


# Gürültülü Verilere Dayanıklı kNN Algoritması Temelli Yeni Bir İkili Sınıflandırma Algoritması

*Araştırma Makalesi/Research Article*

 Müge ACAR

Endüstri Mühendisliği Bölümü, Eskişehir Teknik Üniversitesi, Eskişehir, Türkiye

[msoyuz@eskisehir.edu.tr](mailto:msoyuz@eskisehir.edu.tr)

(Geliş/Received:16.08.2024; Kabul/Accepted:15.10.2024)

DOI: 10.17671/gazibtd.1534334

**Özet**— Sınıflandırma, belirlenmiş bazı kriterlere göre kategoriler halinde sistematik olarak verilerin analizinde kullanılan etkili bir tekniktir. Sınıflandırıcının başarısı, sınıflandırıcının kendisine ve verilerin kalitesine bağlıdır. Bununla birlikte, gerçek hayat uygulamalarında, veri kümelerinin yanlış etiketlenmiş örnekler içermesi kaçınılmazdır. Gerçek hayat verileri gürültü olarak bilinen yanlış etiketlenmiş örnekler içerebilir. Bu da yanlış sınıflandırmalara neden olabilir. Bu çalışma, yeni bir kNN (k en yakın komşuluk algoritması) tabanlı sınıflandırma algoritması ile gürültü verilerinin sınıflandırılmasının nicel bir değerlendirmesini ve verileri verimli bir şekilde sınıflandırarak klasik kNN'nin performansını artırmayı amaçlamaktadır. Bu yeni tekniğin, gürültü verileriyle ikili sınıflandırma problemlerinde yüksek standart doğruluk seviyeleri sağlayabileceğini öneriyoruz. Bu çalışma, sınıflandırmadan önce gürültü noktaları tespit edilmesini dikkate alarak ikili sınıflandırma problemlerinde kNN tekniğinin performansını arttırmaktadır. Yeni kNN ve klasik kNN algoritmalarını farklı gürültü seviyelerinde (%10, %20, %30 ve %40) farklı veri setlerinde test doğruluğu açısından ölçerek karşılaştırdık ve özellikle yüksek gürültü seviyelerinde %84.33, %93.63, %81.81, %88.00 ye varan test doğruluk değerleri ile klasik kNN algoritmasına göre yüksek değerler elde edildi. Ayrıca geliştirdiğimiz algoritmayı popüler sınıflandırma algoritmalarıyla karşılaştırdığımızda bazı verilerde daha yüksek doğruluk değerleri elde edilerek %85.19, %97.07, %96.63'e varan doğruluk değerleri gözlemlendi.

**Anahtar Kelimeler**— ikili sınıflandırma, gürültü verisi, veri madenciliği

## A New Binary Classifier Robust on Noisy Domains Based on kNN Algorithm

**Abstract**— Classification is an effective technique commonly used in data analysis by systematically arranging groups or categories according to established criteria. The classifier's success relies on the classifier itself and the quality of the data. However, in real-world applications, it is inevitable for datasets to contain mislabeled instances, which may cause misclassification challenges that classifiers have to handle. This study aims for a quantitative assessment of the classification of noisy data through a new kNN-based classification algorithm and to increase the performance of classical kNN by efficiently classifying the data. We perform various numerical experiments on real-world data sets to prove our new algorithm's performance. We obtain high standards of accuracy levels on various noisy datasets. We propose that this new technique can provide high standard accuracy levels in binary classification problems. We compared the new kNN and the classical kNN algorithms in terms of test accuracies on different datasets under varying noise levels (10%, 20%, 30%, and 40%). Particularly at higher noise levels, we achieve significantly higher test accuracy values compared to the classical kNN algorithm, with results reaching 84.33%, 93.63%, 81.81%, and 88.00%. Additionally, when we compare our new algorithm with popular classification algorithms, we observed higher accuracy rates on some datasets, with values reaching up to 85.19%, 97.07%, and 96.63%.

**Keywords**— binary classification, noisy data, data mining

## 1. INTRODUCTION

In recent years, data science has high impact in many areas and continues to develop. Various decisions such as budget planning, sales strategies even social media analysis are made based on the data analysis. In all kinds of analysis, classification is commonly applied in making such crucial decisions. All classification techniques perform the analysis based on the training data. However, the training data may not always be perfect. In real-life, the data often contain noise. According to Zhu and Wu [5], noise refers to any irrelevant or meaningless data that interferes with the processing, transmission, or interpretation of information. They states that noise can occur in various forms such as random errors, distortions, or unwanted signals that affect the quality and accuracy of data or communication. Bishop explains that [30] in data science and machine learning, noise is often considered as random variations or outliers in data that do not represent the underlying patterns or trends. Goodfellow et. al. [29] mention that noise can degrade the performance of algorithms and models, leading to inaccurate results or predictions. So according to studies states about noise reveals that noisy data can be challenging for classifiers to carry out the qualified separations.

According to the literature, different types of noisy points are classified. According to Catal [26], noisy points can be seen in different data parts. First data labels may be incorrect. This type is defined as class noise. Second type is called attribute noise, that is about wrong attribute values. Third type is seen as the combination of these two types. Catal [26] says that class noise has performance decreasing effects on accuracy of data rather than attribute noise. This fact can be explained as two main reasons: One of them is that for each data sample, there are multiple attributes but there is a single label value. While it is possible to compensate an error on a certain attribute with another attribute, the error on the label will be less tolerable. Other reason may be explained that as each attribute can have a bold or weak effect on learning methods but labels always have strong effects on sample data. Therefore, we considered class noise in our study.

In the literature, many classification techniques have been presented to perform classification of noisy data in different ways. These may be generalized in two different approaches. The first approach involves pre-processing of the data. In this approach, noise points in the data is cleaned by filtering, and then the cleaned data is classified using a classification algorithm. Although this seems to be an effective way, it can be long and difficult to implement because it requires two different stages to implement. Other approach is directly classify the data containing noise. In this case, the algorithm should have a mechanism to classify the data by considering there will be noise points.

There are many popular classification algorithms (e.g., SVM, decision trees, neural networks, kNN, etc.) that have been used in the literature. Among these, k nearest neighbor (kNN) is an effective and powerful lazy learning algorithm that is also easy-to-implement. However, its performance heavily relies on the quality of training data. Since this algorithm is dependent on data quality, it makes strong predictions when quality data is available, but may not achieve the expected success when there exist data

losses or noisy data. Considering the fact that the real-life data cannot also be controlled and may contain noise or missing values, the kNN algorithm led to decision makers to make wrong decisions.

