataset	B1			B2			B3														
							DROP3			DROP5			ALLKNN			ENN			RENN		
	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R
Iris	80.00	84.00	80.00	70.00	78.00	70.00	**85.71**	**92.00**	**87.00**	76.14	83.00	76.00	79.73	82.00	80.00	**82.79**	**85.00**	**83.00**	**83.42**	**86.00**	**84.00**
Glass	38.32	59.00	38.00	33.01	35.00	33.00	36.54	28.00	37.00	37.90	21.00	38.00	**61.79**	**60.00**	**62.00**	**57.70**	56.00	**58.00**	**71.97**	**71.00**	**72.00**
Voting	92.41	92.00	92.00	92.41	92.00	92.00	79.32	73.00	79.00	61.74	62.00	62.00	62.77	70.00	63.00	63.23	71.00	63.00	62.71	70.00	63.00
Breast cancer	89.56	90.00	90.00	89.56	90.00	90.00	67.38	73.00	67.00	67.54	71.00	68.00	**96.40**	**97.00**	**97.00**	**94.99**	**95.00**	**95.00**	**94.71**	**95.00**	**95.00**
Liver	57.10	33.00	57.00	57.10	33.00	57.00	55.27	**49.00**	55.00	54.64	**46.00**	55.00	**84.17**	**84.00**	**84.00**	**80.54**	**80.00**	**81.00**	**82.71**	**82.00**	**83.00**
Diabetes	63.54	40.00	64.00	63.54	40.00	64.00	59.71	**65.00**	60.00	55.06	55.00	55.00	**66.59**	**79.00**	**67.00**	**66.23**	**78.00**	**66.00**	**66.23**	**78.00**	**66.00**
Banknote	55.03	30.00	55.00	55.03	30.00	55.00	**68.32**	**69.00**	**68.00**	**69.34**	**69.00**	**70.00**	**91.18**	**92.00**	**91.00**	**93.11**	**93.00**	**93.00**	**92.44**	**93.00**	**93.00**
Heart	52.68	28.00	53.00	52.68	28.00	53.00	**61.88**	**63.00**	**62.00**	**58.74**	**57.00**	**59.00**	**66.76**	**58.00**	**67.00**	**63.16**	**43.00**	**63.00**	**65.60**	**54.00**	**66.00**
Average	**66.08**	**57.00**	**66.13**	**64.17**	**53.25**	**64.25**	**64.27**	**64.00**	**64.40**	**60.14**	**58.00**	**60.40**	**76.17**	**77.80**	**76.40**	**75.22**	**75.10**	**75.30**	**77.47**	**78.63**	**77.80**

Table 8: Comparing classification accuracy results for cases B1, B2, and B4

Dataset	B1			B2			B4														
							DROP3			DROP5			ALLKNN			ENN			RENN		
	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R
Iris	80.00	84.00	80.00	70.00	78.00	70.00	54.75	63.00	55.00	62.63	61.00	63.00	57.82	53.00	58.00	61.11	65.00	61.00	61.01	67.00	61.00
Glass	38.32	59.00	38.00	33.01	35.00	33.00	34.88	17.00	35.00	**39.02**	15.00	**39.00**	**53.49**	50.00	**54.00**	**56.27**	**56.00**	**56.00**	**64.19**	**63.00**	**64.00**
Voting	92.41	92.00	92.00	92.41	92.00	92.00	74.70	68.00	75.00	63.34	53.00	63.00	62.71	70.00	63.00	63.26	71.00	63.00	62.71	70.00	63.00
Breast cancer	89.56	90.00	90.00	89.56	90.00	90.00	67.04	78.00	73.00	64.21	57.00	64.00	**96.24**	**96.00**	**96.00**	**94.83**	**95.00**	**95.00**	**94.64**	**95.00**	**95.00**
Liver	57.10	33.00	57.00	57.10	33.00	57.00	54.56	**43.00**	55.00	54.47	**35.00**	55.00	**83.68**	**83.00**	**84.00**	**80.56**	**80.00**	**80.00**	**81.53**	**81.00**	**82.00**
Diabetes	63.54	40.00	64.00	63.54	40.00	64.00	57.28	35.00	57.00	50.80	35.00	51.00	**56.13**	**56.00**	**56.00**	**56.34**	**59.00**	57.00	**56.34**	**59.00**	**57.00**
Banknote	55.03	30.00	55.00	55.03	30.00	55.00	52.00	**51.00**	52.00	53.59	**54.00**	54.00	**87.40**	**89.00**	**87.00**	**85.14**	**86.00**	**85.00**	**85.71**	**86.00**	**86.00**
Heart	52.68	28.00	53.00	52.68	28.00	53.00	**62.41**	**51.00**	**62.00**	**57.81**	**48.00**	**58.00**	**66.42**	**53.00**	**66.00**	**63.16**	**43.00**	**63.00**	**65.77**	**52.00**	**66.00**
Average	**66.08**	**57.00**	**66.13**	**64.17**	**53.25**	**64.25**	**57.20**	**50.80**	**58.00**	**55.73**	**44.80**	**55.90**	**70.49**	**68.80**	**70.50**	**70.08**	**69.40**	**70.00**	**71.49**	**71.60**	**71.80**

Table 9: Confidence value for the first ten rules when neither noise filtering nor built-in pruning was applied at a 5% noise ratio

