Journal of Soft Computing and Artificial Intelligence

ISSN: 2717-8226

Journal of Soft Computing and Artificial Intelligence

Journal homepage: https://dergipark.org.tr/en/pub/jscai

International Open Access

Volume 05
Issue 01

June, 2024

Journal of Soft Computing and Artificial Intelligence (JSCAI) is an international peer-reviewed journal that publishes integrated research articles in all areas of soft computing and artificial intelligence. The aim of the JSCAI journal is to provide a platform for researchers, professionals, and academicians around the world to combine and exchange new developments and their applications in various areas of soft computing and artificial intelligence. Journal of Soft Computing and Artificial Intelligence (JSCAI) is an international peer-reviewed journal that publishes integrated research articles in all areas of soft computing and artificial intelligence. The journal covers all branches of engineering, including mechanics, computer science, electronics, energy, aerospace engineering, materials science, nuclear engineering, systems analysis, alternative technologies, etc.

JSCAI publication, which is open access, is free of charge. There is no article submission and processing charges (APCs).

JSCAI is indexed & abstracted in:

Crossref (Doi beginning: 10.55195/jscai..xxxxxx)

Directory of Research Journals Indexing (DRJI)

Google Scholar

Index Copernicus (ICI Journal Master List)

OpenAIRE

Asos Index

Directory of Open Access scholarly Resources (ROAD)

Authors are responsible from the copyrights of the figures and the contents of the manuscripts, accuracy of the references, quotations and proposed ideas and the Publication Ethics (https://dergipark.org.tr/en/pub/jscai/policy)

Journal of Soft Computing and Artificial Intelligence (JSCAI) allows the author(s) to hold the copyright of own articles.

Table of Contents

Journal of Soft Computing and Artificial Intelligence

Journal homepage: https://dergipark.org.tr/en/pub/jscai

International Open Access

Volume 05
Issue 01

June, 2024

Research Article

Design of Monkeypox Disease Diagnosis Model Using Classical Machine Learning Algorithm

***Ahmed Muhammed Kalo Hamdan*[1*], *Dursun Ekmekci*[2]

[1,2] *Faculty of Engineering, Department of Computer Enginnering,78050, Karabuk, Türkiye*

ARTICLE INFO

Article history:
Received March 30, 2024
Revised May 6, 2024
Accepted May 31, 2024

Keywords:
Monkeypox,
Monkeypox Clinical Symptoms,
Classification,
Machine Learning

ABSTRACT

Monkeypox is a zoonotic viral disease that the World Health Organization (WHO) reported as an epidemic in 2022. In most nations, the rate of these illness infections started to rise over time. Monkeypox can be caught directly from an infected person or via animal contact. In this study, an artificial intelligence-based diagnostic model for early monkeypox infection detection is developed. The proposed method is based on building a model based on K-Nearest Neighbors, Support Vector Classification, Random Forest, Naive Bayes and Gradient Boosting for the classification problem. A voting method was also used to determine the final diagnosis of the proposed model. The system was trained and evaluated using a dataset that represented the clinical signs of monkeypox infection. The dataset comprises one hundred twenty infected patients and 120 typical cases out of 240 probable cases. The suggested model attained 75% accuracy.

1. Introduction

The monkeypox virus (MPXV) spread in 2022, alarming the public and raising experts' concerns due to its quick spread [1]. Between 1 to 11% of cases result in mortality [2]. According to the World Health Organization (WHO), more than 318,000 persons contracted this virus in August 2022, a considerable increase in the number of affected individuals [3]. Figure. 1. according to the World Health Organization, shows the number of monkeypox infections in each country (May 26, 2022). This virus is comparable to zoonotic smallpox and is a member of the corticovirus genus [4]. It affects humans and is brought on by the orthopoxvirus, a hazardous member of the poxviridae family [5].

For the first time, the illness was discovered in Africa, more specifically in the Republic of the Congo [4]. After that, it expanded throughout the world's nations. More than 1,256 cases of monkeypox had been documented as of June 2022 in various parts of Spain. The majority of cases were male. The average age was approximately 36 years at the time [6]. Direct contact with an infected person, animal, or object is the only way to contract monkeypox. It can also spread via mucous from the mouth, nose, or eyes [7]. Transmission of the monkeypox is focused, though not solely, during sex [5].

Smallpox and monkeypox have fairly similar clinical presentations, as shown in Figure. 2. however, the symptoms that develop after infection vary from case to case. Yet, in addition to anogenital lesions, sluggishness, and muscle pain, skin rash is the most typical indicator of infection [7]. Symptoms of monkeypox can last up to four weeks. Also, children are the most at risk [8]. The illness's adverse effects can include bronchiolitis, hypothermia, bacterial infections, and respiratory failure in patients [9]. The illness is challenging to diagnose based on a variety of clinical symptoms.

[1] *Corresponding author
e-mail: 2028126015@ogrenci.karabuk.edu.tr

DOI: 10.55195/jscai.1461849

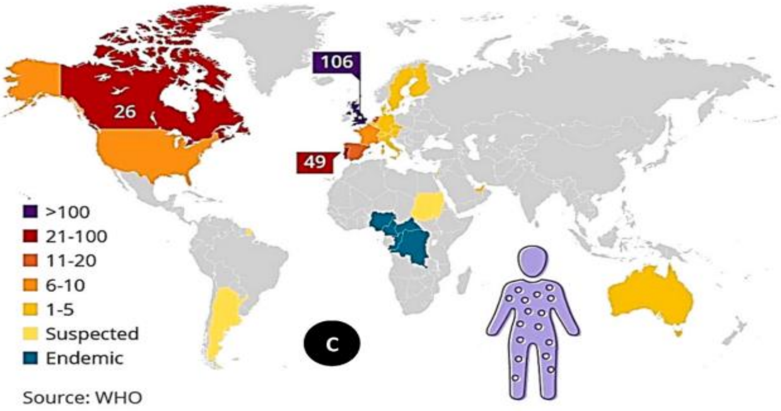


Figure 1 Number of monkeypox infections in each country [10]

It can take a while to learn the findings of laboratory tests before making an appropriate diagnosis of monkeypox since it must be distinguished from other diseases using a molecular test [10].[11].

Researchers have resorted to using artificial intelligence in medical and biological applications to diagnose illness problems as a result of the proliferation of AI applications [12]. Depending on the dataset gathered from the photographs of the lesions or the clinical signs of the infected, they utilised it in a variety of ways. [2] employ several techniques for gathering data. Skin lesion photographs make up the data. gathered manually and by contact with sick people. The study's main objective was to distinguish monkeypox from cases of related smallpox of other forms. Using VGG16 Deep Transfer Learning was the method used. It uses a neural network after three layers of convolutional filters to extract information from the photos. It had the ideal idea to employ transfer learning. The results' degree of accuracy was 86%.

The research conducted by [13] was split into three distinct investigations. These were all carried out using the suggested methodology. Generalization and Regularization are used in Multiclass Classification by Transfer Learning Approaches (GRA-TLA). Skin lesion photos make up the training dataset. It was created to aid the hospital in making decisions. According to computational findings, the first and second investigations could discriminate between those who had monkeypox and those who hadn't with an accuracy of 77% to 88% using the suggested method. In the third research, the residual network (ResNet), with accuracy ranging from 84% to 99%, performed the best for multiclass classification.

In this paper, we present a predictive model to enable the early detection of monkeypox. The proposed model is designed based on the combination of Gradient Boosting, SVC, KNN, Naïve Bayes and Random Forest algorithms. In addition to a voting system to determine the output of predictive filters. The method's performance was tested on clinical data from the bmj centre in London and compared with machine learning algorithms. The results provide that the proposed method could be used for early diagnosis of monkeypox disease.

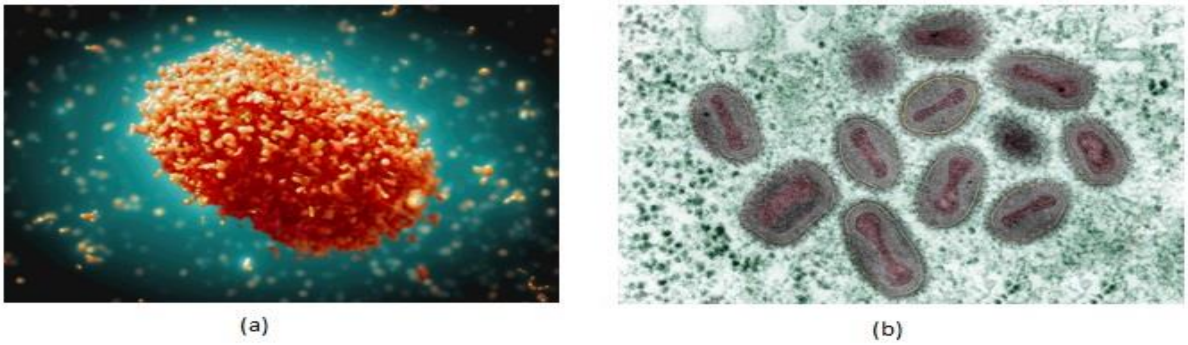


Figure 2 Difference between (a)- monkeypox virus and (b)- smallpox virus [10]

The structure of this paper is organized as follows: Section 2 theoretically describes the applied algorithms and the mechanism of creating the Model. Section 3 presents Experimental Study. Findings and discussions are presented in Section 4. Section 5 reports on the conclusion.

2. Methodology

This section describes a proposed approach for the early detection of monkeypox patients. Attempts to quickly and accurately diagnose patients with monkeypox. This approach consists of five classifiers and a voting system, as shown in Figure. 3. Thus, the detection model of monkeypox will be recognized by dataset based on the clinical symptoms that the patient presents with during the infection period to give the desired results. The proposed model was built using five techniques: Naive Bayes, KNN, SVC, Random Forest, and Gradient Boosting, combined with a voting method to provide the best diagnostic results. Following sections, the mechanism of action of each classification algorithm will be explained separately, as well as how to diagnose the disease based on the voting system in detail.

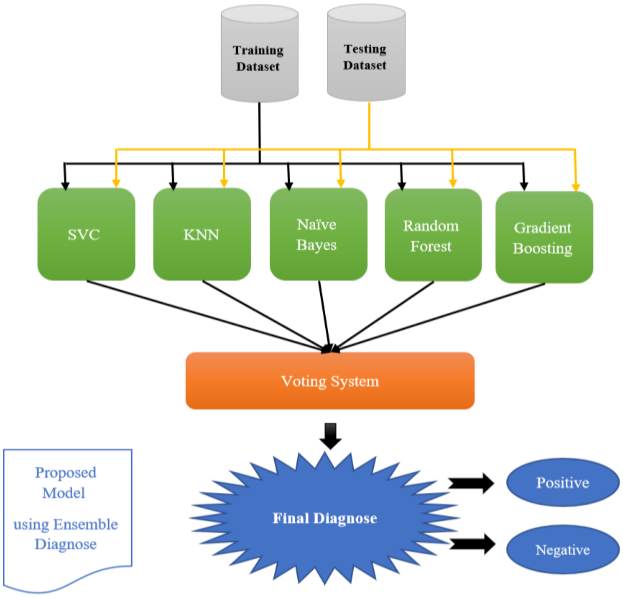


Figure 3 Proposed Diagnose Model

2.1. *Naïve Bayes (NB) Algorithm*

In this section, the working mechanism of the NB algorithm is explained. In fact, NB is one of the traditional machine learning algorithms (Figure. 4.). It is characterized by simplicity, and it has many uses in recognizing images and shapes, and it can also handle real-time problems, and it also gives high efficiency in early detection of diseases in medical systems [14].

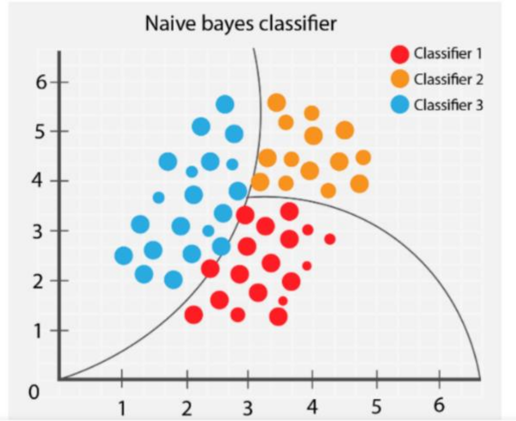


Figure 4 Naïve Bayes Classifier

This algorithm can handle large and small dataset. And it can give quick diagnosis instead of other models. It is also distinguished that it is not affected by noise in the data sets, that is, it is less sensitive to missing data. Bayes' theorem states that the probability of a hypothesis is determined after prior knowledge has been determined. This theory is related to conditional probabilities. As Eq. 1. given below states:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \qquad (1)$$

In fact, although NB is an effective algorithm for real-time applications such as medical diagnostics, one of its disadvantages is that all features are considered equal and independent during training.

2.2. *K-Nearest Neighbors (KNN) Algorithm*

KNN is one of the most popular and easy machine learning algorithms. It is used in many applications such as controlling electrical circuits and driving loads. In its learning, this algorithm depends on the value of K, which determines the number of samples that affect the classification process. In fact, there is no fixed value for the K factor, but it is set experimentally. This depends on the nature of the problem to be solved, as it differs from one unit to another. As shown in Figure. 5.

In this paper, KNN is introduced as a diagnostic model for monkeypox disease. It is based on the optimal value of K to obtain the best possible accuracy.