In this study, we present a new method based on kNN algorithm to be used in the classification of data containing class noise. Our method does not need to consider whether the data is noisy or if needed for preprocessing. We argue that the new algorithm, which is not data dependent and easy to implement, will yield more consistent predictions. We demonstrate this claim with numerical studies in which we test our algorithm with the data containing 10%, 20%, 30%, 40% noise level and compare the results obtained with those of the classical kNN algorithm. The results we obtain indicated that the estimation success of the new kNN algorithm remained more stable even if the noise level increased, whereas the estimation success of the classical kNN algorithm decreased with increasing the noise level. We also compared the test accuracy values of our new algorithm and those of popular classification algorithms in the literature. Moreover, it is also shown that the test success of the new method is independent of the parameter.

## 2. LITERATURE REVIEW

In the classification literature, there are several studies that consider classification with noise. ([1]; [2]; [3]; [4]; [19];[20]; [18];[27];[28]). It is seen that different evaluation methods have been used in these studies. Some studies examine noisy data classification with pre-processing. They prefer to process and clean the data beforehand and classify the cleaned data with a classification algorithm. Saez and Corhado [4] consider pre-processing the data first to repair the attribute noise. They proposed not to filter the data but repair it to increase the performance of the classification algorithm. Sluban and Lavrač [6] suggest filtering the noise. They explored the class noise detection by studying different diversity measures on a range of heterogeneous noise detection ensembles. Luengo et al [7] utilize bag noise filtering for negative instance noise cleaning. Garcia et al [8] focus on a new label noise injection method for the evaluation of noise filters. Wang et al [9] used a new method for cleaning noise by using the LNC-SDAE framework. Mansour et al [31] proposed an adaptive synthetic sampling-based noise detection technique for mobile edge computing. Dash et al [32] suggest an outliers detection and elimination framework in classification, they consider winsorizing method to deal with the outliers.

Unlike the studies focusing on pre-processing, some studies in the literature suggested that pre-processing followed by classification may become complex and be difficult to implement. Thus, they adopted the idea of using an extended classification algorithm for noise data and performing classification in one stage. There are several successful algorithms (e.g., SVM, decision trees, neural networks, kNN, etc.) implemented that provide successful test accuracy results assuming the data is clean. Although these methods are successful in data classification, since their estimation capabilities decrease when noise level increases in data, their new versions have been used in the literature and therefore these new versions are directly resistant to noise data.

Marsala and Petturiti [10] considered a decision tree based method to classify the noisy data. They focus on order discrimination criteria to be used in decision tree induction, that is, functions that can measure the discriminatory power of an attribute with respect to class, taking into account the monotony of the class with respect to the attribute. Zhu et al [11] proposed a new classification technique noise data by using neural networks. They demonstrate that their method has an effective classification ability while comparing the methods in literature. Chao et al [12] developed a new SVM (support vector machine) technique (RTS-SVM) for classifying data with high percentage of noise. They conclude that the main advantage of the proposed SVM is to improve the classification performance by avoid the effect of the entangled noise data when compared with the classic SVM. When we evaluate these methods, either it is difficult and complex to implement but the quality of classification has increased, or the method has compromised success despite being easy to implement. In our opinion, this between implementation difficulty and test success stands out as a trade-off that is worth to be evaluated.

kNN is an effective classification algorithm which is very popular, easy to implement and have successful test results. It has been widely used for clustering and classification of data for various of different fields. ([15]). However, in recent years due to the complexity of the data and the demand to produce accurate and high-quality information with new technological developments, the performance of the kNN algorithm is not sufficient. Thus, there are other studies in the literature that have focused on increasing the performance of the kNN algorithm and improve the disadvantages of the kNN. Some of these studies are based on finding k close samples in the attribute space. Pedrajas and Boyer [14] develop two method for boosting k-NN. Their two approaches both modify the view of the data that the classifier receives so that the accurate classification of difficult instances is favored. Liao and Vemuri, [13] proposes a new technique for develop a new kNN boosting algorithm. Their approach employs the k-Nearest Neighbor (kNN) classifier to categorize each new program behavior into either normal or intrusive classes. Liu et al [25] proposed a kNN based technique. They propose a critical time difference fall incident detection system to detect fall incident events. In addition, some studies focused on big data classification with the kNN method. García-Gil et al., [8] consider a study that focus on suitable noise filtering approach in big data domains. They apply two algorithms removing noisy examples composed of Random Forest, Logistic Regression and K-Nearest Neighbors (KNN) classifiers. [24] consider kNN based classifier for big data applications. They focus on grouped the data into some partitions to classify the whose data in accurate, fast, and robust manner. Also the re are some studies concentrated on the pre-processing of data by kNN as a filtering technique. Triguero et al., [16] apply kNN based filtering algorithm. They focus to mitigate the computational complexity, storage requirements and noise tolerance by eliminating redundant, irrelevant and noisy information. Maillo et al [21] consider the Spark framework due to its balance between scalability and accuracy that improves previous kNN proposals in the literature. Li et al [33] propose an algorithm named quantum kNN algorithm in order to speed up the classical algorithm. Considering the studies in the literature there are many techniques that tackle with class noise and conduct

effective classifications with extra many stages but it is still a need to develop easy implemented and high quality classifier which is aimed to classify noise data even with high level of noise.

In this study, we propose a new classification algorithm that is resistant to noisy data without making any prior application to the data. We generate a new noise robust kNN (k nearest neighborhood) based algorithm in order to classify any data that may include noise even at high percentage. We extend the classical kNN algorithm to provide the algorithm to show more noise robust performance. We focus on the algorithm is easy to implement and also make qualified classifications with even high level of noise. We conduct our experiments in the aim of mentioned ideas. We make numerical experiments for real datasets as for the traditional kNN method and proposed kNN method for different k parameters. The experimental results indicate that the proposed method has high performance and robust results on different datasets. Also, it is shown that the proposed algorithm is parameter independent. Moreover, all the comparisons are evaluated on the datasets when the class label of data samples (10%, 20%, 30%, 40%) are changed to be considered as noisy data. The results indicate that the method can be considered as a robust to noise method that its performance is comparable with other methods.

In the following section, the kNN algorithm is briefly described. In Section 4, the proposed algorithm with an illustrative example is given. Section 5 provides numerical results, and Section 6 concludes the study.