Rule	Confidence
If uniformity of cell size= (-inf-1.9] AND bare nuclei= (-inf-1.9] ==> Class = 2	0.94
If uniformity of cell size= (-inf-1.9] AND bare nuclei= (-inf-1.9] ==> Class = 2	0.94
If bare nuclei= (-inf-1.9] AND normal nucleoli= (-inf-1.9] AND mitoses= (-inf-1.9] ==> Class = 2	0.94
If bare nuclei= (-inf-1.9] AND normal nucleoli= (-inf-1.9] AND mitoses= (-inf-1.9] ==> Class = 2	0.94
If uniformity of cell size= (-inf-1.9] AND normal nucleoli= (-inf-1.9] ==> Class = 2	0.94
If bare nuclei= (-inf-1.9] AND normal nucleoli= (-inf-1.9] ==> Class = 2	0.94
If uniformity of cell size= (-inf-1.9] AND normal nucleoli= (-inf-1.9] ==> Class = 2	0.94
If uniformity of cell shape= (-inf-1.9] ==> Class = 2	0.94
If bare nuclei= (-inf-1.9] AND normal nucleoli= (-inf-1.9] ==> Class = 2	0.94
If uniformity of cell shape= (-inf-1.9] ==> Class = 2	0.94

Table 10: Confidence value for the first ten rules when RENN noise filtering was applied without built-in pruning at a 5% noise ratio

Rule	Confidence
If uniformity of cell size= (-inf-0.9] AND normal nucleoli= (-inf-0.9] ==> Class = 0	1
If uniformity of cell size= (-inf-0.9] AND bare nuclei= (-inf-0] ==> Class = 0	1
If uniformity of cell size= (-inf-0.9] AND normal nucleoli= (-inf-0.9] AND mitoses= (-inf-0.9] ==> Class = 0	1
If uniformity of cell size= (-inf-0.9] AND bare nuclei= (-inf-0] AND mitoses= (-inf-0.9] ==> Class = 0	1
If uniformity of cell shape= (-inf-0.9] ==> Class = 0	1
If uniformity of cell shape= (-inf-0.9] AND mitoses= (-inf-0.9] ==> Class = 0	1
If uniformity of cell size= (-inf-0.9] AND mitoses= (-inf-0.9] ==> Class = 0	0.99
If bare nuclei= (-inf-0] AND normal nucleoli= (-inf-0.9] AND mitoses= (-inf-0.9] ==> Class = 0	0.99
If marginal adhesion= (-inf-0.9] AND bare nuclei= (-inf-0] AND mitoses= (-inf-0.9] ==> Class = 0	0.99
If uniformity of cell size= (-inf-0.9] ==> Class = 0	0.99

Table 11: Comparing classification accuracy results for cases C1, C2, and C3

Dataset	C1			C2			C3														
							DROP3			DROP5			ALLKNN			ENN			RENN		
	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R
Iris	63.33	68.00	63.00	**70.00**	**78.00**	**70.00**	**82.04**	**88.00**	**82.00**	**77.06**	**84.00**	**77.00**	**79.72**	**82.00**	**80.00**	**81.74**	**85.00**	**82.00**	**81.35**	**85.00**	**82.00**
Glass	37.38	41.00	37.00	34.11	35.00	34.00	**43.42**	35.00	**44.00**	**37.42**	27.00	**38.00**	**55.24**	**52.00**	**55.00**	**54.49**	**48.00**	**55.00**	**71.94**	**71.00**	**72.00**
Voting	80.23	80.00	80.00	80.23	80.00	80.00	64.10	62.00	64.00	60.62	60.00	61.00	62.66	69.00	63.00	62.16	69.00	62.00	62.12	69.00	62.00
Breast cancer	73.82	75.00	74.00	**76.11**	**79.00**	**76.00**	67.11	71.00	67.00	67.84	70.00	68.00	**96.19**	**96.00**	**96.00**	**94.94**	**95.00**	**95.00**	**94.90**	**95.00**	**95.00**
Liver	57.97	34.00	58.00	57.97	34.00	58.00	53.30	**47.00**	54.00	54.74	**36.00**	55.00	**82.27**	**82.00**	**82.00**	**80.63**	**80.00**	**81.00**	**83.81**	**83.00**	**84.00**
Diabetes	62.11	39.00	62.00	62.11	39.00	62.00	**63.12**	**70.00**	**63.00**	57.84	**61.00**	58.00	**64.94**	**79.00**	**65.00**	**63.94**	**78.00**	**64.00**	**64.52**	**79.00**	**65.00**
Banknote	54.45	30.00	54.00	54.45	30.00	54.00	**77.99**	**78.00**	**78.00**	**56.29**	**54.00**	**56.00**	**90.48**	**91.00**	**90.00**	**92.49**	**92.00**	**92.00**	**91.96**	**92.00**	**92.00**
Heart	49.84	25.00	50.00	49.84	25.00	50.00	**58.14**	**53.00**	**58.00**	**56.58**	**46.00**	**57.00**	**73.95**	**74.00**	**74.00**	**66.80**	**50.00**	**67.00**	**70.13**	**67.00**	**70.00**
Average	**59.89**	**49.00**	**59.75**	**60.60**	**50.00**	**60.50**	**63.65**	**63.00**	**63.80**	**58.55**	**54.80**	**58.80**	**75.68**	**78.10**	**75.60**	**74.65**	**74.60**	**74.80**	**77.59**	**80.13**	**77.80**