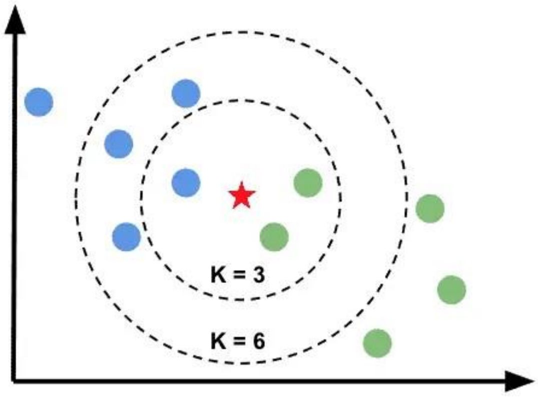


Figure 5 KNN algorithm

The KNN algorithm is based on the Cartesian distance function to measure the distance between samples [14]. The work of this algorithm can be summarized in three steps: In step 1, the distance between each testing sample and each training samples is calculated using the Euclidean distance, mentioned in Eq. 2.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \qquad (2)$$

In step 2, the value of k is set to give the minimum distance between the test sample and the training samples. In step 3, the final test sample is classified depending on the k-neighbor’s diagnoses.

2.3. Support Vector Classification (SVC) Algorithm

SVC falls under traditional machine learning algorithms. It is easy and simple. Data points are classified into two categories. This algorithm is based on tracing the line between the two data types. Always strives to find the best positioning of the dividing line between the two categories [15].

This algorithm can solve classification and regression problems, and it works with both linear and nonlinear arithmetic operations. Therefore, this algorithm can be applied to Classification of images and text categorization, in addition to its use in medical engineering. As shown in Figure. 6.

The logic that the algorithm will use to separate points into different groups and categorize them can be set to one of the following techniques: - Linear: The line separating the two sets of data is a straight line. - Polynomial: This pattern allows the separation of data using polynomials of degrees, which contributes to increasing classification accuracy when using complex data. - RBF: It is an acronym for Gaussian Radial Basis.

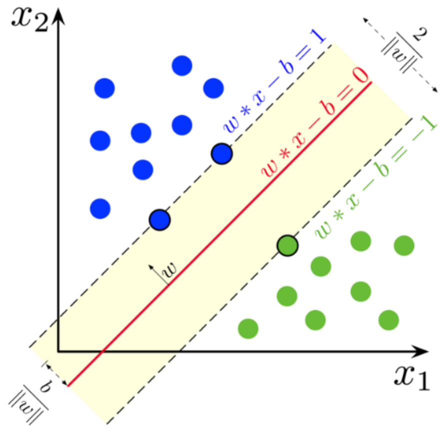


Figure 6 SVC algorithm

This principle works to create Gaussian distributions for each set of data in a way that determines the best distribution for the data. - sigmoid: uses the logic of logistic regression [16].

2.4. Random Forest Algorithm

Random Forest is one of the most used algorithms in machine learning and data science. This algorithm is considered moderated and is widely used in classification problems. One of the advantages of this algorithm is its ability to deal with continuous data, whether in regression problems or classification problems [17].

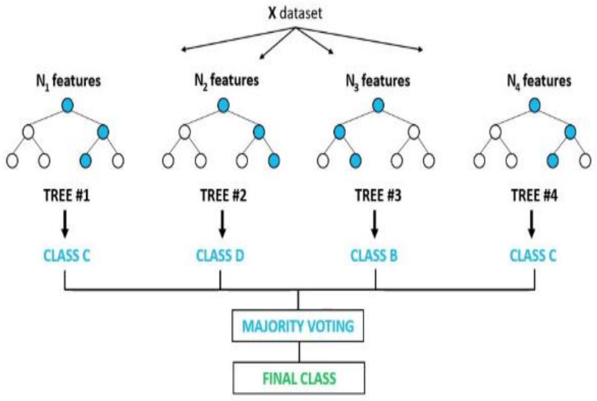


Figure 7 Random Forest algorithm

It can reach high accuracy during data training. This algorithm falls under the name of Ensemble Method. You create a subset of the training data, and the final output depends on the majority vote. The Random Forest model is a set of decision trees that have been used in machine learning problems. All internal nodes and branches in decision trees are associated with the test result. The internal nodes are the result of a specific feature test, while the branch represents the overall test result, and the leaf nodes are the expected result [18]. Its working mechanism consists of the following stages:

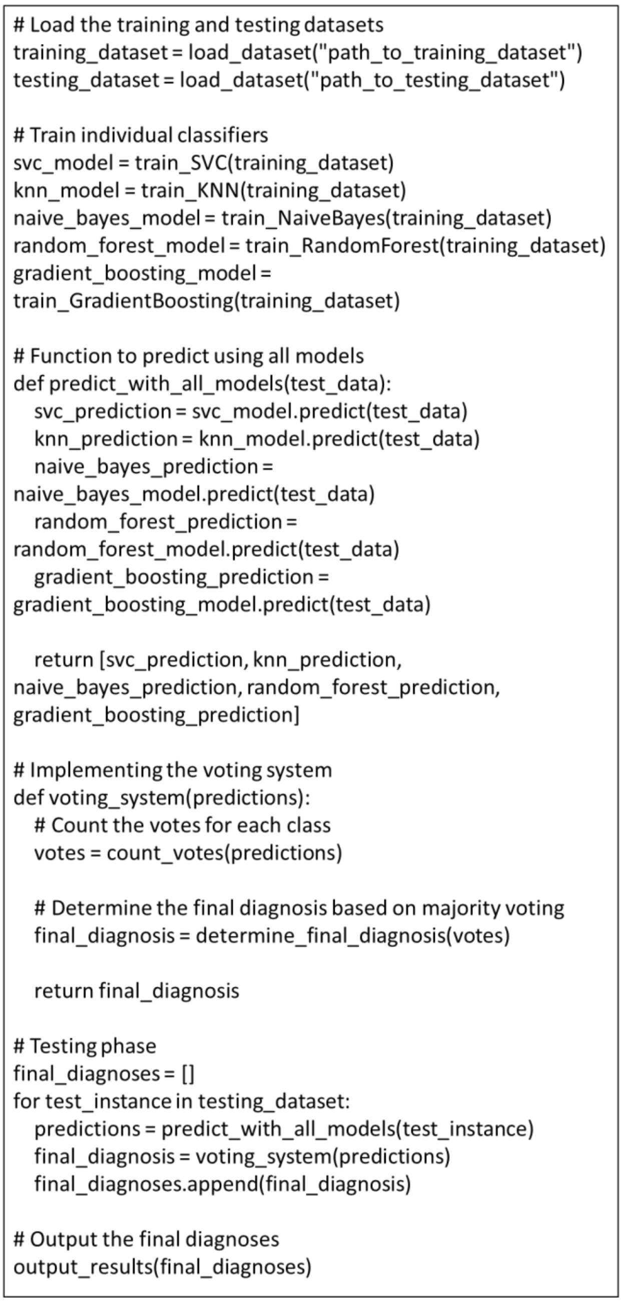
Stage 1: A subset of data points and a subset of features are selected to create each decision tree. Stage 2: Individual decision trees are generated for each sample. Stage 3: Each decision tree will generate an output. Stage 4: The final output is considered on the basis of majority vote or average rating and regression, respectively (Figure 7).

2.5. Gradient Boosting Algorithm

Gradient Boost is one of the most powerful algorithms in machine learning. It is well known that errors in machine learning are divided into bias errors and variance errors [19]. Therefore, the Gradient Boost algorithm is used to reduce the bias error in the model. Unlike the Adaboosting algorithm, we cannot mention the base estimator in the Gradient Boost algorithm [20]. This algorithm can handle both continuous data and categorical data. When this algorithm is used in regression problems, the error function is MSE. While Log Loss is in classification problems. Elements on which the algorithm depends: (a)- Loss Function: In this case, it aims to maximize the loss, which is not fixed and changes according to the nature of the problems. (b)- Weak Learners: These are mainly used for predictions. Decision tree is an example of poor learners. For the required real output values of the divisions, specific regression trees are applied. (c)-Additive Model: Adding more trees at a time. Thus, reducing losses at each addition.

3. Voting Method

In this section, the output of the five classifiers is voted on, as a way to determine the final output of model. The method of voting depends on giving all workbooks the same weight, which is the value 1. Classification cases are collected into two ‘Normal’ or ‘infection’ cases. Then he decides the outcome of the vote according to the largest number determined by the classifiers. That is, based on the majority vote, if the category “Normal” gets three votes and the category “infection” gets two votes, then vote will decide “Normal”. According to this mechanism, the voting system works.



```
# Load the training and testing datasets
training_dataset = load_dataset("path_to_training_dataset")
testing_dataset = load_dataset("path_to_testing_dataset")

# Train individual classifiers
svc_model = train_SVC(training_dataset)
knn_model = train_KNN(training_dataset)
naive_bayes_model = train_NaiveBayes(training_dataset)
random_forest_model = train_RandomForest(training_dataset)
gradient_boosting_model =
train_GradientBoosting(training_dataset)

# Function to predict using all models
def predict_with_all_models(test_data):
    svc_prediction = svc_model.predict(test_data)
    knn_prediction = knn_model.predict(test_data)
    naive_bayes_prediction =
naive_bayes_model.predict(test_data)
    random_forest_prediction =
random_forest_model.predict(test_data)
    gradient_boosting_prediction =
gradient_boosting_model.predict(test_data)

    return [svc_prediction, knn_prediction,
naive_bayes_prediction, random_forest_prediction,
gradient_boosting_prediction]

# Implementing the voting system
def voting_system(predictions):
    # Count the votes for each class
    votes = count_votes(predictions)

    # Determine the final diagnosis based on majority voting
    final_diagnosis = determine_final_diagnosis(votes)

    return final_diagnosis

# Testing phase
final_diagnoses = []
for test_instance in testing_dataset:
    predictions = predict_with_all_models(test_instance)
    final_diagnosis = voting_system(predictions)
    final_diagnoses.append(final_diagnosis)

# Output the final diagnoses
output_results(final_diagnoses)
```

Figure 8 Pseudo Code of Proposal Model.

Figure. 8. shows us the pseudocode outlines a method for building an ensemble learning model to improve diagnostic accuracy. First, it loads the training and testing datasets. Then, it trains five different machine learning models: Support Vector Classifier (SVC), K-Nearest Neighbors (KNN), Naive Bayes, Random Forest, and Gradient Boosting. Each model makes its own predictions for the test data. These predictions are combined using a voting system that decides the final diagnosis based on the majority vote. During testing, each data point is evaluated by this ensemble of models, and the final diagnoses are recorded. This approach combines the strengths of various classifiers, making the diagnostic process more accurate and reliable.

4. Experimental Study

This section gives an overview of the training dataset's processing as well as a summary of some of its features. Together with the suggested model's performance metrics, which have equations to determine each value.

4.1. Dataset

The database provided for the study was obtained from Kaggle [21]. This dataset was based on patients with monkeypox, and other suspected cases. This data is published according to thebmj center in London. This data contains 240 diagnosed cases with 11 features which are described in Table 1.

Table 1 Detail of Dataset

SN	*Attribute*	*Type*	*Value*
1	Patient_ID	Numerical	[1 - 240]
2	Systemic Illness	Nominal	Fever, None, Swollen Lymph Nodes, Muscle Aches and Pain
3	Rectal Pain	Nominal	True, False
4	Sore Throat	Nominal	True, False
5	Penile Oedema	Nominal	True, False
6	Oral Lesions	Nominal	True, False
7	Solitary Lesion	Nominal	True, False
8	Swollen Tonsils	Nominal	True, False
9	HIV Infection	Nominal	True, False
10	Sexually Transmitted Infection	Nominal	True, False
11	MonkeyPox	Nominal	Positive, Negative

A retrospective observational study was performed on individuals with polymerase chain reaction (PCR) confirmed monkeypox virus, who were evaluated and managed at a south London HCID center. This center, one of five HCID centers in the UK, serves the population of inner city central and south London. Diagnostic swabs were taken from lesions at affiliated community sexual health and HIV medicine services, upon hospital admission (either in inpatient wards or emergency departments), or upon transfer of patients suspected of having monkeypox from neighboring NHS trusts. These samples were processed at the Rare and Imported Pathogens Laboratory in Porton Down, UK.21 Individuals suspected or confirmed to have monkeypox were stratified by risk according to disease severity, immune status, and their ability to self-isolate, and were managed accordingly. As part of standard clinical care, individuals were clinically assessed prior to testing. All individuals with a positive PCR test result for monkeypox virus participated in a telephone consultation to receive counseling about their result and to undergo a risk assessment.

This dataset contains newly infected patients who show symptoms of monkeypox. From the 240 data set we have 120 cases of monkeypox, and 120 healthy cases. Patients with monkeypox are considered positive cases, whilst healthy individuals are considered negative instances. A negative case does not always imply that the person is healthy and free from monkeypox. But, based on this information, we can tell if he merely had monkeypox. There are 11 characteristics in this dataset, including the patient's clinical symptoms like fever and inflammation. These characteristics are used to describe the symptoms that a patient experiences in order to convey the patient's condition.

We computed the linear correlation coefficient between the features and observed varying degrees of correlation among the data. Additionally, the dataset contains no null values. We encoded the binary variables as follows: True as 1 and False as 0. The dataset was then divided into two subsets, with 80% allocated for training and 20% for testing. Overall, the proportions of infected and non-infected cases were evenly distributed across these subsets.