### 3. METHOD

#### 3.1. kNN Algorithm

The kNN is a supervised learning algorithm used for classification. Unlike other supervised learning algorithms, it does not have a training stage. The kNN algorithm includes a lazy type of learning in which training and testing stages are pretty much the same thing. It basically searches the closest points to the new point and classifies each point according to nearest neighborhoods. The parameter k represents the number of nearest neighbors of the unknown point. The mechanism of the kNN algorithm has been explained, but like all classification algorithms, it has some disadvantages. Those disadvantages are itemized below to highlight the important aspects of new n-kNN algorithm that we develop.

The classical kNN algorithm may cause several problems:

- As it is known, the kNN algorithm searches the distances between all points in the sample set. Then it labels a new point determined by the maximum number of labels in k nearest neighborhood set. This means that the kNN algorithm heavily relies on sample data which may include noise.
- Because of its sensitiveness to noise, the algorithm performs well on some data sets while not on others according to the noise level.

In our new n-kNN algorithm, we proposed solutions for these disadvantages as explained in the next section.

### 3.2. The Proposed n-kNN Algorithm

The n-kNN algorithm focuses on increasing its reliability, since it may decrease in dealing noise data due to its heavily dependence on data.

Suppose we have a data set with two classes, named A and B. Assume that the sample data set is given as  $sampleList = \{sampleList_1, sampleList_2, sampleList_3, \dots, sampleList_n\}$ . The goal is to design a mechanism that detects if a point in kth nearest neighborhood is noise. The algorithm may decide whether the point is noise or not. The algorithm classifies the data according to this information. The steps of n-kNN algorithm are presented below.

```

BEGIN
SET sampleList = ( 1 ≤ COUNT(sampleList) ≤ n);
INPUT unknownSample;

NUMERIC k = ( 1 ≤ k ≤ n);

SET nearestNeighbourSamples =
(nearestNeighbourSamples count = k);

NUMERIC i = 1, j=1;

// ---- SIGNING NOISE ----
FOR i = 1 to COUNT(sampleList)
  FOR j = 1 to COUNT(sampleList)

    estimatedClassSampleListi = CALL
    kNN(sampleListj, sampleListi);

    IF(estimatedClassSampleListi != sampleListi.class)
    THEN
      sampleListi.noise = TRUE;
    END IF
  END FOR
END FOR

// ---- CLASSIFICATION ----

FOR i = 1 to COUNT(sampleList)
  CALL EuclidDistance(sampleListi, unknownSample);

  IF (sampleListi.noise != TRUE) THEN
    IF ( i ≤ k) THEN
      INCLUDE sampleListi to the
      nearestNeighbourSamples;
    ELSE
      IF (sampleListi is closer unknownSample than
      nearestNeighbourSamples) THEN
        ELEMENATE the farthest neighbour in the
        nearestNeighbourSamples;

        INCLUDE sampleListi to the
        nearestNeighbourSamples;
      END IF
    END IF
  END IF
END FOR

DETERMINE the maximum class label in the
nearestNeighbourSamples;
ASSIGN unknownSample into the maximum class label
category;

```

END;

In the n-kNN algorithm, we perform a determination of noise labeling for each data points in the sample (even if the class of the point is already known). First, we aim to determine the distances between all sample points  $p_i$  and find k nearest neighbors for each. According to k neighbors, the number of points belongs to different classes was considered. If the number of points belonging to which class is higher in that neighborhood, the point to be classified is accepted as a point belonging to that class. According to this determination, we find the class of  $p_i$  and compared the original class with the one that we detected from neighborhoods. If the previously known class and the one calculated are not the same, then that point is labeled as noise. After that, resembling to the classical kNN algorithm structure,

the n-kNN algorithm determines the distances for the new point, then selects k nearest neighbors for the new data point. If there is a data point that is predicted as noise by n-kNN mechanism, then the algorithm do not take that point into consideration and selects the next nearest point instead of the noise point. Then, the algorithm decides the new point's class according to its new neighbours. The algorithm performs all these steps and stops.

In order to explain the performance difference between the n-kNN algorithm and classic k-NN algorithm, we present an illustrative example. In the example, two classes are defined, class A and class B. The class A is denoted with green squares and the class B with red pentagons. Let's assume that there is a blue circle point which is a test point to be classified. It actually belongs to class A and is named as  $p_0$ . For both kNN and n-kNN algorithms, neighborhoods for  $p_0$  are determined for different k parameters as  $k=\{3,5,7\}$ . When the performance of two algorithms is evaluated in the same example for  $k = 3$  both k-NN and n-kNN algorithms considers that  $p_0$  belongs to class A with 3 green squares in  $k=3$  neighborhood. For  $k=5$  both algorithms classify  $p_0$  as class A with 3 green squares and 2 red pentagons. But for  $k=7$ , the algorithms yield different results for  $p_0$ . According to the classical kNN for the  $k=7$  neighborhood, there are 4 red pentagons as class B and 3 green squares as class A. Thus, it is obtained that  $p_0$  belongs to class B. But according to the n-kNN algorithm, there are two noise points and those are marked as noise. Those two points are not considered to belong in the neighborhood and next two nearest points are searched. Consequently, for  $k=7$  neighborhood, the n-kNN algorithm classifies the  $p_0$  point as class A according to 5 green squares and 2 red pentagons. When all these are taken into account, without any noise consideration, there can be some misclassifications for any points. Our n-kNN algorithm takes into account the detection of noise points as it can be seen in Figure 1 and Figure 2.

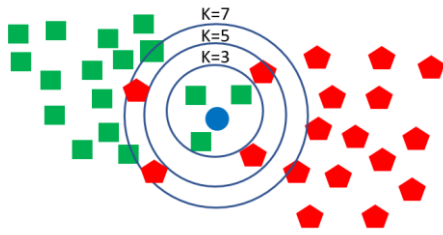


Figure 1. Illustrative example classified by classic kNN algorithm

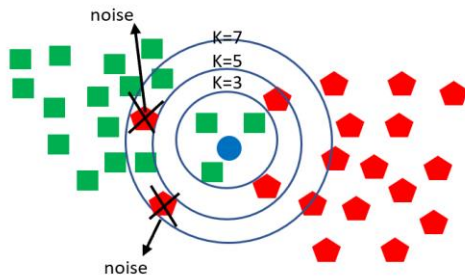


Figure 2. Illustrative example classified by n-kNN algorithm

Dataset	Short Names	# of points	# of features
Fertility	FERTILITY	100	9
Hepatitis	HEPATIT	142	19
Hearth	HEART	303	13
Ionosphere	ION	351	34
Wisconsin	WIS	683	10
Pima	PIMA	768	8
Data Banknote	DATA	1372	4
Spambase	SPAM	4601	57
Puma	PUMA	8192	32
Htru	HTRU	17898	8

Table 1 Description of the datasets.