Table 12: Comparing classification accuracy results for cases C1, C2, and C4

Dataset	C1			C2			C4														
							DROP3			DROP5			ALLKNN			ENN			RENN		
	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R	CC	P	R
Iris	63.33	68.00	63.00	**70.00**	**78.00**	**70.00**	54.91	57.00	55.00	50.44	46.00	50.00	59.56	56.00	60.00	56.92	60.00	57.00	56.54	61.00	57.00
Glass	37.38	41.00	37.00	34.11	35.00	34.00	32.97	18.00	33.00	33.95	14.00	34.00	**52.98**	**51.00**	**53.00**	**53.49**	**46.00**	**53.00**	**60.40**	**60.00**	**60.00**
Voting	80.23	80.00	80.00	80.23	80.00	80.00	69.03	58.00	69.00	63.93	56.00	64.00	62.66	69.00	63.00	62.13	69.00	62.00	62.09	69.00	62.00
Breast cancer	73.82	75.00	74.00	**76.11**	**79.00**	**76.00**	71.26	73.00	71.00	68.78	71.00	63.00	**96.04**	**96.00**	**96.00**	**94.71**	**95.00**	**95.00**	**94.77**	**95.00**	**95.00**
Liver	57.97	34.00	58.00	57.97	34.00	58.00	53.68	**44.00**	54.00	55.00	**35.00**	55.00	**81.52**	**81.00**	**82.00**	**80.49**	**80.00**	**80.00**	**83.30**	**82.00**	**83.00**
Diabetes	62.11	39.00	62.00	62.11	39.00	62.00	55.16	37.00	55.00	51.82	36.00	52.00	55.76	**42.00**	56.00	56.12	**59.00**	56.00	55.21	**51.00**	55.00
Banknote	54.45	30.00	54.00	54.45	30.00	54.00	**56.99**	**56.00**	**57.00**	47.73	**43.00**	48.00	**89.49**	**90.00**	**89.00**	**88.24**	**89.00**	**88.00**	**85.94**	**88.00**	**86.00**
Heart	49.84	25.00	50.00	49.84	25.00	50.00	**59.52**	**47.00**	**60.00**	**57.10**	**45.00**	**57.00**	**73.32**	**74.00**	**73.00**	**66.80**	**50.00**	**67.00**	**69.34**	**64.00**	**69.00**
Average	**59.89**	**49.00**	**59.75**	**60.60**	**50.00**	**60.50**	**56.69**	**48.80**	**56.80**	**53.59**	**43.30**	**52.90**	**71.42**	**69.90**	**71.50**	**69.86**	**68.50**	**69.80**	**70.95**	**71.30**	**70.90**

The use of filtering algorithms also enhanced the confidence of the discovered association rules in both 0% and 5% noise cases. The confidence values of the association rules with the 10%-noise case will be compared. Tab. 13 shows the first ten rules produced for the Breast-Cancer dataset and their confidence without applying filtering or built-in pruning with a 10% noise ratio. Tab. 14 shows the first ten rules with their confidence produced after applying RENN. It can be noticed from Tab. 13 that the ten rules have a 0.9 confidence value, which is lower than both the 0% noise and 5% noise cases, and this is an expected result due to noise. Again, the confidence value for the constructed association rules improved when the RENN noise filter was applied; it ranges from 0.94 to 0.91, as shown in Tab. 14.

Table 13: Confidence value for the first ten rules when neither noise filtering nor built-in pruning was applied at a 10% noise ratio

Rule	Confidence
If uniformity of cell size= (-inf-1.9] AND uniformity of cell shape= (-inf-1.9] AND bare nuclei= (-inf-1.9] AND normal nucleoli= (-inf-1.9] ==> Class = 2	0.9
If uniformity of cell size= (-inf-1.9] AND uniformity of cell shape= (-inf-1.9] AND bare nuclei= (-inf-1.9] ==> Class = 2	0.9
If uniformity of cell shape= (-inf-1.9] AND bare nuclei= (-inf-1.9] ==> Class = 2	0.9
If uniformity of cell shape= (-inf-1.9] AND bare nuclei= (-inf-1.9] AND normal nucleoli= (-inf-1.9] ==> Class = 2	0.9
If uniformity of cell size= (-inf-1.9] AND uniformity of cell shape= (-inf-1.9] AND bare nuclei= (-inf-1.9] AND mitoses= (-inf-1.9] ==> Class = 2	0.9
If uniformity of cell size= (-inf-1.9] AND uniformity of cell shape= (-inf-1.9] AND normal nucleoli= (-inf-1.9] ==> Class = 2	0.9
If uniformity of cell shape= (-inf-1.9] AND bare nuclei= (-inf-1.9] AND mitoses= (-inf-1.9] ==> Class = 2	0.9
If uniformity of cell size= (-inf-1.9] AND uniformity of cell shape= (-inf-1.9] AND bare nuclei= (-inf-1.9] AND normal nucleoli= (-inf-1.9] ==> Class = 2	0.9
If single epithelial cell size= (1.9–2.8] AND bare nuclei= (-inf-1.9] AND normal nucleoli= (-inf-1.9] ==> Class = 2	0.9
If uniformity of cell size= (-inf-1.9] AND bare nuclei= (-inf-1.9] AND normal nucleoli= (-inf-1.9] ==> Class = 2	0.9

4.3 Effect of Pruning

As previously discussed, the results show that applying pruning without filtering did not improve the classification accuracy compared to using noise filtering alone. Tab. 15 illustrates the average accuracy for the used datasets on cases when neither pruning nor filtering (i.e., baseline) was used and when cases when only pruning was used at different noise ratios. It is shown that accuracy drops steadily as the noise ratio rises for the baseline and pruning-only experiments. Therefore, using pruning to overcome the noise effect is not a good choice.