In actuality, the dataset for monkeypox was split into two sets of data: 168 cases for the training set and 72 cases for the test set. There are 120 positive instances and 120 negative cases total. Initial instances often contain noisy and missing values. Therefore, it is necessary to pre-process the raw data to achieve good results. All data set used has been verified. In (Systemic Illness) we converted the categorical attribute to numeric. The dataset does not have any missing values. Furthermore, we performed correlation analysis on these datasets, when two attributes are closely related, one of them needs to be omitted to achieve better results.

4.2. Performance Evaluation

Algorithms for classifying data are measured for accuracy using statistical techniques. These techniques help establish the accuracy, precision, F1-score, and sensitivity standards for the used algorithm (They are shown in Table. 2.). If the individuals have been correctly categorized, monkeypox in our dataset can be classed as True Positive or True Negative. If misdiagnosed, it may be labelled as a False Positive or False Negative. Figure. 9. illustrates these properties. As a result, the following estimated values are provided:

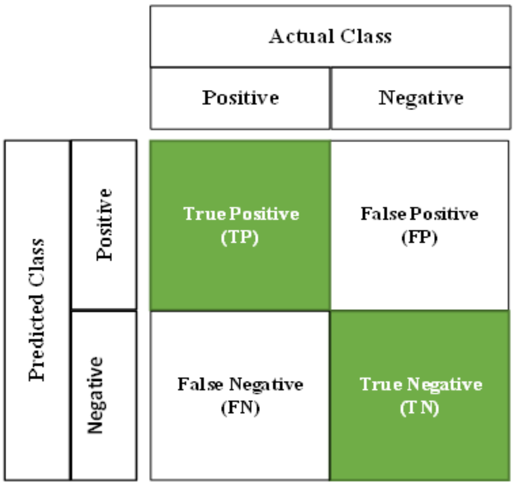


Figure 9 Confusion Matrix

- *True Positive (TP):* It predicts positive values when its true values are positive.
- *True Negative (TN):* It predicts negative values when its true values are negative.
- *False Positive (FP):* It predicts positive values when its true values are negative.
- *False Negative (FN):* It predicts negative values when its true values are positive.

Tools by which algorithm accuracy is measured depending on Confusion Matrix are:

Table 2 Statistical methods for measuring the accuracy of a machine learning model.

Method Name	*Equation*
Accuracy	$\frac{T_p + F_p}{T_p + T_n + F_p + F_n}$
Precision	$\frac{T_p}{T_p + F_p}$
Sensitivity	$\frac{T_p}{T_p + F_n}$
F1-score	$2 * \frac{Recall * Precision}{Recall + Precision}$

4.2.1. Accuracy

This way is used to describe the performance of a classifier based on the correctly predicted states versus the overall states. As in Eq. 3.

$$Accuracy = \frac{T_p + F_p}{T_p + T_n + F_p + F_n} \tag{3}$$

This measure is not considered sufficient to be considered the best model, if the data set is not balanced.

4.2.2. Precision

It determines the ratio between actual positive values and all projections that are positive. When the model assumes more false positives, the accuracy value drops. As in Eq. 4.

$$Precision = \frac{T_p}{T_p + F_p} \tag{4}$$

4.2.3. Sensitivity

The percentage of positive diagnostics that were diagnosed as positive. As in Eq. 5.

$$Sensitivity = \frac{T_p}{T_p + F_n} \tag{5}$$

4.2.4. F1-score

The F1-Score runs from 0 to 1, and it is a harmonic mean of precision and recall. Low false negative and false positive readings produce this metric's higher value. As in Eq. 6.

$$F1 - score = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{6}$$

5. Result and Discussion

A sample dataset of 72 values from the dataset, which is a combination of monkeypox cases gathered at thebmj center in London, is used to evaluate the proposed model.

The only instances of monkeypox with a variety of symptoms were included in the sample data. This also addresses how many positive and negative instances there are with various symptoms, and therefore it suggests a five-classifier model diagnostic approach: Gradient boosting, SVC, KNN, Naïve Bayes and Random Forest.

Table 3 Performance of proposed model

phase	*Performance*			
	Accuracy	*F1-Score*	*Precision*	*Sensitivity*
Training	87 %	87 %	87 %	88 %
Testing	75 %	75 %	69 %	81 %

Table. 3. presents the results of the proposed model during training and testing. The performance of the model was measured by several criteria. They appear as follows: in the training period accuracy, F1-score, precision and sensitivity take the values 87%, 87%, 87%, 88%. Respectively. When testing, accuracy, F1-score, precision, and sensitivity standards were taken as 75%, 75%, 69%, 81%. Respectively. Figure. 10. presents the confusion matrix of the proposed model during training and testing.

Table 4 Performance of RF, SVC, KNN, NB, GB

Algorithms	*Performance*			
	Accuracy	*F1-Score*	*Precision*	*Sensitivity*
RF	52 %	51 %	53 %	52 %
SVC	58 %	58 %	60 %	58 %
KNN	60 %	60 %	60 %	60 %
NB	61 %	60 %	62 %	61 %
GB	63 %	62 %	63 %	61 %

Table. 4. reveals that Gradient Boosting (GB) outperformed other algorithms across all metrics, demonstrating superior accuracy, F1-Score, precision, and sensitivity. Naive Bayes (NB) and K-Nearest Neighbors (KNN) also performed well, particularly in precision and sensitivity. Support Vector Classifier (SVC) showed moderate performance, while Random Forest (RF) had the lowest scores in all evaluated metrics. These results suggest that Gradient Boosting is the most effective model among those tested for this dataset, offering a balanced performance in both identifying and accurately predicting positive cases.

6. Conclusion

The study briefly overviews the origin of the zoonotic illness monkeypox, which is spread from animals to people. The very virulent Orthopoxvirus family includes this virus. Many people are alarmed by the disease's societal proliferation. As a result, society needs an automated system for early detection that aids in diagnosing this illness infection, should it arise. Early diagnosis can save lives by preventing complications for those with the condition. This work aims to develop a model for identifying monkeypox infection based on the clinical symptoms of the illness that manifest in the infected person. There are five machine-learning algorithms in the suggested model. A 75% accuracy rate was reached using the proposed model. As the current strategy to develop a diagnostic mechanism for monkeypox illness is backed by several published literature that uses an AI-based diagnostic model, we hope that this paper will assist future researchers and practitioners gain from the presented approach.

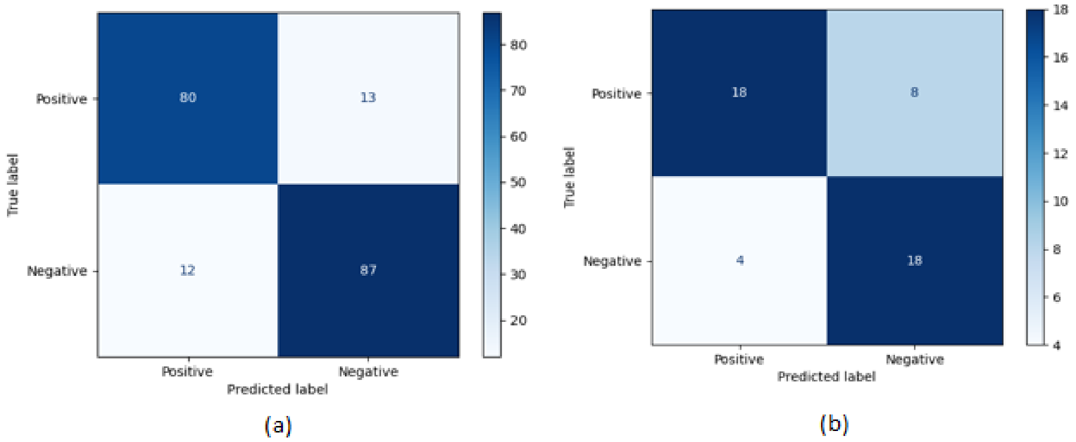


Figure 10 (a) Confusion matrix of training phase for the proposed model. (b) Confusion matrix for testing

The proposed model is substantiated by numerous scholarly publications that utilize AI-based diagnostic models. It is our aspiration that this article will aid future researchers and practitioners in leveraging the outlined approach to create a diagnostic framework for monkeypox disease. In subsequent research, we intend to develop an AI methodology capable of extracting features for monkeypox using real-time data and achieving higher classification accuracy.

References

[1] Singh, S., Chauhan, P., and Singh, N. J., "Capacity optimization of grid connected solar/fuel cell energy system using hybrid ABC-PSO algorithm", *International Journal Of Hydrogen Energy*, 45 (16): 10070–10088 (2020).

[2] Shahyeez Ahamed, B. S. H., Usha, R., and Sreenivasulu, G., "A Deep Learning-based Methodology for Predicting Monkey Pox from Skin Sores", (2022).

[3] Rimmer, S., Barnacle, J., Gibani, M. M., Wu, M. S., Dissanayake, O., Mehta, R., Herdman, T., Gilchrist, M., Muir, D., Ebrahimsa, U., Mora-Peris, B., Dosekun, O., Garvey, L., Peters, J., Davies, F., Cooke, G., and Abbara, A., "The clinical presentation of monkeypox: a retrospective case-control study of patients with possible or probable monkeypox in a West London cohort", *International Journal Of Infectious Diseases*, 126: 48–53 (2023).

[4] Yinka-Ogunleye, A., Aruna, O., Dalhat, M., Ogoina, D., McCollum, A., Disu, Y., Mamadu, I., Akinpelu, A., Ahmad, A., Burga, J., Ndoreraho, A., Nkunzimana, E., Manneh, L., Mohammed, A., Adeoye, O., Tom-Aba, D., Silenou, B., Ipadeola, O., Saleh, M., Adeyemo, A., Nwadiutor, I., Aworabhi, N., Uke, P., John, D., Wakama, P., Reynolds, M., Mauldin, M. R., Doty, J., Wilkins, K., Musa, J., Khalakdina, A., Adedeji, A., Mba, N., Ojo, O., Krause, G., Ihekweazu, C., Mandra, A., Davidson, W., Olson, V., Li, Y., Radford, K., Zhao, H., Townsend, M., Burgado, J., and Satheshkumar, P. S., "Outbreak of human monkeypox in Nigeria in 2017–18: a clinical and epidemiological report", *The Lancet Infectious Diseases*, 19 (8): 872–879 (2019).

[5] Rodríguez, B. S., Guzmán Herrador, B. R., Franco, A. D., Sánchez-Seco Fariñas, M. P., del Amo Valero, J., Aginagalde Llorente, A. H., Pérez de Agreda, J. P. A., Malonda, R. C., Castrillejo, D., Chirlaque López, M. D., Chong Chong, E. J., Balbuena, S. F., García, V. G., García-Cenoz, M., Hernández, L. G., Montalbán, E. G., Carril, F. G., Cortijo, T. G., Bueno, S. J., Sánchez, A. L., Linares Dópido, J. A., Lorusso, N., Martins, M. M., Martínez Ochoa, E. M., Mateo, A. M., Peña, J. M., Negredo Antón, A. I., Otero Barrós, M. T., del Carmen Pacheco Martinez, M., Jiménez, P. P., Pérez Martín, O. G., Rivas Pérez, A. I., García, M. S., Soria, F. S., Sierra Moros, M. J., Brandini Romersi, A. M., Lozano, C. G., Vallejo-Plaza, A., Campelli, G. S., Balader, P. S., San Miguel, L. G., Cano, E. A., Ruiz-Algueró, M., Simón, L., Arias, P., Vázquez, A., Sánchez, P., Herrero, L., Molero, F., Torres, M., Sánchez, L., Cejudo, C., Polo, R., Castellá, J. G., Koerting, A., Vazquez Rincon, I. M., Ugarriza, A. V., Remacha, C. M., Boone, A. L., Huerta, M. H., Riutort, A. N., Torres Lana, Á. L., Herrera, A. A., García, I. F., Guijarro, M. V., Duran, G. R., Marín, V. R., Saavedra, M. R. R., Arribas, S. F., Rodríguez, H. M., Calvo, N. R., Rio, V. A., Garzón, N. G., Martínez-Pino, I., Jesús Rodríguez Recio, M., Roig Sena, F. J., Pública, S., del Mar López-Tercero Torvisco, M., del Carmen García Bañobre, M., del Pilar Sánchez Castro, M., Martínez Soto, M. R., García, M. A., Martínez, F. M., Jose Domínguez Rodríguez, M., Morales, L. M., Navarro, A. H., Deorador, E. C., Forte, A. N., Julia, A. N., Largo, N. C., Ortíz, C. S., Marín, N. G., Díaz, J. S., Belen, M., Zarzuelo, R., Pariente, N. M., del Buey, J. F. B., Velasco Rodríguez, M. J., Peña, A. A., Baena, E. R., Benito, A. M., Meixeira, A. P., Martinez, J. I., Ordobas, M., Arce, A., Naranjo, A. S. M., Castilla, J., Casado, I., Burgui, C., Egües, N., Ezpeleta, G., Carollo, O. M., Larraitz, A., Arricibita, E. H., and Ibáñez Pérez, A. C., "Epidemiologic Features and Control Measures during Monkeypox Outbreak, Spain, June 2022", *Emerging Infectious Diseases*, 28 (9): 1847–1851 (2022).