#### 4. RESEARCH FINDINGS AND DISCUSSION

In this section, the numerical results of the n-kNN algorithm applied to real-world datasets are presented and compared with the results obtained by the popular methods in the literature. 10 binary datasets from the UCI repository have been used in our experiments [22]. The dataset information is given in Table 1. We perform 10-fold cross-validation method to conduct our

experiments for each dataset. We conduct numerical experiments according to three aspects: Firstly, it is aimed to show the

comparison between the accuracies of the proposed n-kNN algorithm and the classical kNN algorithm with different levels of noise (%0, %10, %20, %30, %40). In order to carry out the noise data experiments, we add random class noise with different levels using Weka filters. We follow a procedure described by Mantas and Abell'an (2014b) to add random class noise. We use the default parameter settings in Weka for applying the numerical results just as Mantas and Abell'an (2014b) applied. Also we use same dataset for each comparison. Secondly, In order to show the robustness of our algorithm we conduct experiments with different k parameter values ( $k = \{3,5,7,9\}$ ). It is important to show that the accuracy results of the proposed algorithm are also successful for different parameter values, and the algorithm should be able to give consistent results regardless of the parameter value. Finally, we compared the n-kNN algorithm with different methods (support vector machines (SVM), Credit 4.5 (C 4.5), random forest (RF), artificial neural networks (ANN), Decision Tree (DT)) recently used in the literature. We apply those methods SVM, C 4.5, RF, ANN, DT algorithms via Weka tool as Mantas and Abell'an (2014b) applied. We use the default parameters on Weka to apply the analysis.

We use accuracy in our study as the performance measures in our comparisons. It is the mostly common measure for benchmarking in data mining field. It is based on a the consideration that a test sample could be either a false positive (FP), or a false negative (FN), or a true positive (TP), or a true negative (TN). If the classifier predicts the label of the positive and negative test samples correctly, they are named TP and TN, respectively. If the test sample is classified into a positive class, while it is negative, it is called false positive (FP). If it is classified as a negative, but it is positive, it is known as false negative (FN). Moreover, It is given the accuracy calculation according to Eq(1).

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FN + FP)} \quad (1)$$

The experiments with 0%, 10%, 20%, 30% and 40% levels of noise in each dataset are performed to compare the n-kNN algorithm and the classical kNN algorithm for different k parameters. It is given in Table 2. Table 2 shows that better accuracy results are obtained by n-kNN algorithm when comparing classical kNN algorithm with different k parameter values. Although the values of the parameter k changes, the n-

Table 2 Noise data test accuracy results for kNN and n-kNN for different noise levels and k values

DATA SET	NOISE LEVEL %	kNN k=3	n-kNN k=3	kNN k=5	n-kNN k=5	kNN k=7	n-kNN k=7	kNN k=9	n-kNN k=9
ION	%0	<b>84.33</b>	82.34	84.90	84.90	83.48	<b>84.33</b>	83.20	83.77
	%10	83.19	82.91	84.04	<b>85.19</b>	83.48	84.06	83.77	83.48
	%20	76.94	82.91	78.91	84.05	80.63	84.05	80.64	<b>84.06</b>
	%30	70.08	84.33	72.36	84.33	74.91	84.33	76.05	<b>84.34</b>
	%40	60.36	83.18	62.93	84.04	60.64	<b>84.33</b>	64.94	82.90
WIS	%0	<b>97.21</b>	96.92	97.65	<b>97.21</b>	97.07	96.92	97.22	96.77
	%10	93.70	96.92	96.19	96.78	96.19	<b>97.07</b>	96.49	96.92
	%20	86.62	<b>96.92</b>	91.21	96.78	93.99	96.63	95.46	96.33
	%30	77.45	<b>97.36</b>	80.98	96.33	86.66	96.33	88.86	96.48
	%40	60.04	<b>96.63</b>	62.95	96.04	68.94	95.75	73.78	96.04
HEPATITIS	%0	75.43	<b>81.81</b>	79.00	80.38	79.00	81.81	79.67	<b>81.81</b>
	%10	70.62	<b>81.81</b>	74.14	79.67	76.29	<b>81.81</b>	74.95	<b>81.81</b>
	%20	65.67	<b>82.52</b>	70.62	81.81	72.10	81.81	75.62	81.81
	%30	60.48	81.10	62.05	<b>81.81</b>	67.71	<b>81.81</b>	71.19	<b>81.81</b>
	%40	55.57	<b>80.38</b>	57.57	<b>81.81</b>	60.43	<b>81.81</b>	63.95	<b>81.81</b>
HEART	%0	62.32	<b>64.71</b>	62.02	64.34	59.39	65.35	64.34	66.32
	%10	62.02	62.70	62.71	<b>63.70</b>	59.39	65.68	63.69	65.34
	%20	55.81	63.73	60.12	<b>64.70</b>	58.74	63.74	60.76	63.4
	%30	57.75	<b>63.39</b>	58.08	63.38	58.4	63.39	58.76	63.38
	%40	56.12	61.71	55.82	<b>63.09</b>	56.15	61.42	56.8	62.41
FERTILITY	%0	87.00	87.00	87.00	88.00	88.00	88.00	88.00	88.00
	%10	84.00	<b>88.00</b>	85.00	88.00	86.00	88.00	87.00	88.00
	%20	75.00	86.00	77.00	<b>88.00</b>	79.00	<b>88.00</b>	79.00	<b>88.00</b>
	%30	70.00	<b>89.00</b>	72.00	88.00	74.00	88.00	77.00	88.00
	%40	52.00	86.00	56.00	<b>88.00</b>	56.00	<b>88.00</b>	67.00	<b>88.00</b>
SPAM	%0	<b>80.61</b>	78.79	80.13	78.31	79.87	77.55	79.85	76.98
	%10	77.70	78.27	78.40	78.11	78.44	77.03	<b>78.59</b>	76.22
	%20	72.74	<b>77.77</b>	73.70	77.37	75.35	76.66	75.61	75.92
	%30	67.77	<b>77.72</b>	68.94	77.66	70.46	76.4	71.77	75.85