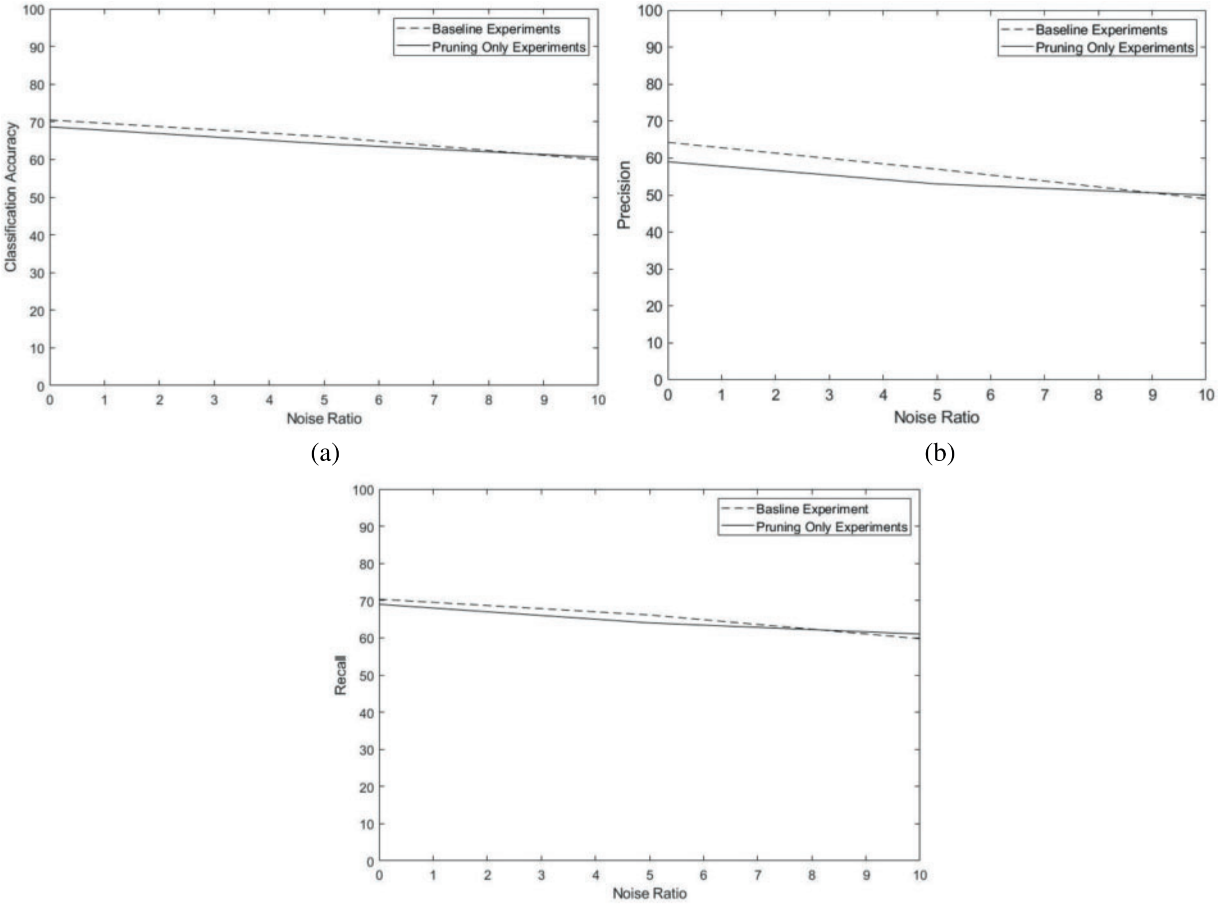
Table 14: Confidence value for the first ten rules when RENN noise filtering was applied without built-in pruning at a 10% noise ratio

Rule	Confidence
If uniformity of cell shape= (-inf-0.9] AND bare nuclei= (-inf-0] AND mitoses= (-inf-0.9] ==> Class = 0	0.94
If uniformity of cell shape= (-inf-0.9] AND bare nuclei= (-inf-0] ==> Class = 0	0.93
If uniformity of cell size= (-inf-0.9] AND bare nuclei= (-inf-0] AND mitoses= (-inf-0.9] ==> Class = 0	0.92
If bare nuclei= (-inf-0] AND normal nucleoli= (-inf-0.9] AND mitoses= (-inf-0.9] ==> Class = 0	0.92
If uniformity of cell size= (-inf-0.9] AND bare nuclei= (-inf-0] AND normal nucleoli= (-inf-0.9] AND mitoses= (-inf-0.9] ==> Class = 0	0.92
If uniformity of cell shape= (-inf-0.9] AND mitoses= (-inf-0.9] ==> Class = 0	0.92
If uniformity of cell size= (-inf-0.9] AND bare nuclei= (-inf-0] ==> Class = 0	0.92
If marginal adhesion= (-inf-0.9] AND bare nuclei= (-inf-0] AND normal nucleoli= (-inf-0.9] AND mitoses= (-inf-0.9] ==> Class = 0	0.92
If uniformity of cell shape= (-inf-0.9] ==> Class = 0	0.91
If uniformity of cell size= (-inf-0.9] AND uniformity of cell shape= (-inf-0.9] AND mitoses= (-inf-0.9] ==> Class = 0	0.91

Table 15: Average performance comparison between baseline experiments and pruning only experiments at different noise ratios

Noise ratio	Baseline experiments			Pruning only experiments		
	Correctly classified%	Precision %	Recall %	Correctly classified%	Precision %	Recall %
0%	70.47	64.25	70.38	68.64	59.00	69.00
5%	66.08	57.00	66.13	64.17	53.00	64.00
10%	59.89	49.00	59.75	60.60	50.00	61.00

Fig. 4 compares baseline experiments with Pruning only experiments at different noise ratios in terms of classification accuracy, Precision, and Recall. Both sets of experiments show performance degradation in the presence of noise. No noticeable improvement can be observed when pruning only was applied.



(a)

(b)

(c)

Figure 4: Comparison of average performance of baseline experiments with pruning only experiments at different noise ratios. a) classification accuracy, b) Precision, and c) Recall