[6] Kannan, S. R., Sachdev, S., Reddy, A. S., Kandasamy, S. L., Byrareddy, S. N., Lorson, C. L., and Singh, K., "Mutations in the monkeypox virus replication complex: Potential contributing factors to the 2022 outbreak", *Journal Of Autoimmunity*, 133: (2022).

[7] Thornhill, J. P., Barkati, S., Walmsley, S., Rockstroh, J., Antinori, A., Harrison, L. B., Palich, R., Nori, A., Reeves, I., Habibi, M. S., Apea, V., Boesecke, C., Vandekerckhove, L., Yakubovsky, M., Sendagorta, E., Blanco, J. L., Florence, E., Moschese, D., Maltez, F. M., Goorhuis, A., Pourcher, V., Migaud, P., Noe, S., Pintado, C., Maggi, F., Hansen, A.-B. E., Hoffmann, C., Lezama, J. I., Mussini, C., Cattelan, A., Makofane, K., Tan, D., Nozza, S., Nemeth, J., Klein, M. B., and Orkin, C. M., "Monkeypox Virus Infection in Humans across 16 Countries — April–June 2022", *New England Journal Of Medicine*, 387 (8): 679–691 (2022).

[8] Dwivedi, M., Tiwari, R. G., and Ujjwal, N., "Deep Learning Methods for Early Detection of Monkeypox Skin Lesion", (2023).

[9] Patel, A., Bilinska, J., Tam, J. C. H., Da Silva Fontoura, D., Mason, C. Y., Daunt, A., Snell, L. B., Murphy, J., Potter, J., Tuudah, C., Sundramoorthi, R., Abeywickrema, M., Pley, C., Naidu, V., Nebbia, G., Aarons, E., Botgros, A., Douthwaite, S. T., Van Nispen Tot Pannerden, C., Winslow, H., Brown, A., Chilton, D., and Nori, A., "Clinical features and novel presentations of human monkeypox in a central London centre during the 2022 outbreak: Descriptive case series", *The BMJ*, (2022).

[10] Irmak, M. C., Aydın, T., and Yağanoğlu, M., "Monkeypox Skin Lesion Detection with MobileNetV2 and VGGNet Models", (2022).

[11] de la Calle-Prieto, F., Estébanez Muñoz, M., Ramírez, G., Díaz-Menéndez, M., Velasco, M., Azkune Galparsoro, H., Salavert Lletí, M., Mata Forte, T., Blanco, J. L., Mora-Rillo, M., Arsuaga, M., de Miguel Buckley, R., Arribas, J. R., and Membrillo, F. J., .

[12] Matuszewski, D. J. and Sintorn, I. M., "TEM virus images: Benchmark dataset and deep learning classification", *Computer Methods And Programs In Biomedicine*, 209: (2021).

[13] Ahsan, M. M., Uddin, M. R., Ali, M. S., Islam, M. K., Farjana, M., Sakib, A. N., Momin, K. Al, and Luna, S. A., "Deep transfer learning approaches for Monkeypox disease diagnosis", *Expert Systems With Applications*, 216: (2023).

[14] Saleh, A. I. and Rabie, A. H., "Human monkeypox diagnose (HMD) strategy based on data mining and artificial intelligence techniques", *Computers In Biology And Medicine*, 152: (2023).

[15] Liu, T., Jin, L., Zhong, C., and Xue, F., "Study of thermal sensation prediction model based on support vector classification (SVC) algorithm with data preprocessing", *Journal Of Building Engineering*, 48: (2022).

[16] Loger, B., Dolgui, A., Lehuédé, F., and Massonnet, G., "Improving the Tractability of SVC-based Robust Optimization", (2022).

[17] Cao, M., Yin, D., Zhong, Y., Lv, Y., and Lu, L., "Detection of geochemical anomalies related to mineralization using the Random Forest model optimized by the Competitive Mechanism and Beetle Antennae Search", *Journal Of Geochemical Exploration*, 249: 107195 (2023).

[18] Gao, W., Xu, F., and Zhou, Z. H., "Towards convergence rate analysis of random forests for classification", *Artificial Intelligence*, 313: (2022).

[19] Nhat-Duc, H. and Van-Duc, T., "Comparison of histogram-based gradient boosting classification machine, random Forest, and deep convolutional neural network for pavement raveling severity classification", *Automation In Construction*, 148: 104767 (2023).

[20] Li, Y., Feng, Y., and Qian, Q., "FDPBoost: Federated differential privacy gradient boosting decision trees", *Journal Of Information Security And Applications*, 74: 103468 (2023).

[21] "Monkey-Pox PATIENTS Dataset. | Kaggle", https://www.kaggle.com/datasets/muhammad4hmed/monkeypox-patients-dataset (2023).

Journal of Soft Computing and Artificial Intelligence

Journal homepage: https://dergipark.org.tr/en/pub/jscai

International Open Access

Volume 05
Issue 01

June, 2024

Research Article

A New Fast Filter-based Unsupervised Feature Selection Algorithm Using Cumulative and Shannon Entropy

Samet Demirel[1][1], ***Fatih Aydın***[2]

[1]*Distance Education Application and Research Center. Balıkesir University, 10145, Balıkesir, Türkiye*
[2]*Department of Computer Engineering, Faculty of Engineering. Balıkesir University, 10145, Balıkesir, Türkiye*

ARTICLE INFO

Article history:
Received April 03, 2024
Revised May 05, 2024
Accepted May 27, 2024

Keywords:
Machine Learning
Unsupervised Feature Selection
Univariate-filter Approach
Cumulative Entropy
Shannon Entropy

ABSTRACT

The feature selection process is indispensable for the machine learning area to avoid the curse of dimensionality. Hereof, the feature selection techniques endeavor to handle this issue. Yet, the feature selection techniques hold several weaknesses: (i) the efficacy of the machine learning methods could be quite different on the chosen features (ii) by depending on the selected subset, substantial differences in the effectiveness of the machine learning algorithms could also be monitored (iii) the feature selection algorithms can consume much time on massive data. In this work, to address the issues above, we suggest a new and quick unsupervised feature selection procedure, which is based on a filter and univariate technique. The offered approach together regards both the Shannon entropy computed by the symmetry of the distribution and the cumulative entropy of the distribution. As a consequence of comparisons done with some cutting-edge feature selection strategies, the empirical results indicate that the presented algorithm solves these problems in a better way than other methods.

1. Introduction

Machine learning algorithms suffer from high-dimensional data sets. In this respect, the Feature Selection (FS) algorithms would be a supporting element for reconstructing the model fast and increasing its performance. FS is the task of determining features that allow preserving or, in some data sets, enhancing the model performance without needing the use of all original features [1]. FS is beneficial in learning tasks such as classification, regression, or clustering since as well as decreasing the storage and computing requirements, it affords to dismiss the curse of dimensionality [2] and allows to form of models that have better generalization ability [3]. Thus, the feature subset that best represents the original data set is selected. The selected features refer to information that affects the model outcome and cannot be provided by other features.

Figure **1** describes this process.

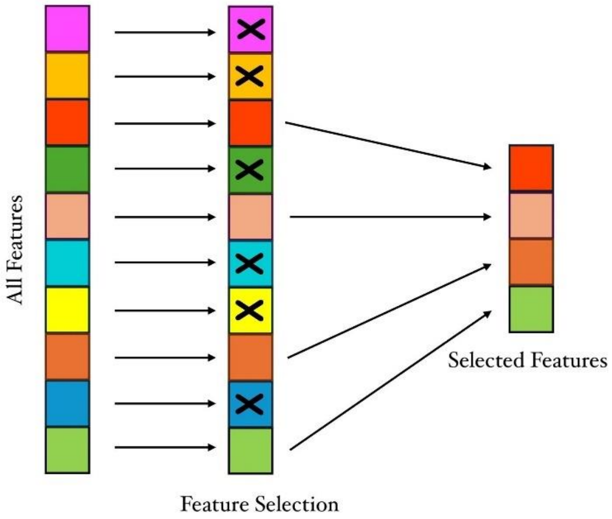


Figure 1 The feature selection scheme.

[1] * Corresponding author
e-mail: sametdemirel@balikesir.edu.tr
DOI: 10.55195/jscai.1464638

FS algorithms are divided into three as supervised, semi-supervised, and unsupervised, in terms of the use of class information. Unsupervised Feature Selection (UFS) algorithms have three significant supremacies: (i) they are unbiased, (ii) they can process data even when prior knowledge is unavailable, and (iii) they can decrease over-fitting in contrast to supervised ones [1]. FS algorithms are separated into four basic approaches: filter, wrapper, hybrid, and embedded, according to the selection strategy of features [4].

Figure 2 shows the categorization of the feature selection methods in terms of the use of class information and the selection strategies.

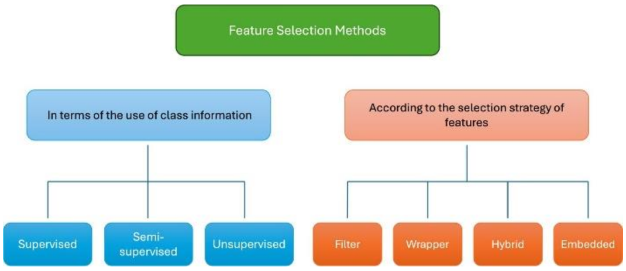

Feature Selection Methods
In terms of the use of class information
According to the selection strategy of features
Supervised
Semi-supervised
Unsupervised
Filter
Wrapper
Hybrid
Embedded

Figure 2 The categorization of the feature selection methods.

The filter approach focuses on the intrinsic and statistical properties of data sets. Hence, they are rather fast in comparison to the other approaches. The wrapper technique is based on the machine learning algorithm selected. Therefore, they are slower than the filter approach. The hybrid strategy incorporates the filter and wrapper approaches. Finally, the embedded methods simultaneously perform the related learning task and feature selection.

In the last decades, hundreds of remarkable UFS algorithms have been introduced. These unsupervised feature selection algorithms address troubles in the subfields such as big data, heterogeneous attributes, high-dimensional data sets, image processing, data clustering, categorical data sets, rule induction, text mining, and biomarker discovery. Besides, there exist various approaches employed to develop feature selection algorithms in the literature. These techniques are adaptive graph-based approach, adaptive similarity learning, autoencoder, bio-inspired approach, clustering, differential evolution, Dirichlet process, discriminative analysis, extreme learning machine, graph representation, Gravitational Search Algorithm, hidden Markov model, Hilbert-Schmidt independence criterion, integer programming, Kolmogorov-Smirnov test, k-nearest neighbors, Laplace score, latent representation, Local Sensitive Dual Concept Learning, local structure learning, Locality Preserving Projection, manifold learning, matrix factorization, Maximal Information Compression Index, metaheuristic algorithms, metric learning, mutual information, nonparametric Bayesian mixture model, particle swarm optimization, principal component analysis, regression-based approach, self-representation learning, sparse learning, spectral learning, statistical learning, subspace learning, and symmetrical uncertainty.

We categorize the unsupervised feature selection algorithms in the literature in terms of the techniques they have applied. Accordingly, in the context of neighborhood relationships, LS (Laplacian score for unsupervised feature selection) [5] uses the locality-preserving capability by finding the nearest neighbors of each feature and thereby selects features. RNE (Robust Neighborhood Embedding) characterizes the local geometry of the data by linear coefficients that rebuild each point via k-nearest neighbors to get the weight matrix and it solves the model based on the Taxicab-norm through the alternation direction method of multipliers [6]. According to clustering approaches, MCFS (Multi-Cluster Feature Selection) [7] conserves the multi-cluster structure of the data by solving a sparse eigenproblem and a least-squares problem and thus selects relevant features.

As for self-representation approaches, RSR (unsupervised feature selection method based on Regularized Self-Representation) [8] selects features by inducing low-rank representation in subspace clustering where any feature can be reproduced as the linear combination of other convenient features. DISR (feature selection method via Diversity-Induced Self-Representation) [9] selects features by reducing redundant features based on diversity and the internal self-representation characteristic of features. In respect of the use of information-theoretic approaches, IUFS (Information-theoretic Unsupervised Feature Selection) [10] aims to maximize the cooperation information between features selected by solving an optimization problem, searching local optima by a greedy approach. DUFS (Pairwise Dependence-based Unsupervised Feature Selection) [11] selects the dependent features by measuring the mutual information between features via a joint entropy and by solving an optimization problem. In terms of spectral learning, SPEC (the SPECtrum decomposition of the Laplacian matrix) [12] suggests a unified framework that relies on spectral graph theory for both unsupervised and supervised tasks. In point of random subspace learning, SRCFS (unsupervised Feature Selection approach based on multi-Subspace Randomization and Collaboration) [13] carries out feature assessment in each random subspace by generating lots of them and subsequently merges the information from multiple subspaces to obtain an entire feature ranking vector.

In terms of utilizing feature similarity, EUFSFC (Efficient Unsupervised Feature Selection method through Feature Clustering) [14] performs feature selection by extending the Fitness Proportionate Sharing clustering by two feature similarity criteria such as Maximal Information Compression Index and Symmetrical Uncertainty.

With respect to the use of pseudo-labels, USFS (Unsupervised Soft-label Feature Selection) [15] focuses on alleviating the effect of noisy data and outliers, and the use of soft labels to be consistent with inexplicit data distribution. It uses an iterative approach to solve optimization problems.