	<b>%40</b>	58.62	<b>77.37</b>	59.47	76.55	61.25	75.66	60.77	75.33
<b>PUMA</b>	<b>%0</b>	57.85	61.97	59.22	63.45	60.28	64.39	61.29	<b>64.50</b>
	<b>%10</b>	56.24	61.69	57.42	63.61	58.35	64.12	59.02	<b>64.33</b>
	<b>%20</b>	54.38	61.94	55.71	63.68	56.62	64.22	57.68	<b>64.48</b>
	<b>%30</b>	54.04	62.11	55.05	62.99	55.47	63.90	55.6	<b>64.32</b>
	<b>%40</b>	51.68	60.90	51.23	62.61	51.70	63.48	52.31	<b>63.87</b>
<b>PIMA</b>	<b>%0</b>	64.99	64.99	64.99	64.99	64.99	64.99	64.99	64.99
	<b>%10</b>	61.05	61.35	61.05	<b>64.99</b>	58.93	<b>64.99</b>	61.5	<b>64.99</b>
	<b>%20</b>	60.88	64.08	59.82	<b>64.99</b>	64.53	<b>64.99</b>	64.53	<b>64.99</b>
	<b>%30</b>	62.26	64.99	56.66	<b>64.99</b>	58.31	<b>64.99</b>	56.92	<b>64.99</b>
	<b>%40</b>	52.87	61.05	52.24	<b>64.99</b>	51.74	<b>64.99</b>	58.00	<b>64.99</b>
<b>DATA BANKNOTE AUTH</b>	<b>%0</b>	<b>96.86</b>	96.43	96.43	95.77	96.64	94.75	95.55	94.09
	<b>%10</b>	93.87	<b>95.99</b>	95.19	95.19	95.48	94.31	94.97	94.46
	<b>%20</b>	86.51	<b>95.77</b>	90.16	94.82	91.91	94.46	92.93	94.09
	<b>%30</b>	77.18	<b>95.41</b>	82.28	94.02	83.67	94.17	84.76	93.80
	<b>%40</b>	64.07	<b>95.55</b>	66.19	94.97	68.59	94.31	72.16	93.51
<b>HTRU</b>	<b>%0</b>	<b>97.18</b>	97.04	97.17	96.96	97.14	96.9	97.11	96.82
	<b>%10</b>	94.03	<b>97.01</b>	95.85	96.88	96.63	96.84	96.86	96.82
	<b>%20</b>	86.71	<b>97.01</b>	91.06	96.89	93.36	96.83	94.68	96.74
	<b>%30</b>	76.06	<b>97.00</b>	80.99	96.85	84.46	96.79	86.65	96.73
	<b>%40</b>	63.26	<b>96.95</b>	66.49	96.86	69.37	96.78	71.36	96.71

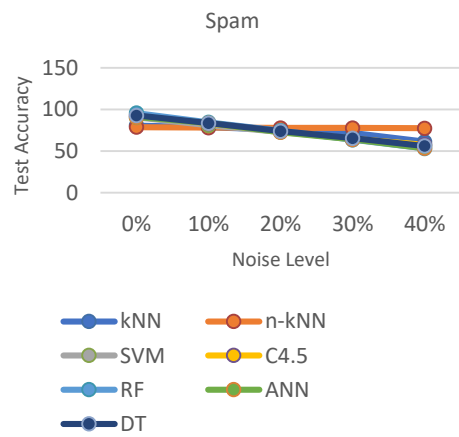
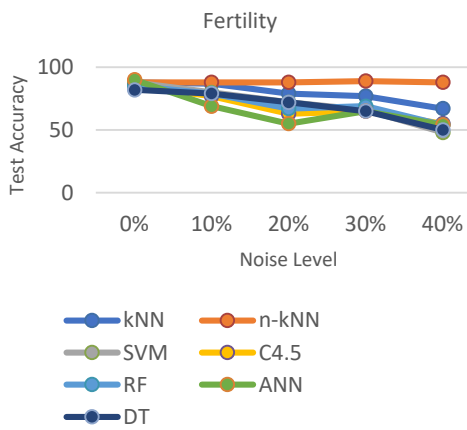
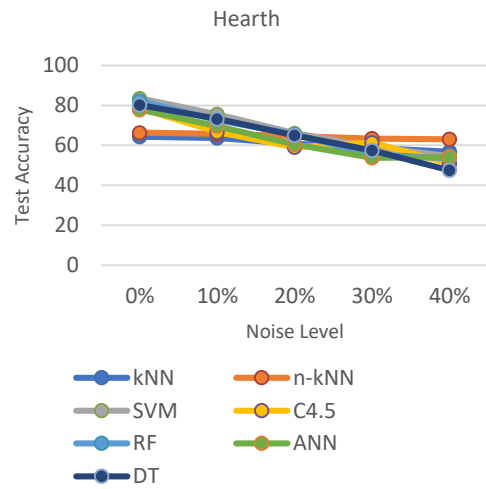
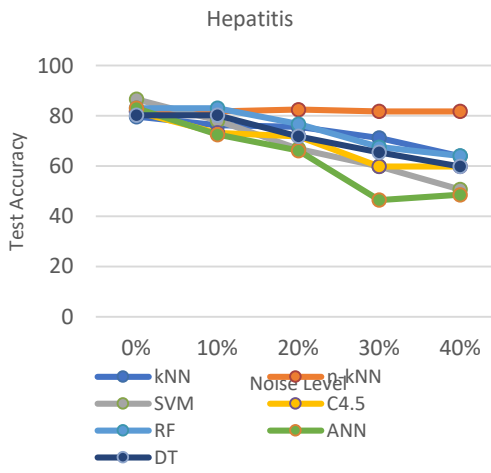
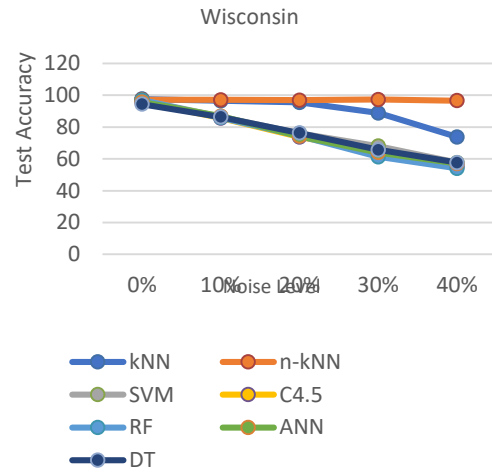
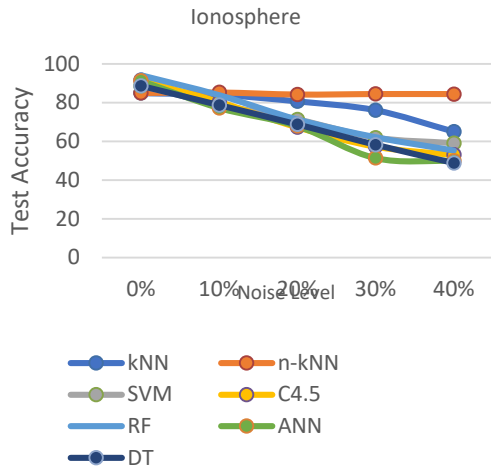
Table 3 Test accuracy results for benchmarking