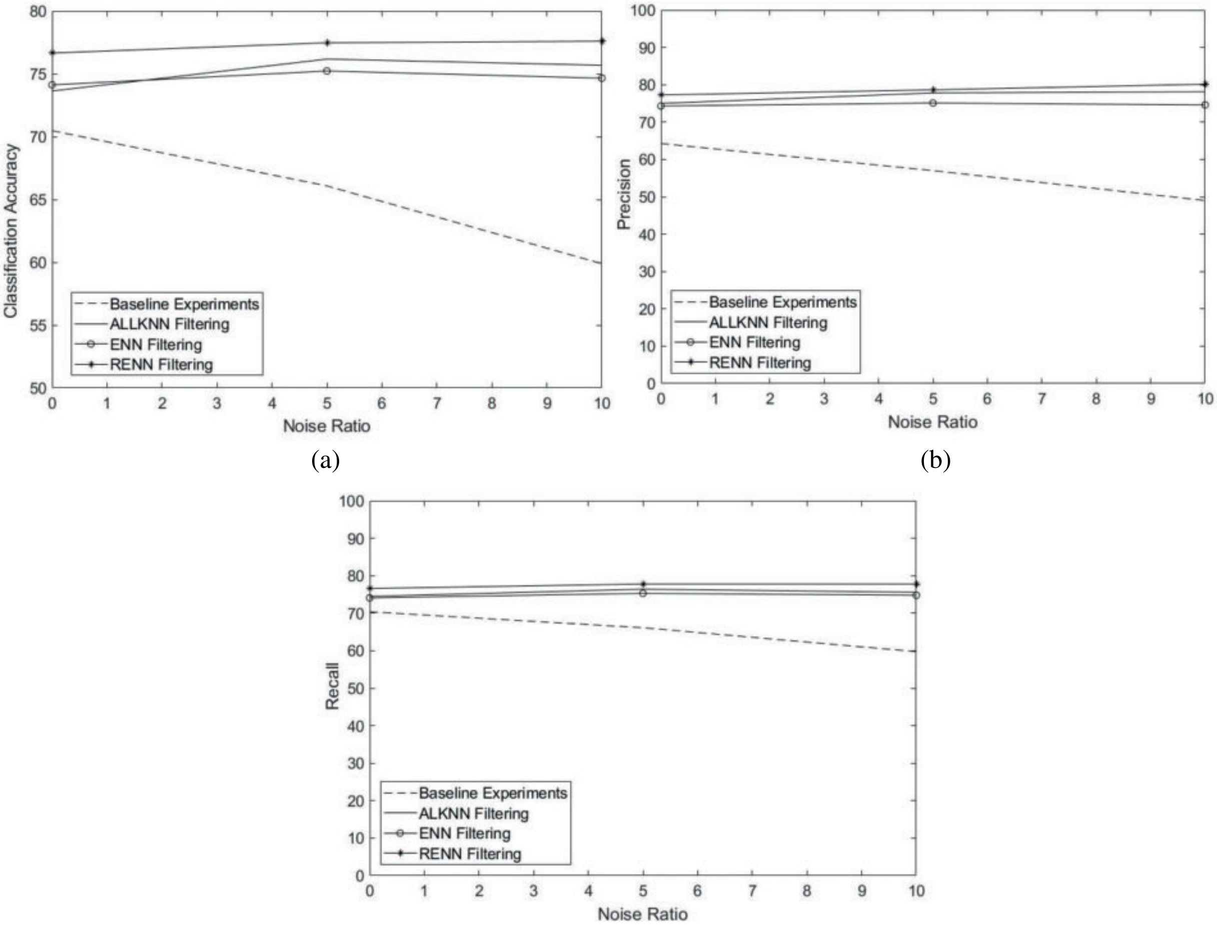
4.4 Effect of Filtering

This section compares the performance of the best three noise filters, namely: ALKNN, ENN, and RENN, with the baseline experiments' performance at different noise ratios.

The purpose of this experiment is to investigate whether implementing filtering algorithms alone will be a good choice to overcome the overfitting problem and improve the classification accuracy. Tab. 16 shows the average accuracy for the used datasets on the baseline experiments and with cases when ALLKNN, ENN, and RENN were used at different noise ratios. It is shown that accuracy drops steadily as the noise ratio rises for the baseline case. Noise filters, however, show greater resistance to noise. The best noise filter was RENN. Even the difference between the noise filter performance and baseline was more apparent with the increase in noise ratio. The accuracy difference between RENN and baseline was 6.18% (76.65–70.47), 11.39% (77.47–66.08), and 17.7% (77.59–59.89) with 0%, 5% and 10% noise ratios, respectively. Figs. 5a, 4b, and 4c compare baseline experiments with filtering only experiments at different noise ratios in terms of classification accuracy, Precision, and Recall, respectively.

Table 16: Average performance comparison between baseline experiments and filtering only experiments at different noise ratios

Noise ratio	Baseline experiments			ALLKNN			ENN			RENN		
	Correctly classified%	Precision %	Recall %	Correctly classified%	Precision %	Recall %	Correctly classified%	Precision %	Recall %	Correctly classified%	Precision %	Recall %
0%	70.47	64.25	70.38	73.63	75.00	74.50	74.12	74.30	74.10	76.65	77.25	76.60
5%	66.08	57.00	66.13	76.17	77.80	76.40	75.22	75.10	75.30	77.47	78.63	77.80
10%	59.89	49.00	59.75	75.68	78.10	75.60	74.65	74.60	74.80	77.59	80.13	77.80



(a)

(b)

(c)

Figure 5: Comparison of average performance of baseline experiments with filtering only experiments at different noise ratios. a) classification accuracy, b) Precision, and c) Recall

Applying instance reduction techniques as noise filters showed a significant improvement in classification accuracy, especially with a higher noise ratio. It appears that filtering techniques work better in higher noise domains, which is the desired result. The results of applying filtering alone were even higher than those of applying both filtering and pruning, as was the evidence in Tabs. 3, 4, 7, 8, 11, and 12. Filtering only is also less complex during the building of the classifier.

4.5 Statistical Analysis of the Results

Training data is a random sample from their populations in machine learning problems. These samples could be representative of their populations or not. This means that the results obtained from applying any technique vary depending on the data; a systematic way is needed to exclude the probability of getting extreme results due to sampling errors.

The null hypothesis of the learning problem needs to be tested and either accepted or rejected. The null hypothesis assumes that all treatments have similar means: H0: $\mu 1 = \mu 2 = \mu 3 \ldots = \mu k$.

If the null hypothesis is accepted, there is no significant difference among the tested algorithms. If the null hypothesis is rejected, it can be concluded that there is a significant difference among the tested algorithms. In case of rejection, paired t-Tests have to be conducted to show where the performance has been improved.