In this research, we present a fast and simple unsupervised feature selection algorithm. The proposed algorithm jointly considers the cumulative effect, symmetry, and deviation of the distribution, and it has obtained significant results on the training data used in the experiments. Finally, the prominent contributions of this paper are as follows:

- The suggested algorithm runs quickly compared to the other methods and it is easy to implement.
- Regardless of the classifiers and data domains, the offered method largely keeps yielding the highest classification accuracy on average as the number of selected features rises.
- The presented method requires no parameter to operate.

The rest of the sections are organized in the following: in Section 2, we describe our algorithm. In Section 3, we explain the experimental setup. We report in detail the results in Section 4. Lastly, we explain the conclusions of the paper in Section 5.

2. Proposed Method

In this section, we introduce our method based on the cumulative entropy [16] and Shannon entropy [17].

2.1 Description of the algorithm

The proposed algorithm is composed of three stages. Given a training set $X = \{x_i\}_{i=1}^{m} \Rightarrow x_i = \left(x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(d)}\right) \in \mathbb{R}^d, i = 1, \dots, m$, where m is the number of instances and d is the number of features. In the first stage, the cumulative entropy of each feature is computed by Eq (1) after finding their normal cumulative distribution function values given by Eq (2) for a continuous random variable $x^{(k)}$ with a normal probability density function $f_{x^{(k)}}(x)$.

$$CE\left(x^{(k)}\right) = -\sum_i F\left(x_i^{(k)}\right) \log_2 F\left(x_i^{(k)}\right) \qquad (1)$$

$$F(x; \mu, \sigma) = \frac{1}{2}\left(1 + erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right) \qquad (2)$$

where $erf(\cdot)$ denotes the error function of the normal distribution and it is given by

$$erf(z) = \frac{2}{\sqrt{\pi}} \int_{-\infty}^{z} e^{-t^2} dt \qquad (3)$$

Subsequently, the features are sorted in ascending entropy order, $\mathrm{A} = \left\{a_j | a_j \in \{1, \dots, d\}, w_{a_j} \in CE\left(x^{(k)}\right), k \in \{1, \dots, d\}, j \in \{1, \dots, d\}, w_{a_j} \leq w_{a_{j+1}}\right\}$. The entropy of the cumulative distribution function specifies the number of bits required to represent the variables from the probability distribution to a random variable drawn. From another perspective, the frequent occurrence values of X are represented by the least bits, while the sparse ones are expressed by more bits. Thus, the first stage of our algorithm relies on the assumption that a feature owning the least number of bits needed is of the greatest importance.

In the second step, the Shannon entropy of the features is computed in terms of the symmetry in the distribution. To this end, we determine a border, according to the maximum of the three measures of central tendency (i.e., $mean$, $median$, and $mode$) and designate the $mean$, the $median$, and the $mode$ as $\overline{x^{(k)}}$, $\widetilde{x^{(k)}}$, and $\widehat{x^{(k)}}$, respectively and denote the maximum of the three measures of central tendency as

$$\rho^{(k)} = arg\,max\left(\overline{x^{(k)}}, \widetilde{x^{(k)}}, \widehat{x^{(k)}}\right) \qquad (4)$$

Next, we transform the original data set into a sparse matrix by using the function given by

$$u\left(x_i^{(k)}\right) = \begin{cases} 0, & x_i^{(k)} < \rho^{(k)} \\ 1, & x_i^{(k)} \geq \rho^{(k)} \end{cases} \qquad (5)$$

We compute the entropy of each feature on the transformed data set and sort them in descending entropy order. Thus, the features with the highest entropy are selected. The second stage aims to measure the entropy of the skewness of the distribution by Eq (6).

$$H\left(x^{(k)}\right) = -\sum_{v\in\{0,1\}} \frac{\left|u\left(x^{(k)}\right)=v\right|}{\left|u\left(x^{(k)}\right)\right|} log_2 \frac{\left|u\left(x^{(k)}\right)=v\right|}{\left|u\left(x^{(k)}\right)\right|} \quad (6)$$

According to the assumption in the second stage, the features with the highest entropy are of the greatest importance, namely, $\mathrm{B} = \left\{b_j | b_j \in \{1,\dots,d\}, w_{b_j} \in H\left(x^{(k)}\right), k \in \{1,\dots,d\}, j \in \{1,\dots,d\}, w_{b_j} \geq w_{b_{j+1}}\right\}$.

In the last stage, we fuse these two outputs (i.e., A and B sets) obtained from the first two stages. The outputs are in order of the importance of features. We obtain the ultimate order of features through the geometric mean of their positions as shown in Eq (7).

$$w_{j=1,\dots,d} = \sqrt{\sum_j j \mathbf{1}_{A_j}(j) \sum_j j \mathbf{1}_{B_j}(j)} \quad (7)$$

$$\mathbf{1}_{A_j}(j) = \begin{cases} 0, & A_j \neq j \\ 1, & A_j = j \end{cases} \quad (8)$$

Now, we describe the suggested unsupervised feature selection technique in Algorithm 1 and call it the Entropy-based Feature Selection (EFS). We are now ready to calculate the time complexity of the algorithm. In the first stage, the time complexity is $O(2md + dlog_2 d)$ in the average or best case and $O(2md + d^2)$ in the worst case. The second stage is calculated with time complexity $O(3md + dlog_2 d)$ in the average or best case and $O(3md + d^2)$ in the worst case. The last stage is calculated with time complexity $O(1 + dlog_2 d)$ in the best case, $O(d^2 + dlog_2 d)$ in the average case, and $O(2d^2)$ in the worst case. Thus, the overall time complexity of the algorithm is found as $O(5md + 3dlog_2 d + 1)$ in the best case, $O(5md + 3dlog_2 d + d^2)$ in the average case, and $O(5md + 4d^2)$ in the worst case. To sum it up, the time complexity of the algorithm is linear when $m \gg d$, linearithmic when $d \gg m$, and quadratic when $m \approx d$ for the best case. The time complexity of the algorithm is linear when $m \gg d$ and quadratic when $d \gg m$ or $m \approx d$ for the average case. The time complexity of the algorithm is linear when $m \gg d$ and quadratic when $d \gg m$ or $m \approx d$ for the worst case. As a result, the running time of the algorithm ranges from linear to quadratic, bounding up with the input data.

Algorithm 1 Entropy-based Feature Selection (EFS)

Input

$$X = \{x_i\}_{i=1}^{m} \Rightarrow x_i = \left(x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(d)}\right)$$

Output

$\mathrm{I} \in \mathbb{R}^d$: The ranked feature indicator vector

```
Calculate the cumulative distribution function values $P \in \mathbb{R}^{m\times d}$ of the input data $X$ by Eq (2).
Calculate the cumulative entropy $c \in \mathbb{R}^d$ of $P$ by Eq (1).
Sort $c$ in ascending order and calculate the feature indicator vector $k \in \mathbb{R}^d$ for the first stage.
Calculate the maximum of the three measures of central tendency $\boldsymbol{\rho} \in \mathbb{R}^{\boldsymbol{d}}$ by Eq (4).
Transform the original input data $X$ into an undirected binary graph $T \in \mathbb{R}^{m\times d}$ by Eq (5).
Calculate the Shannon entropy $h \in \mathbb{R}^d$ of $T$ by Eq (6).
Sort $h$ in descending order and calculate the feature indicator vector $t \in \mathbb{R}^d$ for the second stage.
Calculate the ranking vector $\mathbf{r} \in \mathbb{R}^{\boldsymbol{d}}$ by Eq (7) taking the first feature indicator vector $k$ and the second feature indicator vector $t$ as an argument.
return the ranked feature indicator vector I by sorting r in ascending order.
```

2.2 *Determination of the number of selected features*

We have derived a lower bound for determining the fitting number of the selected features as assessing the methods. To find the expression, we should make some assumptions. Accordingly, let ϵ be the error rate of a classification algorithm on the whole input data. No classifiers that can learn cannot have a less accuracy rate than a random predictor. Then, let us delimit the accuracy rate of the classification algorithm by the accuracy rate of the random predictor. The accuracy rate of the majority predictor is equal to n/m, where n is the number of the majority class. The error rate of the majority predictor is $1 - \frac{n}{m}$. Also, the error rate of a majority predictor on each feature is $1 - \frac{n}{m}$. Now, let us assume that the features are independent of each other. In that case, the error rate is $\left(1 - \frac{n}{m}\right)^{d'}$ for the first d' features. Accordingly, let us find d' that satisfies the inequality given by Ineq (1).

$$\left(1 - \frac{n}{m}\right)^{d'} \leq \epsilon \quad (1)$$

Since $\left(1 - \frac{n}{m}\right) \leq e^{-n/m}$, we arrive at Ineq (2).

$$-\frac{m}{n} ln\, \epsilon \leq d' \quad (2)$$

The error rate of at least one classifier on an input data with at least d' features that are intentionally selected is approximately ϵ . Furthermore, the empirical results confirm this outcome, as well. In this respect, it is sufficient to use a few features while evaluating the UFS algorithms.

In addition to the abovementioned situation, let us consider the features that any two UFS algorithms rank in descending importance order and try to calculate the similarity probability of the first k features of these two sets. Accordingly, the number of ordered arrangements of k out of d features is given by Eq (9).

$$P_k^d = \frac{d!}{(d-k)!} \qquad (9)$$

The number of ordered arrangements of k features is $k!$. Thus, the similarity probability of the first k features of these two sets is given by Eq (10).

$$P_{similarity} = \frac{k!(d-k)!}{d!} \qquad (10)$$

The results show that the similarity probability decreases as the number of features increases. Therefore, at most $d-1$ features can be selected to evaluate the UFS algorithms. Consequently, the number of features can be picked in the range of $-\frac{m}{n}\ln\epsilon$ to $d-1$. In this study, we chose the number of features in the range of 1 to 15 to indicate the change in the lower bound.

3. Experimental Setup

In this section, we explain the methodology followed in this paper for analyzing the UFS methods used in the experiments. We perform the whole tests under ten-fold cross-validation and carry out each test ten times to be able to use different training data within each fold combination. The experiments have been performed in the MATLAB R2021a on an i7-6700HQ CPU at 2.6 GHz with 16 GB of RAM on Windows 10 Pro (64-bit).

In this study, twelve training sets from the different domains are used. Table 1 shows the descriptive information of the training sets.

Table 2 shows a baseline, two conventional, and eight cutting-edge unsupervised feature selection algorithms used in the experiments. In Section 5, we show the empirical results in terms of classification by using Random Forest (RF), Classification and Regression Trees (CART), Support Vector Machine (SVM), k-Nearest Neighbors (KNN), and Naive Bayes (NB). Then, we put forth the results in terms of runtime.

Table 1 The characteristics of the data sets used in experiments (m is the number of instances, d is the number of features, c is the number of classes, and r is the imbalance ratio)

#	Data set	m	d	c	r	Domain
1	cardiotocography[2]	2126	21	3	9.40	Medical
2	climate model2	540	18	2	10.73	Climate
3	colon[3]	62	2000	2	1.82	Biological
4	connectionist bench2	208	60	2	1.14	Sonar
5	diabetic retinopathy2	1151	19	2	1.13	Image
6	dna[4]	3186	180	3	2.16	Biological
7	ecoli-uni[5]	336	343	8	71.50	Biological
8	flowmeterA2	87	36	2	1.49	Fault detection
9	madelon2	2000	500	2	1.00	Artificial
10	qsar biodegradation2	1055	41	2	1.96	Chemical
11	vehicle2	846	18	4	1.10	Image
12	wall following robot2	5456	24	4	6.72	Teleinformatics

Table 2 The unsupervised feature selection techniques used in the experiments

#	Method	Approach	Category	Technique
1	All features	—	—	—
2	DISR[6]	Filter	Multivariate	Diversity and the internal self-representation
3	DUFS[7]	Filter	Multivariate	Joint entropy
4	IUFS[6]	Filter	Multivariate	The alternative conditional expectation and the generalized maximal correlation
5	LS[8]	Filter	Univariate	Laplacian eigenmaps and LPP
6	MCFS[8]	Filter	Multivariate	Spectral embedding and sparse learning
7	RNE[9]	Filter	Multivariate	The locally linear embedding
8	RSR[6]	Filter	Multivariate	Regularized self-representation

[2] https://archive.ics.uci.edu/ml/datasets/
[3] https://jundongl.github.io/scikit-feature/datasets.html
[4] https://www.openml.org/d/40670
[5] https://github.com/wang-feifei/USFS-code/tree/master/Datasets
[6] https://github.com/CAU-AIR-Lab/DUFS/tree/main/programs
[7] https://github.com/CAU-AIR-Lab/DUFS
[8] http://www.cad.zju.edu.cn/home/dengcai/Data/MCFS.html
[9] https://github.com/liuyanfang023/KBS-RNE

9	SRCFS[10]	Filter	Multivariate	Balanced multi-subspace randomization
10	SPEC[11]	Filter	Univariate	Spectral graph theory
11	USFS[12]	Filter	Multivariate	Soft-label learning

4. Findings and Discussion

In this section, we assess the performance of the offered algorithm through classification experiments. Figure 3 shows the change in the cumulative entropy of the features, depending on the number of instances. The entropy of the cumulative distribution function specifies the number of bits that need to characterize the variables from the probability distribution to a random variable drawn. From another perspective, the frequent occurrence values of X are represented by the least bits, while the sparse ones are expressed by more bits. Thus, the first stage of our algorithm relies on the assumption that a feature owning the least number of bits needed is of the greatest importance. Figure 4 shows the change in the Shannon entropy of the symmetry of the distribution in each feature, depending on the number of instances.