DATA SET	NOISE LEVEL%	BEST classical kNN	BEST n-KNN	SVM	C4.5	RF	ANN	DT
<b>ION</b>	<b>%0</b>	84.90	84.90	88.60	91.45	<b>94.02</b>	91.17	88.60
	<b>%10</b>	84.04	<b>85.19</b>	78.92	80.34	83.76	77.21	78.63
	<b>%20</b>	80.64	<b>84.06</b>	71.23	67.24	71.23	67.81	68.66
	<b>%30</b>	76.05	<b>84.34</b>	61.82	57.27	62.11	51.28	58.12
	<b>%40</b>	64.94	<b>84.33</b>	58.97	52.99	55.27	49.86	48.72
<b>WIS</b>	<b>%0</b>	<b>97.65</b>	97.21	96.93	96.05	96.93	95.75	94.44
	<b>%10</b>	96.49	<b>97.07</b>	86.53	85.65	85.94	86.82	86.38
	<b>%20</b>	95.46	<b>96.92</b>	75.99	73.79	74.52	74.23	76.28

	<b>%30</b>	88.86	<b>97.36</b>	68.08	64.28	61.35	63.98	65.74
	<b>%40</b>	73.78	<b>96.63</b>	57.69	55.20	54.03	57.10	57.69
<b>HEPATİTİS</b>	<b>%0</b>	79.67	81.81	<b>86.62</b>	81.69	83.10	83.10	80.28
	<b>%10</b>	76.29	81.81	78.17	73.24	<b>83.10</b>	72.54	80.28
	<b>%20</b>	75.62	<b>82.52</b>	66.90	71.83	76.76	66.20	71.83
	<b>%30</b>	71.19	<b>81.81</b>	59.86	59.86	67.61	46.48	65.49
	<b>%40</b>	63.95	<b>81.81</b>	50.7	59.86	64.09	48.59	59.86
<b>HEART</b>	<b>%0</b>	64.34	66.32	<b>83.5</b>	78.55	82.18	77.89	80.2
	<b>%10</b>	63.69	65.68	<b>75.58</b>	66.67	72.61	69.64	73.27
	<b>%20</b>	60.76	64.70	<b>66.01</b>	59.08	65.35	60.40	65.02
	<b>%30</b>	58.76	<b>63.39</b>	59.41	61.39	55.45	53.80	57.43
	<b>%40</b>	56.80	<b>63.09</b>	54.46	50.83	53.47	54.13	47.53
<b>FERTILITY</b>	<b>%0</b>	88.00	88.00	88.00	85.00	85.00	<b>90.00</b>	82.00
	<b>%10</b>	87.00	<b>88.00</b>	80.00	77.00	79.00	69.00	79.00
	<b>%20</b>	79.00	<b>88.00</b>	72.00	63.00	67.00	55.00	72.00
	<b>%30</b>	77.00	<b>89.00</b>	66.00	65.00	69.00	65.00	65.00
	<b>%40</b>	67.00	<b>88.00</b>	48.00	55.00	54.00	54.00	50.00
<b>SPAM</b>	<b>%0</b>	80.61	78.79	90.44	92.98	<b>95.65</b>	90.98	92.76
	<b>%10</b>	78.59	78.27	80.90	83.13	<b>84.48</b>	83.55	83.48
	<b>%20</b>	75.61	<b>77.77</b>	72.70	73.79	74.61	72.53	73.88
	<b>%30</b>	71.77	<b>77.72</b>	64.68	64.68	63.55	64.01	65.44
	<b>%40</b>	61.25	<b>77.37</b>	57.01	57.57	53.05	53.29	56.03
<b>PUMA</b>	<b>%0</b>	61.29	64.50	65.21	86.21	<b>88.26</b>	84.45	87.50
	<b>%10</b>	59.02	64.33	62.18	79.08	<b>80.44</b>	73.17	80.30
	<b>%20</b>	57.68	64.48	58.13	<b>72.30</b>	71.70	62.67	72.29
	<b>%30</b>	55.60	64.32	52.65	<b>64.33</b>	61.77	52.43	64.22
	<b>%40</b>	52.31	<b>63.87</b>	49.84	56.8	53.69	49.51	55.19
<b>PIMA</b>	<b>%0</b>	64.99	64.99	<b>77.34</b>	73.83	75.52	75.13	75.52
	<b>%10</b>	61.50	64.99	70.31	<b>68.49</b>	67.06	<b>68.49</b>	67.45



	<b>%20</b>	64.53	<b>64.99</b>	64.71	62.76	62.11	61.46	60.42
	<b>%30</b>	62.26	<b>64.99</b>	54.82	52.6	52.73	53.78	55.08
	<b>%40</b>	58.00	<b>64.99</b>	52.73	52.99	43.49	49.22	51.04
<b>DATA BANKNOTE AUTH</b>	<b>%0</b>	96.86	96.43	98.03	98.54	99.34	<b>99.93</b>	98.47
	<b>%10</b>	95.48	<b>95.99</b>	88.78	88.12	88.85	89.87	88.19
	<b>%20</b>	92.93	<b>95.77</b>	78.94	76.53	75.58	79.37	77.7
	<b>%30</b>	84.76	<b>95.41</b>	68.95	66.69	62.83	68.22	67.93
	<b>%40</b>	72.16	<b>95.55</b>	59.84	57.14	53.57	57.51	59.69
<b>HTRU</b>	<b>%0</b>	97.18	97.04	97.56	97.84	<b>98.02</b>	97.97	97.8
	<b>%10</b>	96.86	<b>97.01</b>	86.98	88.18	88.2	88.28	88.17
	<b>%20</b>	94.68	<b>97.01</b>	76.99	78.51	78.28	78.6	78.61
	<b>%30</b>	86.65	<b>97.00</b>	67.77	69.16	67.89	69.25	69.08
	<b>%40</b>	71.36	<b>96.95</b>	58.75	59.58	56.5	59.54	59.31