One-way Analysis of Variance (ANOVA) is one of the techniques used to test the null hypothesis. It computes the *p*-value, which determines whether there is a significant difference or not. *P*-value expresses the significance level. This value usually takes one of the values: 0.01, 0.05, or 0.10. In this research, 0.05 will be used, corresponding to 95% confidence in the results. Tab. 17 shows the ANOVA results on the 0% noise, 5% noise, and 10% noise cases.

Table 17: ANOVA-single factor for 0% noise, 5% noise and 10% noise. Each group has a count of 8

	0% noise			5% noise			10% noise		
Groups	Sum	Average	Variance	Sum	Average	Variance	Sum	Average	Variance
No pruning, no filtering	563.74	70.4675	381.489	528.64	66.08	371.1025	479.13	59.89125	180.7864
With pruning, no filtering	549.09	68.6363	453.96	513.33	64.1663	387.5751	484.82	60.6025	225.5392
DROP3 filtering	499.57	62.4463	160.838	514.13	64.2663	226.9585	509.22	63.6525	157.574
DROP5 filtering	479.61	59.9513	120.801	481.1	60.1375	135.7991	468.39	58.54875	129.0992
ALLKNN filtering	589.02	73.6275	272.573	609.39	76.1738	182.5565	605.45	75.68125	200.7292
ENN filtering	592.97	74.1213	275.383	601.75	75.2188	210.4981	597.19	74.64875	222.1044
RENN filtering	613.22	76.6525	184.279	619.79	77.4738	157.0744	620.73	77.59125	150.9541
DROP3+pruned	446.79	55.8488	79.6985	457.62	57.2025	138.3458	453.52	56.69	135.6953
DROP5+pruned	453.3	56.6625	86.3477	445.87	55.7338	70.20786	428.75	53.59375	112.1549
ALLKNN+pruned	557.57	69.6963	300.423	563.89	70.4863	265.1169	571.33	71.41625	262.9937
ENN+pruned	531.55	66.4438	361.838	560.67	70.0838	214.7943	558.9	69.8625	251.6669
RENN+pruned	560.19	70.0238	235.237	571.9	71.4875	191.626	567.59	70.94875	227.4864
ANOVA									
Source of variation	F	*p*-value	F crit.	F	*p*-value	F crit.	F	*p*-value	F crit.
Between groups	1.56049	**0.1258**	1.905	2.010671	**0.03716**	1.9045	2.860114	**0.00313**	1.9045

The overall *p*-value for the 0% noise is $0.1258 > 0.05$, indicating that there is no significant difference among these treatments. The null hypothesis is accepted; therefore, there is no need to perform paired t-Test for this case. This is not a surprising result as the superiority of applying noise filtering techniques is expected to be more apparent in noisy domains.

In Tab. 18, only DROP3-Pruned and DROP5-Pruned showed significant differences compared to the base case with p-values less than 0.05. They are significantly worse than the baseline case with 95% confidence.

Table 18: Paired t-test results for 5%-noise case

One-tailed p-value		With pruning, no filtering	Filtering techniques					Filtering + pruning				
			DROP3	DROP5	ALKNN	ENN	RENN	DROP3	DROP5	ALKNN	ENN	RENN
No pruning, no filtering		0.0965	0.3366	0.1426	0.1020	0.1153	0.0890	**0.0363**	**0.0274**	0.2971	0.3021	0.2570
With pruning, no filtering			0.4917	0.2428	0.0708	0.0808	0.0662	0.0557	0.0561	0.2144	0.2166	0.1858
Filtering techniques	DROP3			0.0586	0.0460	0.0479	0.0386	0.0569	0.0141	0.2143	0.2137	0.1782
	DROP5				0.0026	0.0022	0.0025	0.2422	0.0516	0.0544	0.0463	0.0380
	ALLKNN					0.1751	0.1941	0.0077	0.0013	0.0374	0.0119	0.0467
	ENN						0.1213	0.0105	0.0019	0.0988	0.0519	0.1394
	RENN							0.0082	0.0013	0.0478	0.0190	0.0326
Filtering + pruning	DROP3								0.2605	0.0323	0.0300	0.0251
	DROP5									0.0143	0.0107	0.0085
	ALLKNN										0.3336	0.2638
	ENN											0.1005
	RENN											-

The p-value for the 5% noise is 0.03716, which is less than 0.05, and it means that the reported improvement is statistically significant. The next step is to determine what treatments produced a significant improvement. Tab. 18 shows the results of paired t-Test for the 5%-noise case. P-values less than 0.05 indicate a significant difference. There is a particular interest in the difference compared to the base case (no pruning, no filtering). These values are marked in **boldface**.

Similarly, the p-value for the 10% noise (from Tab. 17) is 0.0313, which is less than 0.05, which means that the reported improvement is statistically significant. The next step is to determine what treatments produced a significant improvement. Tab. 19 shows the results of paired t-Test for the 10%-noise case. P-values less than 0.05 indicate a significant difference. Particular interest in the difference compared to the base case (no pruning, no filtering). These values are marked in boldface. In Tab. 19, RENN, ALKNN, and ENN showed significant differences compared to the base case with p-values less than 0.05. They are significantly better than the baseline case with 95% confidence. On the other hand, DROP5-Pruned is significantly worse than the baseline with 95% confidence.