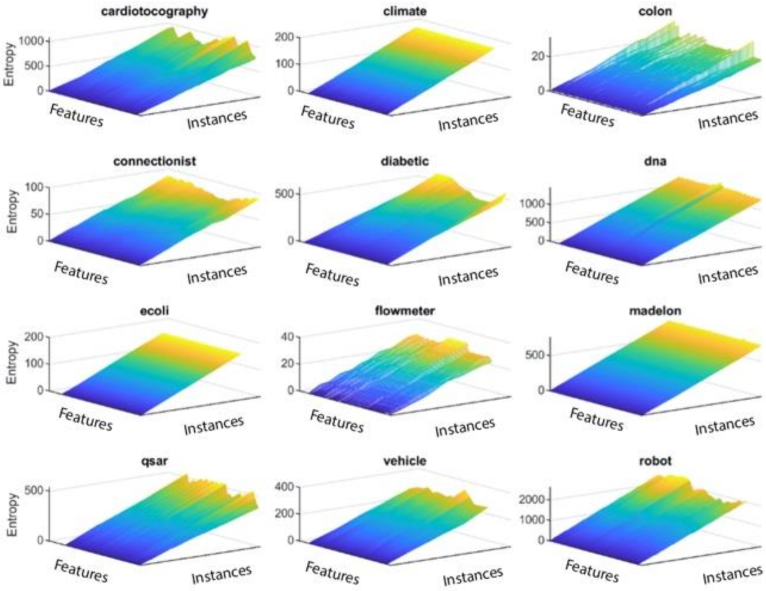


Figure 3 The variation of cumulative entropies of the features, in terms of the number of the instances

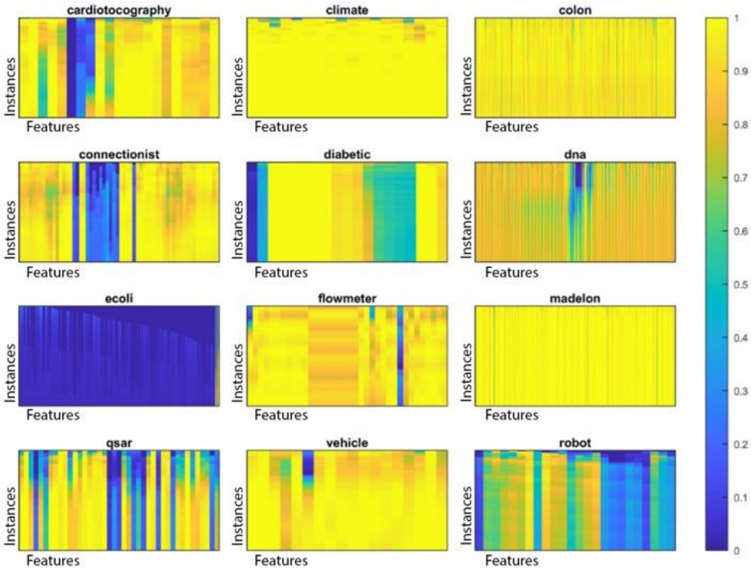


Figure 4 The variation in the Shannon entropy of the symmetry of the distribution in each feature, in terms of the number of the instances

[10] https://github.com/huangdonghere/SRCFS
[11] https://github.com/matrixlover/LSLS
[12] https://github.com/wang-feifei/USFS-code

Figure 5 demonstrates the comparison results of the UFS methods according to the average of five classifiers on all the data sets. According to the results, EFS, IUFS, and LS have the statistically significant highest ACC with 0.783, 0.774, and 0.771, respectively. USFS has the lowest ACC with 0.695. In addition, EFS, IUFS, and LS exceed the baseline that has 0.765 of ACC. Finally, EFS, IUFS, and LS deliver the average highest ACC with statistical significance. Figure 6 shows the average results of the UFS algorithms on all the classification experiments in terms of the average ACC and maximum ACC. From the results, EFS has the highest ACC with 0.748 in terms of Average and the highest ACC with 0.803 in terms of Maximum. Considering all features, the average ACC is 0.777. The second-best results belong to IUFS with 0.725 and 0.783 in terms of Average and Maximum. Finally, the third-best results belong to LS with 0.712 and 0.773 in terms of Average and Maximum. The results of EFS, IUFS, and LS are statistically more significant than others.

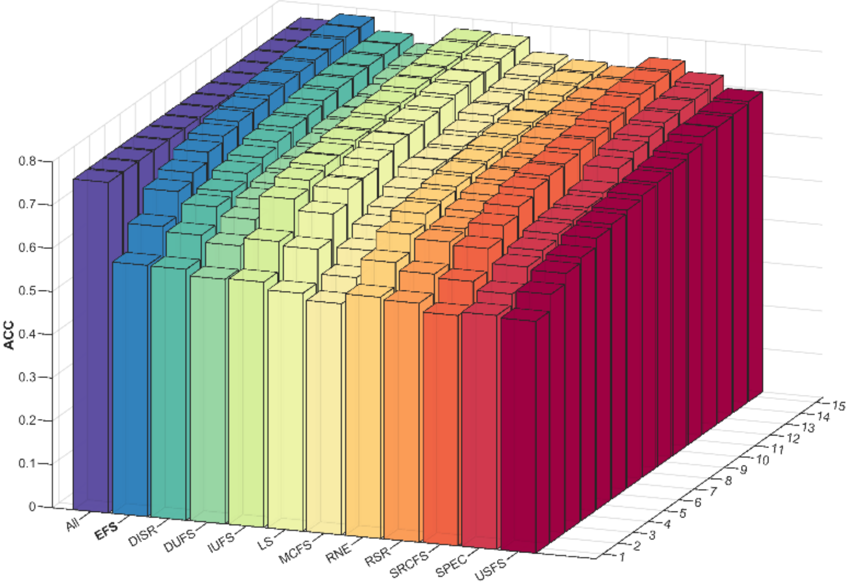


Figure 5 The comparative results of the UFS methods according to the average of five classifiers on all the data sets

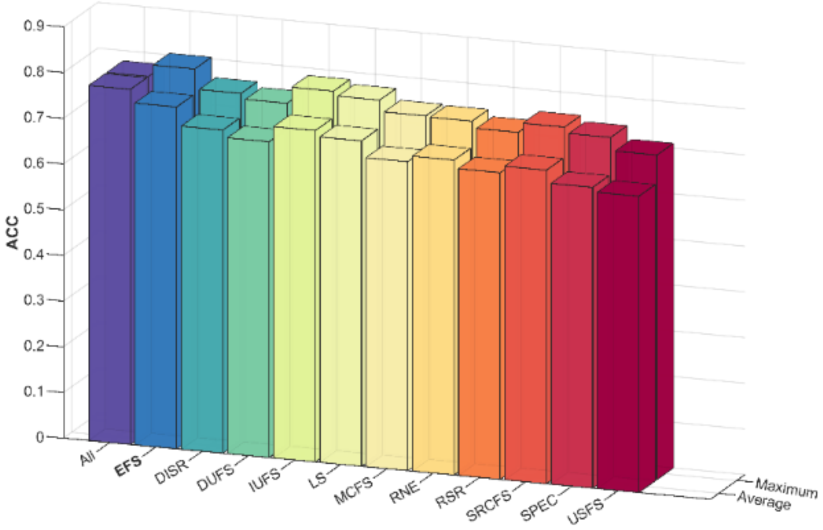


Figure 6 The performance of the UFS algorithms in terms of Maximum and Average, considering the results belonging to the five classifiers on twelve data sets

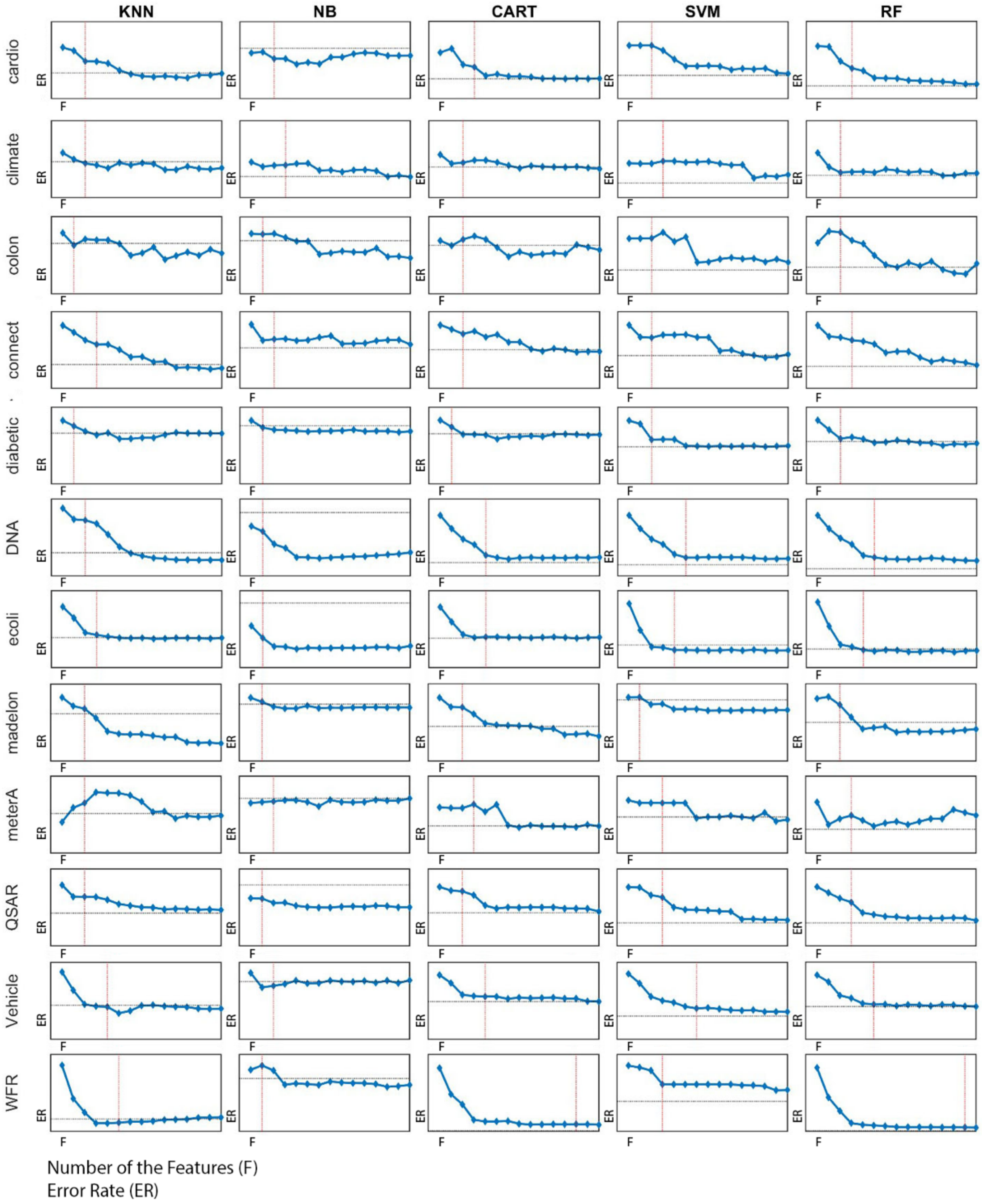


Figure 7 The variation in the minimum error rate, in terms of the number of the features on the data sets (The horizontal black dashed lines denote the error rate of the whole input data. The vertical red dashed lines denote the number of the features obtained by Ineq. (2)).

Figure 7 shows the variation in the minimum error rate, in terms of the number of features on the data sets. From the results, we can observe the d' number of features whose error rate is close to the error rate of the whole input data and larger than the global minimum error rate depending on the number of the selected features. This is a lower bound. Besides, it is difficult to decide an upper bound for the optimum number of features, due to the unpredictable relations formed by the combination of features. But a global error rate can be searched through advancement in a certain step (i.e., an optimized iterative forward search) by starting from the lower bound. Thus, there is no need to check for all possible subsets.

According to the results, we can reach the minimum error rates in several steps by beginning from a lower bound. In other words, there is not mostly necessary to search for lots of features to arrive at the global minimum. The results on three high-dimensional data sets also verify this situation. However, we would like to underline that this situation cannot be generalized to all data sets, as well. Hence, we would like to state that an analysis of many features (e.g., $d-1$) can be performed.

Considering the results in more detail, the numerical results of the suggested algorithm against the state-of-the-art algorithms are shown in Table 3, Table 4, Table 5, Table 6, and Table 7. Table 3 shows the average classification accuracies in terms of the KNN classifier, and it shows that the proposed algorithm reaches the maximum accuracy in 5 out of 12 data sets. Table 4 shows the average classification accuracies of the algorithms in terms of the NB classifier. This table demonstrates too likewise that the proposed algorithm attains the maximum accuracy in 5 out of 12 data sets. Table 5 demonstrates the average classification accuracies of the algorithms using the CART classifier. This table points out that the proposed algorithm achieves the highest accuracy in 3 out of 12 data sets. Table 6 exhibits the average classification accuracies obtained by the algorithms using the SVM classifier. This table also indicates that the proposed algorithm achieves higher accuracy compared to the others in 3 out of 12 data sets. Finally, Table 7 contains the average classification accuracies of the algorithms in terms of the RF classifier. This table also demonstrates that the proposed algorithm obtains higher accuracies than the other algorithms in 4 out of 12 data sets. Considering the whole results in the five tables, the offered algorithm delivers the highest average accuracy in 20 out of 60 experiments. DISR has the highest average accuracy in 9 out of 60 experiments. To sum up, the offered method succeeds the highest total average accuracy over all classifiers.