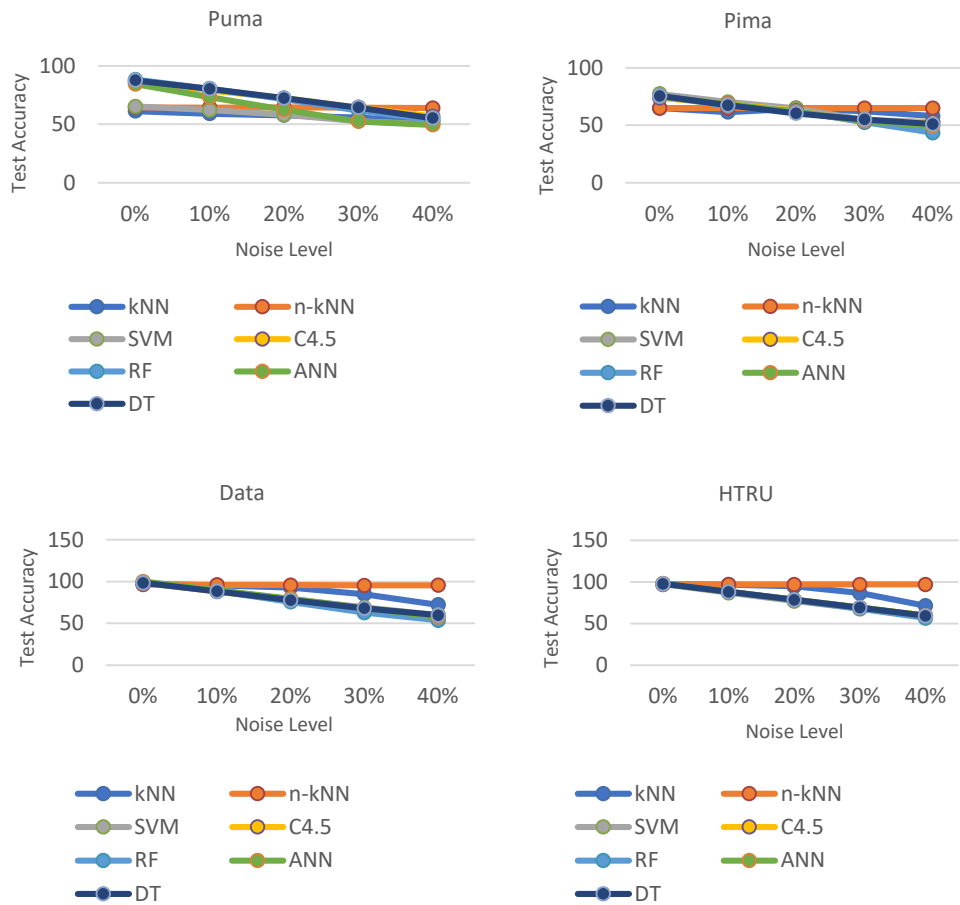


Figure 3. Noise level relationships between the n-kNN and other algorithms on different datasets

kNN algorithm gives better results than the kNN algorithm. Also, as can be seen in the Table 2 that the accuracy values decreases rapidly in classical kNN as the noise level increases. In the n-kNN algorithm, we can see a smaller decrease than classical kNN. Moreover, it can be seen that better accuracy results are obtained with n-KNN algorithm for many datasets. We can make some inferences according to the results obtained from Table 2. Firstly, the proposed kNN algorithm is more resistant to noise level than classical kNN because as noise level increases accuracy results

decreases fastly for kNN algorithm but according to the results of n-kNN algorithm results decreases slowly than classical kNN. Secondly, better accuracy results are obtained by n-kNN comparing with classical kNN. So it can be said that our proposed algorithm has a better performance than classical kNN. Thirdly, for different k values, n-kNN has still the better results than kNN and also better results can change according to the value of k parameter's value. This means that our algorithm is independent from the parameter values.

We also conducted further experiments to benchmark the performance of the n-kNN algorithm with some of the best-known algorithms in the literature (Table 3). We apply most popular classification methods from the literature as SVM, C 4.5, RF, ANN, DT algorithms via Weka tool and take best k values in order to make the comparison. It can be seen that for most of the

noise levels and for various datasets, the n-kNN provided best accuracy results than other methods. Also, the results indicated that when the noise level increased, the test accuracy values did not change much in the n-kNN algorithm while decreased evidently in the other algorithms tested. This is also can be seen from Figure 3.

We visualize our experiments with the graphs it can be seen in Figure 3. We compare our proposed algorithm and other algorithms for different noise levels and test accuracies. It can be obtained that n-kNN algorithm does not changes much as accuracy results when noise level increases. But other algorithms decreases dramatically when the noise level increases. According to this results, we can say that the proposed algorithm is more robust to noise data than the compared algorithms and it is more reliable for binary classification with noisy data.

### 5. CONCLUSION

This paper proposes an effective binary classification algorithm named n-kNN algorithm that is based on the well-known kNN algorithm. We suggest that our algorithm is resistant to noisy data and has reliable performance with high levels of noise data. To approve the performance we conduct numerical experiments by comparing the classical kNN and our new algorithm. We achieve significantly higher test accuracy values compared to the classical

kNN algorithm, with results reaching 84.33%, 93.63%, 81.81%, and 88.00% on real-world datasets. By these numerical results it is also confirmed that in noisy environments, the n-kNN algorithm can be a reliable classifier as the algorithm is not heavily affected by noise in the data. Furthermore, according to the experiments performed on real-world datasets, comparing proposed algorithm and several other methods in the literature based on the popular algorithms, most of the best accuracy results observed by our n-kNN algorithm reaching up to 85.19%, 97.07%, and 96.63%. Furthermore, it can be seen that in different values of the parameter k have better results and this means that the n-kNN algorithm is not dependent on the parameter k. Thus, it can be concluded that the n-kNN is a reliable algorithm because its accuracy results do not change much, stay stable in other words, with increasing noise levels. In this study, we apply our algorithm to binary classification data. However, in the future, we plan to modify and apply it to the data containing more than two classes. We aim to study the multi-class version of the n-kNN algorithm that can be applied more types of datasets. As a future work, we also will explore the data coming from the following popular areas and make some application on different fields cases.