Table 19: Paired t-test results for 10%-noise case

One-tailed p-value		With pruning, no filtering	Filtering techniques					Filtering + pruning				
			DROP3	DROP5	ALKNN	ENN	RENN	DROP3	DROP5	ALKNN	ENN	RENN
No pruning, no filtering		0.2507	0.2238	0.3549	**0.0148**	**0.0198**	**0.0137**	0.1070	**0.0214**	0.0599	0.0795	0.0685
With pruning, no filtering			0.2607	0.2679	0.0182	0.0235	0.0182	0.0986	0.0276	0.0786	0.1044	0.0918
Filtering techniques	DROP3			0.0434	0.0146	0.0165	0.0117	0.0722	0.0389	0.1261	0.1752	0.1525

(Continued)

Table 19: Continued

One-tailed p-value		With pruning, no filtering	Filtering techniques					Filtering + pruning				
			DROP3	DROP5	ALKNN	ENN	RENN	DROP3	DROP5	ALKNN	ENN	RENN
	DROP5				0.0028	0.0042	0.0033	0.2912	0.0940	0.0376	0.0555	0.0474
	ALLKNN					0.1697	0.2081	0.0020	0.0010	0.0664	0.0318	0.0830
	ENN						0.1061	0.0039	0.0022	0.1644	0.0789	0.1640
	RENN							0.0032	0.0014	0.0540	0.0227	0.0328
Filtering + Pruning	DROP3								0.0183	0.0102	0.0165	0.0149
	DROP5									0.0044	0.0068	0.0054
	ALLKNN										0.0455	0.3660
	ENN											0.1624
	RENN											-

4.6 Summary of Results per Dataset

To summarize the impact of using the best noise filtering algorithms on the eight datasets, Tab. 20 shows the number of datasets improved by applying ALLKNN, ENN and RENN compared with the base case. These algorithms produced the highest average accuracy.

Table 20: Number of improved datasets with ENN and RENN

	Number of improved datasets		
Case	0% noise	5% noise	10% noise
No pruning + ALLKNN	5	6	7
No pruning + ENN	5	7	7
No pruning + RENN	5	7	7
With pruning + ALLKNN	4	5	5
With pruning + ENN	4	5	6
With pruning + RENN	5	6	5

RENN was the best noise filtering algorithm in terms of classification accuracy and the number of improved datasets. It will further be analyzed to check its performance on each dataset. Tab. 21 compares the datasets that improved significantly with RENN compared to the baseline scenario using 0%, 5%, and 10% noise ratios. The + + implies a statistically significant improvement, while + implies an improvement but not statistically significant. Similarly, the−−indicates that accuracy decreased significantly, while-indicates that the decrease in accuracy was not significant. Comparing the number of improved datasets based on the numbers in Tab. 21, it can be noticed that in the zero-noise case, the classification accuracy with RENN improved statistically significantly in five out of eight datasets compared to the baseline.

In the 5%-noise case, seven datasets improved. In six of them, RENN improvement was statistically significant than the base case. In the 10%-noise case, seven datasets improved. In six of them, the improvement was statistically significant. For the remaining dataset, the difference was not significant. The Voting dataset is the only dataset that did not improve the classification accuracy for all noise ratios. A possible justification is that learners (in this case: Association rules) can be biased to specific data sets. It does not necessarily mean the dataset itself is inaccurate or not balanced.

Table 21: Comparison of significantly improved datasets with RENN for all noise-cases

Dataset	0% noise	5%-noise	10%-noise
Iris	−−	+	++
Glass	++	++	++
Voting	−−	−−	−−
Breast cancer-diagnostic	++	++	++
Liver	++	++	++
Pima-indians-diabetes	++	++	++
Banknote authentication	+	++	+
Heart-cleveland	++	++	++

5 Conclusions and Future Work

5.1 Conclusions

This research introduced a novel approach to increase association rules-based classification accuracy in noisy domains. The proposed approach applied instance reduction techniques to the datasets before generating association rules. This step works as a data cleaning procedure to eliminate noisy instances as much as possible before building the classifier. Unlike pre and post-pruning procedures that consume a large number of probably unnecessary computations, applying noise filtering algorithms will result in cleaner datasets and avoid extracting low confident association rules, leading to building an efficient association rules-based classifier on unseen examples.

The findings and contribution of this research are that:

- Five filtering algorithms were tested: DROP3, DROP5, ALLKNN, ENN, and RENN.
- The experiments were conducted on three noise levels: 0%, 5%, and 10% injected into datasets.
- Average classification accuracy improved remarkably compared to the base case where neither noise filtering nor built-in pruning was applied.
- The improvement in the classification accuracy was even more apparent with the increase in noise ratios, which is the intended goal of this research.
- Association rules' classification accuracy remarkably improved when applying ALLKNN, ENN and RENN, especially with higher noise levels, while the results of RENN were the most promising with a significant improvement in 7 out of 8 datasets, with 5% and 10% noise.
- As the use of filtering techniques led to removing noisy instances, this saved the unnecessary extraction of low confident association rules that contribute to the problem of overfitting.

RENN's average classification accuracy for the eight datasets in the zero-noise case improved from 70.47% to 76.65% compared to the base case when RENN was not used. The average accuracy was improved from 66.08% to 77.47% for the 5%-noise case and finally, from 59.89% to 77.59% in the 10%-noise case. This improvement in classification accuracy qualifies RENN to be an excellent solution to increase the accuracy of association rules classification, especially in noisy domains. It can be noticed that the improvement was more remarkable with the increase in noise levels.

5.2 Future Work

The idea of applying noise filtering algorithms to improve association rules classification accuracy can still be investigated to test its effectiveness on massive size datasets from different sources. Datasets available at www.data.gov, an open-source data repository of the US government that contains a large number of variant datasets in different fields.

Another research direction may involve applying instance reduction algorithms, as noise filters, on other machine learning techniques to improve classification accuracy, especially in noisy domains.

Dimensionality reduction algorithms such as Principal Component Analysis (PCA) can also be utilized to reduce the dimension of a learning problem. Dimensionality reduction minimizes the set of attributes measured in each itemset without affecting classification accuracy. This may enhance classification accuracy and reduce processing time significantly.

Acknowledgement: The APC was funded by the Deanship of Scientific Research, Saudi Electronic University.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] F. Berzal, I. Blanco, D. Sanchez and M. Vila, "Measuring the accuracy and interest of association rules: A new framework," *Intelligent Data Analysis*, vol. 6, no. 3, pp. 221–235, 2002.

[2] C. Wang and X. Zheng, "Application of improved time series Apriori algorithm by frequent itemsets in association rule data mining based on temporal constraint," *Evolutionary Intelligence*, vol. 13, no. 1, pp. 39–49, 2020.