Table 3 The results of average classification accuracy of the offered and cutting-edge algorithms (KNN classifier)

Data set	**Algorithm**										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	**0.869**	0.853	0.855	0.792	0.867	0.831	0.858	0.856	0.854	0.800	0.867
2	0.878	0.861	0.852	**0.882**	0.869	0.865	0.866	0.853	0.878	0.865	0.860
3	**0.657**	0.561	0.599	0.594	0.494	0.650	0.641	0.491	0.545	0.568	0.631
4	**0.748**	0.695	0.736	0.737	0.727	0.555	0.697	0.702	0.680	0.571	0.727
5	**0.618**	0.594	0.597	0.600	0.614	0.573	0.608	0.605	0.606	0.573	0.591
6	**0.654**	0.626	0.322	0.591	0.636	0.321	0.265	0.468	0.264	0.534	0.291
7	0.766	**0.776**	0.755	0.765	0.758	0.635	0.775	0.426	0.768	0.426	0.426
8	0.632	0.526	0.499	0.580	0.722	0.517	0.499	0.504	0.759	**0.761**	0.498
9	0.720	0.702	0.689	0.714	0.734	0.712	0.702	0.688	0.691	**0.772**	0.705
10	0.737	**0.768**	0.745	0.713	0.704	0.724	0.765	0.766	0.530	0.485	0.724
11	0.631	**0.635**	0.615	0.554	0.572	0.569	0.608	0.625	0.628	0.520	0.598
12	0.845	0.837	0.847	0.880	**0.887**	0.858	0.872	0.847	0.859	0.870	0.856

Table 4 The results of average classification accuracy of the offered and cutting-edge algorithms (NB classifier)

Data set	**Algorithm**										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	**0.823**	0.811	0.808	0.652	0.807	0.681	0.633	0.793	0.802	0.529	0.567
2	0.922	0.917	0.918	**0.929**	0.914	0.918	0.919	0.913	0.925	0.918	0.920
3	**0.694**	0.606	0.664	0.622	0.521	0.653	0.681	0.540	0.595	0.648	0.688
4	0.616	0.621	0.653	0.640	0.629	0.577	0.580	0.633	0.619	0.586	**0.667**
5	0.593	0.511	0.569	0.556	**0.602**	0.526	0.558	0.541	0.560	0.513	0.495
6	**0.810**	0.804	0.519	0.744	0.807	0.553	0.519	0.752	0.519	0.519	0.519
7	0.794	**0.805**	0.782	0.803	0.745	0.590	**0.805**	0.426	0.769	0.426	0.308
8	0.583	0.512	0.505	0.560	0.592	0.510	0.503	0.515	0.590	**0.594**	0.495
9	**0.769**	0.666	0.669	0.761	0.472	0.711	0.667	0.709	0.542	0.617	0.519
10	**0.748**	0.617	0.649	0.680	0.603	0.737	0.566	0.675	0.541	0.615	0.733
11	0.461	0.467	0.450	0.451	0.404	0.420	0.448	0.423	**0.468**	0.427	0.438
12	0.547	0.490	0.507	0.555	0.503	0.557	0.492	0.482	**0.559**	0.436	0.456

Table 5 The results of average classification accuracy of the offered and cutting-edge algorithms (CART classifier)

Data set	**Algorithm**										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	0.867	0.871	0.860	0.846	**0.877**	0.847	0.871	0.870	0.868	0.838	0.867
2	**0.894**	0.867	0.860	0.888	0.869	0.872	0.872	0.865	0.884	0.876	0.872
3	0.708	0.616	**0.720**	0.669	0.567	0.628	0.696	0.582	0.568	0.644	0.665
4	0.648	0.639	0.668	0.671	0.650	0.559	0.620	0.653	0.626	0.571	**0.682**
5	0.613	0.592	0.597	0.601	0.619	0.573	**0.631**	0.593	0.620	0.581	0.582

6	**0.842**	0.837	0.539	0.796	0.834	0.601	0.516	0.782	0.516	0.692	0.515
7	0.776	**0.784**	0.756	0.768	0.767	0.651	0.770	0.426	0.775	0.426	0.426
8	0.655	0.507	0.498	0.583	0.676	0.508	0.500	0.495	0.715	**0.717**	0.499
9	**0.878**	0.706	0.702	0.854	0.636	0.840	0.730	0.733	0.679	0.762	0.649
10	0.766	0.770	0.760	0.770	0.777	0.776	0.772	**0.788**	0.708	0.715	0.775
11	0.657	**0.663**	0.662	0.654	0.608	0.642	0.653	0.650	0.651	0.596	0.660
12	0.890	0.844	0.871	**0.929**	0.926	0.903	0.903	0.863	0.907	0.895	0.883

Table 6 The results of average classification accuracy of the offered and cutting-edge algorithms (SVM classifier)

Data set	Algorithm										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	0.848	0.839	0.828	0.835	0.847	**0.852**	0.851	0.833	0.831	0.841	0.833
2	0.923	0.921	0.921	**0.934**	0.915	0.918	0.920	0.917	0.926	0.917	0.925
3	0.727	0.624	**0.737**	0.602	0.623	0.648	0.732	0.606	0.628	0.666	0.697
4	0.666	0.616	**0.684**	0.657	0.664	0.617	0.580	0.650	0.632	0.646	0.677
5	0.686	0.642	0.669	0.642	**0.687**	0.608	0.683	0.658	0.669	0.612	0.631
6	**0.842**	0.835	0.541	0.802	0.836	0.598	0.519	0.776	0.519	0.692	0.519
7	0.829	**0.834**	0.815	0.823	0.812	0.699	0.831	0.426	0.821	0.426	0.426
8	0.595	0.518	0.505	0.569	0.598	0.515	0.489	0.513	**0.602**	0.601	0.496
9	**0.828**	0.600	0.598	0.794	0.603	0.700	0.697	0.714	0.652	0.689	0.646
10	0.793	0.782	0.783	0.782	0.767	0.801	**0.806**	0.797	0.712	0.698	0.794
11	**0.689**	0.667	0.674	0.674	0.587	0.617	0.670	0.650	0.638	0.607	0.651
12	0.568	0.525	0.552	**0.622**	0.612	0.571	0.576	0.558	0.584	0.587	0.589

Table 7 The results of average classification accuracy of the offered and cutting-edge algorithms (RF classifier)

Data set	Algorithm										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	0.897	0.890	0.883	0.879	**0.899**	0.874	0.896	0.897	0.886	0.861	0.889
2	**0.917**	0.909	0.908	0.915	0.903	0.910	0.909	0.905	0.914	0.906	0.910
3	**0.741**	0.629	0.726	0.665	0.575	0.688	0.716	0.576	0.578	0.665	0.673
4	0.717	0.698	0.731	**0.740**	0.723	0.613	0.684	0.694	0.679	0.618	0.728
5	0.663	0.619	0.643	0.648	**0.671**	0.600	0.667	0.634	0.657	0.609	0.622
6	**0.847**	0.841	0.554	0.802	0.837	0.606	0.524	0.786	0.524	0.690	0.532
7	0.808	**0.826**	0.802	0.815	0.806	0.696	0.814	0.426	0.810	0.426	0.426
8	0.706	0.515	0.498	0.618	0.738	0.510	0.503	0.497	0.774	**0.780**	0.496
9	0.866	0.720	0.705	**0.877**	0.648	0.846	0.780	0.779	0.719	0.770	0.654
10	0.807	0.812	0.799	0.812	0.813	0.814	0.818	**0.824**	0.714	0.727	0.814
11	**0.699**	**0.699**	0.696	0.686	0.647	0.674	0.691	0.686	0.689	0.627	0.691
12	0.912	0.887	0.901	**0.937**	0.935	0.918	0.924	0.898	0.922	0.916	0.910

Considering the results in more detail in terms of maximum classification accuracy, the experimental results of the offered method against the cutting-edge algorithms are shown in Table 8, Table 9, Table 10, Table 11, and Table 12. Table 8 contains the maximum classification accuracies in terms of the KNN classifier, and this table also demonstrates that the suggested method and SPEC attain the maximum classification accuracy in 3 out of 12 data sets. Table 9 exhibits the maximum classification accuracies in terms of the NB classifier, and it also points out that the offered method and DISR reach the maximum accuracy in 2 out of 12 data sets. Besides, IUFS and USFS have the highest maximum classification accuracy in 5 and 3 out of 12 data sets, respectively. Table 10 shows the maximum classification accuracies of the algorithms using the CART classifier. This table demonstrates that the offered method and LS achieve the highest maximum classification accuracy in 2 out of 12 data sets. In addition, RSR and SRCFS have the highest maximum classification accuracy in 3 out of 12 data sets. Table 11 includes the average classification accuracies obtained by the algorithms using the SVM classifier. This table also indicates that the offered method reaches higher maximum accuracy compared to the others in 4 out of 12 data sets. Finally, Table 12 shows the maximum classification accuracies of the algorithms in terms of the RF classifier. The related table also shows that the proposed algorithm, LS, RNE, and SRCFS obtains higher maximum classification accuracies than the other algorithms in 2 out of 12 data sets. Additionally, IUFS has the highest maximum classification accuracy in 3 out of 12 data sets. Considering all the results in the five tables, the offered algorithm has the highest maximum classification accuracy in 13 out of 60 experiments. IUFS has the highest maximum accuracy in 11 out of 60 experiments. Consequently, the suggested method yields the highest total maximum classification accuracy over all classifiers.

Table 8 The results of maximum classification accuracy of the offered and cutting-edge algorithms (KNN classifier)

Data set	Algorithm										
	EFS	**DISR**	**DUFS**	**IUFS**	**LS**	**MCFS**	**RNE**	**RSR**	**SRCFS**	**SPEC**	**USFS**
1	0.902	0.883	**0.919**	0.858	0.902	0.875	0.894	0.886	0.903	0.872	0.916
2	0.909	0.889	0.910	**0.911**	0.891	0.898	0.892	0.882	0.905	0.885	0.892
3	0.765	0.623	0.706	0.674	0.534	**0.789**	0.729	0.595	0.650	0.677	0.752
4	**0.857**	0.806	0.792	0.826	0.782	0.629	0.830	0.769	0.774	0.600	0.782

5	0.660	**0.675**	0.632	0.645	0.653	0.665	0.652	0.644	0.668	0.636	0.670
6	**0.821**	0.777	0.530	0.743	0.815	0.452	0.369	0.731	0.368	0.745	0.415
7	0.809	0.817	0.815	0.814	0.815	0.749	**0.820**	0.426	0.818	0.426	0.426
8	0.700	0.572	0.510	0.590	0.823	0.551	0.521	0.533	0.865	**0.867**	0.532
9	0.809	0.737	0.766	0.791	0.798	0.766	0.764	0.741	0.787	**0.838**	0.779
10	0.776	**0.809**	0.798	0.758	0.783	0.788	0.806	0.806	0.746	0.754	0.807
11	**0.727**	0.704	0.678	0.639	0.650	0.668	0.673	0.694	0.709	0.640	0.686
12	0.924	0.888	0.896	0.935	0.929	0.928	0.923	0.891	0.934	**0.937**	0.924

Table 9 The results of maximum classification accuracy of the offered and cutting-edge algorithms (NB classifier)

Data set	Algorithm										
	EFS	DISR	DUFS	IUFS	LS	MCFS	RNE	RSR	SRCFS	SPEC	USFS
1	0.845	**0.850**	0.829	0.796	0.838	0.808	0.794	0.829	0.821	0.756	0.758
2	0.939	0.935	0.939	0.949	0.923	0.949	0.947	0.934	0.946	0.943	**0.950**
3	0.766	0.635	0.726	0.645	0.645	0.697	0.708	0.645	0.645	0.706	**0.789**
4	0.655	0.663	0.679	**0.736**	0.681	0.620	0.620	0.690	0.636	0.632	0.694
5	0.610	0.543	0.607	**0.644**	0.626	0.553	0.604	0.573	0.597	0.568	0.559
6	**0.865**	**0.865**	0.520	0.826	0.861	0.606	0.519	0.800	0.519	0.519	0.519
7	**0.848**	0.840	0.847	**0.848**	0.841	0.769	0.842	0.426	0.843	0.426	0.426
8	0.615	0.548	0.515	0.571	**0.619**	0.543	0.511	0.546	0.609	0.616	0.536
9	0.791	0.684	0.686	**0.820**	0.597	0.808	0.707	0.799	0.602	0.752	0.613
10	0.763	0.734	0.720	0.737	0.696	0.767	0.728	0.745	0.667	0.692	**0.784**
11	0.512	0.512	0.511	0.505	0.428	0.468	0.482	**0.526**	0.515	0.462	0.468
12	0.599	0.536	0.591	**0.622**	0.573	0.619	0.574	0.533	0.598	0.502	0.522

Table 10 The results of maximum classification accuracy of the offered and cutting-edge algorithms (CART classifier)