## 6. Author contribution statements

In the scope of this study, the Author 1 in the formation of the idea, the design and the literature review, assessment of obtained results, supplying the materials used and examining the results; the spelling and checking the article in terms of content were contributed.

## 7. Ethics committee approval and conflict of interest statement

There is no conflict of interest with any person / institution in the article prepared. There is no need for an ethics committee approval in the prepared article.

## REFERENCES

- [1] Bootkrajang J. "A generalised label noise model for classification in the presence of annotation errors." *Neurocomputing*, 192, 61-71, 2016.
- [2] García LP, De Carvalho AC, Lorena AC. "Effect of label noise in the complexity of classification problems." *Neurocomputing*, 160, 108-119, 2015.
- [3] Sáez JA, Galar M, Luengo J, Herrera, F. "Tackling the problem of classification with noisy data using multiple classifier systems: analysis of the performance and robustness." *Information Sciences*, 247, 1-20, 2013.
- [4] Sáez, JA, Corchado, E. "ANCES: A novel method to repair attribute noise in classification problems." *Pattern Recognition*, 121, 108198, 2022
- [5] Zhu X, Wu X. "Class noise vs. attribute noise: A quantitative study." *Artificial Intelligence Review*, 22(3), 177-210, 2004.
- [6] Sluban B, Lavrač, N. "Relating ensemble diversity and performance: A study in class noise detection." *Neurocomputing*, 160, 120-131, 2015.
- [7] Luengo J, Sánchez-Tarragó D, Prati RC, Herrera F. "Multiple instance classification: Bag noise filtering for negative instance noise cleaning." *Information Sciences*, 579, 388-400, 2021.
- [8] García-Gil D, Luengo J, García S, Herrera F. "Enabling smart data: noise filtering in big data classification." *Information Sciences*, 479, 135-152, 2019.
- [9] Wang ZY, Luo XY, Liang J. "A Label Noise Robust Stacked Auto-Encoder Algorithm for Inaccurate Supervised Classification Problems." *Mathematical Problems in Engineering*, 2019.
- [10] Marsala C, Petturiti D. "Rank discrimination measures for enforcing monotonicity in decision tree induction." *Information Sciences*, 291, 143-171, 2015.
- [11] Zhu J, Liao S, Lei Z, Li S Z "Multi-label convolutional neural network based pedestrian attribute classification." *Image and Vision Computing*, 58, 224-229, 2017.
- [12] Chao L, Zhipeng J, Yuanjie Z. "A novel reconstructed training-set SVM with roulette cooperative coevolution for financial time series classification." *Expert Systems with Applications*, 123, 283-298, 2019.
- [13] Liao Y, Vemuri VR. "Use of k-nearest neighbor classifier for intrusion detection." *Computers & Security*, 21(5), 439-448, 2002.
- [14] García-Pedrajas N, Ortiz-Boyer D. "Boosting k-nearest neighbor classifier by means of input space projection." *Expert Systems with Applications*, 36(7), 10570-10582, 2009.
- [15] Wang ZY, Luo XY, Liang J. "A Label Noise Robust Stacked Auto-Encoder Algorithm for Inaccurate Supervised Classification Problems." *Mathematical Problems in Engineering*, 2019.
- [16] Triguero I, García-Gil D, Mailló J, Luengo J, García S, Herrera F. "Transforming big data into smart data: An insight on the use of the k-nearest neighbors algorithm to obtain quality data." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(2), e1289, 2019.
- [17] Mantas C J, Abellan J. "Analysis and extension of decision trees based on imprecise probabilities: Application on noisy data." *Expert Systems with Applications*, 41(5), 2514-2525, 2014a.
- [18] Alam MM, Gazuruddin M, Ahmed N, Motaleb A, Rana M, Shishir RR, Rahman RM., "Classification of deep-SAT images under label noise. Applied" *Artificial Intelligence*, 35(14), 1196-1218, 2021.
- [19] Mantas CJ, Abellan J. "Credal-C4.5 decision tree based on imprecise probabilities to classify noisy data." *Expert Systems with Applications*, 41(10), 4625-4637, 2014b.
- [20] Mantas, C. J., Abellan, J., & Castellano, J. G. "Analysis of Credal-C4.5 for classification in noisy domains." *Expert Systems with Applications*, 61, 314-326, 2016.
- [21] Mailló J, García S, Luengo J, Herrera, F, Triguero, I. "Fast and scalable approaches to accelerate the fuzzy k-Nearest neighbors classifier for big data." *IEEE Transactions on Fuzzy Systems*, 28(5), 874-886, 2019.
- [22] Dua D, Graff C. "UCI Machine Learning Repository" [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science. 2019.

- [23] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH, "The WEKA data mining software: an update". *ACM SIGKDD explorations newsletter*, 11(1), 10-18, 2009.
- [24] Shokrzade A, Ramezani M, Tab FA, Mohammad, MA. "A novel extreme learning machine based kNN classification method for dealing with big data." *Expert Systems with Applications*, 115293, 2021.
- [25] Liu CL, Lee CH, Lin PM. "A fall detection system using k-nearest neighbor classifier." *Expert systems with Applications*, 37(10), 7174-7181, 2010.
- [26] Catal C., "Software fault prediction: A literature review and current trends." *Expert Systems with Applications*, 38(4), 4626-4636, 2011
- [27] Yıldırım S, Yıldız T. "Türkçe için karşılaştırmalı metin sınıflandırma analizi" *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 24(5), 879-886, 2018
- [28] Sağlam A, Baykan NA. "Continuous time threshold selection for binary classification on polarized data" *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi* 25(5), 596-602, 2019
- [29] Goodfellow, I., Bengio, Y., & Courville, A. *Deep learning*. MIT Press.,2016
- [30] Bishop, C. M. *Pattern recognition and machine learning*. Springer, 2006
- [31] Mansour, R. F., Abdel-Khalek, S., Hilali-Jaghdam, I., Nebhen, J., Cho, W., & Joshi, G. P. An intelligent outlier detection with machine learning empowered big data analytics for mobile edge computing. *Cluster Computing*, 1-13. 2023.
- [32] Dash, C. S. K., Behera, A. K., Dehuri, S., & Ghosh, A. An outliers detection and elimination framework in classification task of data mining. *Decision Analytics Journal*, 6, 100164. 2023
- [33] Li, J., Zhang, J., Zhang, J., & Zhang, S., Quantum KNN classification with K Value selection and neighbor selection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2023