[3] L. Fang and Q. Qizhi, "The study on the application of data mining based on association rules," in *The Int. Conf. on Communication Systems and Network Technologies*, Rajkot, Gujrat India, 2012, pp. 477–480.

[4] R. Agrawal, T. Imielinski and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. of the 1993 ACM SIGMOD Conf.*, Washington DC, USA, pp. 207–216, 1993.

[5] T. M. Mitchel, *Machine Learning*, New York, USA: McGraw-Hill, 1997.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed., USA: Springer, 2006.

[7] K. El-Hindi and M. Al-Akhras, "Smoothing decision boundaries to avoid overfitting in neural network training," *Neural Network World*, vol. 21, no. 4, pp. 311–325, 2011.

[8] K. Prasad, T. A. Faruquie, S. Joshi, S. Chaturvedi, L. V. Subramaniam *et al.,* "Data cleansing techniques for large enterprise datasets," in *The Annual SRII Global Conf.*, San Jose, CA, USA, pp. 135–144. 2011.

[9] D. Wilson and T. Martinez, "Reduction techniques for instance-based learning algorithms," *Machine Learning*, vol. 38, pp. 257–286, 2000.

[10] Z. Darwish, M. Al-Akhras and M. Habib, "Use filtering techniques to improve the accuracy of association rules," in *The IEEE Jordan Conf. on Applied Electrical Engineering and Computing Technologies (AEECT)*, Aqaba, Jordan, pp. 1–6, 2017.

[11] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. of the 20th Very Large Data Bases (VLDB) Conf.*, Santiago, Chile, pp. 487–499, 1994.

[12] J. Hipp, U. Guntzer and G. Nakhaeizadeh, "Algorithms for association rules mining-a general survey and comparison," *SIGKDD Explorations*, vol. 2, no. 1, pp. 58, 2000.

[13] C. Borgelt, "An implementation of the FP-growth algorithm," in *Proc. of the 1st Int. Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations*, Chicago Illinois, United States, pp. 1–5, 2005.

[14] B. Liu, W. Hsu and W. Ma, “Pruning and summarizing the discovered associations,” in *The ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, San Diego California, USA, pp. 125–134, 1999.

[15] M. Bramer, “Using J-pruning to reduce overfitting of classification rules in noisy domains,” *Springer-Verlag Berlin Heidelberg*, vol. 2453, pp. 433–442, 2002.

[16] B. Krawczyk, I. Triguero, S. García, M. Woźniak and F. Herrera, “Instance reduction for one-class classification,” *Knowledge and Information Systems*, vol. 59, no. 3, pp. 601–628, 2019.

[17] R. M. Cameron-Jones, “Instance selection by encoding length heuristic with random Mutation Hill Climbing,” in *Proc. of the Eighth Australian Joint Conf. on Artificial Intelligence*, Canberra, Australia, 1995, pp. 99–106.

[18] I. Tomek, “An experiment with the edited nearest neighbor rule,” *IEEE Transactions on Systems, Man and Cybernetics SMC*-vol. 6, pp. 448–452, 1976.

[19] D. Wilson, “Asymptotic properties of nearest neighbor rules using edited data,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-2, no. 3, pp. 408–421, 1972.

[20] L. Yang, Q. Zhu, J. Huang and D. Cheng, “Adaptive edited natural neighbor algorithm,” *Neurocomputing*, vol. 230, pp. 427–433, 2017.

[21] P. Juryon, “Efficient tree-structured items pruning for both positive and negative association rules,” *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 17, pp. 4697–4711, 2019.

[22] X. Dong, F. Hao, L. Zhao and T. Xu, “An efficient method for pruning redundant negative and positive association rules,” *Neurocomputing*, vol. 393, pp. 245–258, 2019.

[23] G. Du, Y. Shi, A. Liu and T. Liu, “Variance risk identification and treatment of clinical pathway by integrated Bayesian network and association rules mining,” *Entropy*, vol. 21, no. 12, pp. 1–15, 2019.

[24] Q. Yang, S. -Y. Wang and T. -T. Zhang, “Pruning and summarizing the discovered time series association rules from mechanical sensor data,” in *The 3rd Annual Int. Conf. on Electronics, Electrical Engineering and Information Science (EEEIS 2017)*, Atlantis Press, Guangzhou, China, pp. 40–45, 2017.

[25] M. K. Najafabadi, M. N. Mahrin, S. Chuprat and H. M. Sarkan, “Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data,” *Computers in Human Behavior*, vol. 67, pp. 113–128, 2017.

[26] T. Yang, K. Qian, D. C. -T. Lo, Y. Xie, Y. Shi *et al.,* “Improve the prediction accuracy of Naïve Bayes classifier with association rule mining,” in *The 2016 IEEE 2nd Int. Conf. on Big Data Security on Cloud (BigDataSecurity), IEEE Int. Conf. on High Performance and Smart Computing (HPSC), and IEEE Int. Conf. on Intelligent Data and Security (IDS)*, New York, NY, USA, pp. 129–133, 2016.

[27] H. T. Nguyen, H. H. Huynh, L. P. Phan and H. X. Huynh, “Improved collaborative filtering recommendations using quantitative implication rules mining in implication field,” in *Proc. of the 3rd Int. Conf. on Machine Learning and Soft Computing*, Da Lat, Vietnam, pp. 110–116, 2019.

[28] T. Zhang, L. Ma and G. Yang, “MOPNAR-II: An improved multi-objective evolutionary algorithm for mining positive and negative association rules,” in *2019 IEEE 31st Int. Conf. on Tools with Artificial Intelligence (ICTAI)*, Oregon, USA, pp. 1648–1653, 2019.

[29] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann *et al.,* “The WEKA data mining software,” *ACM SIGKDD Explorations Newsletter*, vol. 11, pp. 10, 2009.

[30] C. Blake and C. Merz, “UCI repository of machine learning databases,” [Online]. Available: https://archive.ics.uci.edu/ml. 2007.