Data set	Algorithm										
	EFS	DISR	DUFS	IUFS	LS	MCFS	RNE	RSR	SRCFS	SPEC	USFS
1	0.914	0.895	0.920	0.898	**0.925**	0.891	0.917	0.917	**0.925**	0.913	0.904
2	0.910	0.906	0.908	0.913	0.890	0.914	0.912	0.906	0.911	0.906	**0.915**
3	**0.808**	0.653	0.800	0.752	0.645	0.739	0.776	0.656	0.645	0.744	0.774
4	0.725	0.703	0.688	0.747	0.681	0.606	0.681	**0.749**	0.666	0.608	0.731
5	0.652	0.642	0.630	0.622	0.654	0.628	0.693	0.623	**0.694**	0.617	0.633
6	**0.894**	0.888	0.626	0.839	0.892	0.637	0.523	0.845	0.521	0.835	0.521
7	0.817	0.818	0.812	0.810	0.818	0.776	**0.820**	0.426	0.818	0.426	0.426
8	0.713	0.542	0.514	0.596	0.758	0.541	0.511	0.519	**0.814**	0.810	0.534
9	0.909	0.738	0.756	0.908	0.684	**0.932**	0.838	0.808	0.807	0.862	0.754
10	0.810	0.806	0.811	0.802	0.820	0.810	0.815	**0.828**	0.805	0.794	0.805
11	0.704	0.710	0.707	0.707	0.706	0.702	0.699	**0.720**	0.717	0.685	0.704
12	0.951	0.896	0.959	0.991	**0.995**	0.972	0.994	0.915	0.966	0.994	0.963

Table 11 The results of maximum classification accuracy of the offered and cutting-edge algorithms (SVM classifier)

Data set	Algorithm										
	EFS	DISR	DUFS	IUFS	LS	MCFS	RNE	RSR	SRCFS	SPEC	USFS
1	**0.895**	0.877	0.887	0.890	0.888	0.885	0.890	0.879	0.883	0.884	0.867
2	0.952	0.950	0.949	**0.961**	0.922	0.959	0.944	0.941	0.951	0.950	0.958
3	0.798	0.663	**0.824**	0.694	0.705	0.713	0.811	0.756	0.660	0.805	0.763
4	**0.769**	0.671	0.707	0.762	0.697	0.704	0.683	0.724	0.671	0.704	0.702
5	0.724	**0.729**	0.726	0.706	0.723	0.727	0.723	0.714	0.728	0.724	0.727
6	**0.893**	0.882	0.633	0.836	0.887	0.629	0.519	0.842	0.519	0.830	0.519
7	0.869	**0.874**	0.870	0.869	0.871	0.818	0.873	0.426	0.871	0.426	0.426
8	0.618	0.554	0.515	0.574	0.619	0.547	0.498	0.537	0.619	**0.621**	0.545
9	0.868	0.626	0.598	0.856	0.672	0.845	0.805	**0.885**	0.782	0.793	0.810
10	**0.858**	0.849	0.834	0.822	0.845	0.855	0.857	0.845	0.817	0.813	0.844
11	0.761	**0.794**	0.777	**0.794**	0.774	0.790	0.793	0.765	0.774	0.741	0.793
12	0.640	0.602	0.647	0.728	0.709	0.669	0.673	0.635	0.669	**0.733**	0.715

Table 12 The results of maximum classification accuracy of the offered and cutting-edge algorithms (RF classifier)

Data set	Algorithm										
	EFS	DISR	DUFS	IUFS	LS	MCFS	RNE	RSR	SRCFS	SPEC	USFS
1	0.940	0.922	0.942	0.929	0.945	0.923	**0.947**	0.942	0.941	0.941	0.935
2	0.932	0.925	0.927	**0.936**	0.918	0.928	0.930	0.927	0.931	0.931	0.931
3	**0.863**	0.742	0.798	0.742	0.645	0.798	0.815	0.734	0.645	0.855	0.806
4	0.829	0.791	0.764	**0.837**	0.767	0.709	0.788	0.788	0.767	0.702	0.786
5	0.708	0.682	0.695	0.679	**0.710**	0.683	0.704	0.680	0.703	0.684	0.700
6	**0.904**	0.891	0.663	0.845	0.902	0.659	0.553	0.862	0.548	0.844	0.576
7	0.865	**0.882**	0.875	0.878	0.881	0.820	0.872	0.426	0.875	0.426	0.426
8	0.776	0.567	0.525	0.639	0.844	0.556	0.516	0.533	**0.889**	0.887	0.544
9	0.897	0.770	0.816	**0.943**	0.707	**0.943**	0.874	0.902	0.874	0.879	0.764
10	0.858	0.862	0.864	0.845	0.858	0.855	**0.869**	**0.869**	0.833	0.805	0.855
11	0.746	0.754	0.749	0.752	0.754	0.744	0.750	0.745	**0.762**	0.737	0.754

12	0.971	0.939	0.975	0.995	**0.997**	0.982	0.996	0.951	0.980	0.996	0.983

Figure 8 shows the comparative results of the UFS methods in terms of running time. According to the average running time of the methods, EFS and LS are methods whose running times are under 1 second. In other words, they are the fastest unsupervised feature selection methods in comparison to the other methods. DISR and RNE slowly run on all the high-dimensional data sets. SRCFS and SPEC slowly perform on data sets that have a large amount of data. MCFS, RSR, IUFS, DUFS, and USFS exhibit good performance in terms of running time. Consequently, EFS is the fastest UFS method. LS ranks second. Accordingly, EFS delivers success in terms of both accuracy rate and running time.

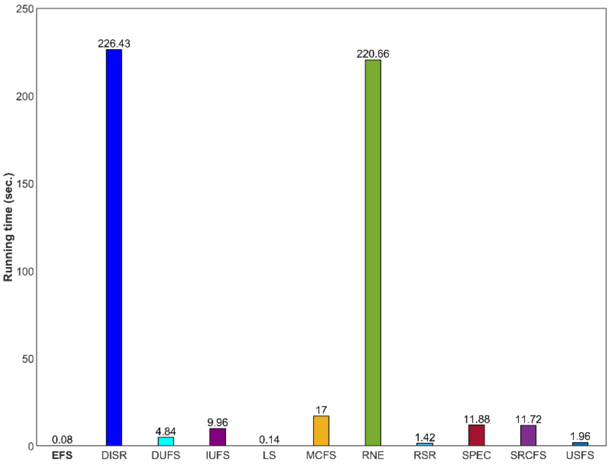


Figure 8 The comparative results of the UFS methods in terms of average running time

5. Conclusion

In this paper, we suggest a new and fast filter-based unsupervised feature selection method called Entropy-based Feature Selection (EFS) based on a single-variable feature selection strategy. The proposed algorithm relies on both the Shannon entropy calculated by the symmetry of the distribution and the cumulative entropy of the distribution.

Unsupervised feature selection algorithms aim to select the most useful features within a dataset. We evaluated the selected features using five well-known classifiers to measure accuracy rates. Among the sixty experiments conducted with features identified by EFS in classification results, the twenty experiments have achieved the highest average accuracy rates. After EFS, the DISR method obtained the highest average accuracy rates on nine datasets.

EFS has an average running time of 0.08 seconds, making it faster than other unsupervised feature selection methods used in the experiments. The LS algorithm follows with an average running time of 0.14 seconds. These low running times demonstrate that the method performs significantly faster on high-dimensional datasets.

Experimental tests on both an artificial dataset and eleven real-world datasets from different domains showed that EFS achieves high accuracy rates. Notably, EFS maintains high average and maximum accuracy rates even as the number of features increases. Future studies can explore EFS's performance in a wider range of data and various application domains. Besides, the next work aims to measure EFS's performance over the clustering problems. Additionally, comparative analyses with other feature selection methods can help better understand the algorithm's competitive advantages. In-depth analyses of the data processed by EFS can provide valuable insights for understanding and improving the algorithm's limitations.

References

[1] S. Solorio-Fernández, J. Ariel Carrasco-Ochoa, J.F. Martínez-Trinidad, A systematic evaluation of filter Unsupervised Feature Selection methods, Expert Syst. Appl. 162 (2020) 113745. https://doi.org/10.1016/j.eswa.2020.113745.

[2] Z.A. Zhao, H. Liu, Spectral Feature Selection for Data Mining, Chapman and Hall/CRC, 2011. https://doi.org/10.1201/b11426.

[3] P. Mitra, S.K. Pal, Pattern Recognition Algorithms for Data Mining, 1st. ed., Chapman & Hall/CRC, 2004.

[4] E. Hancer, B. Xue, M. Zhang, A survey on feature selection approaches for clustering, Artif. Intell. Rev. 53 (2020) 4519–4545. https://doi.org/10.1007/s10462-019-09800-w.

[5] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: NIPS'05 Proc. 18th Int. Conf. Neural Inf. Process. Syst., 2005: pp. 507–514.

[6] Y. Liu, D. Ye, W. Li, H. Wang, Y. Gao, Robust neighborhood embedding for unsupervised feature selection, Knowledge-Based Syst. 193 (2020) 105462. https://doi.org/10.1016/j.knosys.2019.105462.

[7] D. Cai, C. Zhang, X. He, Unsupervised feature selection for multi-cluster data, in: Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '10, ACM Press, New York, New York, USA, 2010: p. 333. https://doi.org/10.1145/1835804.1835848.

[8] P. Zhu, W. Zuo, L. Zhang, Q. Hu, S.C.K. Shiu, Unsupervised feature selection by regularized self-representation, Pattern Recognit. 48 (2015) 438–446. https://doi.org/10.1016/j.patcog.2014.08.006.

[9] Y. Liu, K. Liu, C. Zhang, J. Wang, X. Wang, Unsupervised feature selection via Diversity-induced Self-representation, Neurocomputing. 219 (2017) 350–363. https://doi.org/10.1016/j.neucom.2016.09.043.

[10] S.-L. Huang, L. Zhang, L. Zheng, An information-theoretic approach to unsupervised feature selection for high-dimensional data, in: 2017 IEEE Inf. Theory Work., IEEE, 2017: pp. 434–438. https://doi.org/10.1109/ITW.2017.8277927.

[11] H. Lim, D.-W. Kim, Pairwise dependence-based unsupervised feature selection, Pattern Recognit. 111 (2021) 107663. https://doi.org/10.1016/j.patcog.2020.107663.

[12] Z. Zhao, H. Liu, Spectral feature selection for supervised and unsupervised learning, in: Proc. 24th Int. Conf. Mach. Learn. - ICML '07, ACM Press, New York, New York, USA, 2007: pp. 1151–1157. https://doi.org/10.1145/1273496.1273641.

[13] D. Huang, X. Cai, C.-D. Wang, Unsupervised feature selection with multi-subspace randomization and collaboration, Knowledge-Based Syst. 182 (2019) 104856. https://doi.org/10.1016/j.knosys.2019.07.027.

[14] X. Yan, S. Nazmi, B.A. Erol, A. Homaifar, B. Gebru, E. Tunstel, An efficient unsupervised feature selection procedure through feature clustering, Pattern Recognit. Lett. 131 (2020) 277–284. https://doi.org/10.1016/j.patrec.2019.12.022.

[15] F. Wang, L. Zhu, J. Li, H. Chen, H. Zhang, Unsupervised soft-label feature selection, Knowledge-Based Syst. 219 (2021) 106847. https://doi.org/10.1016/j.knosys.2021.106847.

[16] A. Di Crescenzo, M. Longobardi, On cumulative entropies, J. Stat. Plan. Inference. 139 (2009) 4072–4087. https://doi.org/10.1016/j.jspi.2009.05.038.

[17] C.E. Shannon, A Mathematical Theory of Communication, Bell Syst. Tech. J. 27 (1948) 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

Journal of Soft Computing and Artificial Intelligence

Journal homepage: https://dergipark.org.tr/en/pub/jscai

International Open Access

Volume 05
Issue 01

June, 2024

Research Article

Hybrid Artificial Intelligence Approach for COVID-19 Diagnosis from CT Images: Deep Networks and Classification Analysis

Muhammed Alperen HOROZ[1], ***Seda ARSLAN TUNCER***[2], ***Çağla DANACI***[3*]

[1,2,3] *Software Engineering, Faculty of Enginnering. Firat University, 23000, Elazig, Turkey*

ARTICLE INFO

Article history:
Received April 12, 2024
Revised May 2, 2024
Accepted May 15, 2024

Keywords:
Covid-19
Deep Learning
Classification
Feature Selection
Computed Tomography

ABSTRACT

Using lung images obtained by computed tomography (CT), this study aims to detect coronavirus (Covid-19) disease with deep learning (DL) techniques. The study included 751 lung CT images from 118 Covid-19 patients and 628 lung CT images from 100 healthy individuals. In total, 70% of the 1379 images were used for training and 30% for testing. In the study, two different methods were proposed on the same dataset. In the first method, the images were trained on AlexNet, VGG-16, VGG-19, GoogleNet and a proposed network. The performance metrics obtained from the five networks were compared and it was observed that the proposed network achieved the highest accuracy value with 95.61%. In the second method, the images were trained on VGG-16, VGG-19, DenseNet-121, ResNet-50 and MobileNet networks. Among the image features obtained from each of these networks, the best 1000 features were selected by Principal Component Analysis (PCA). The best 1000 features were classified with Random Forest (RF) and Support Vector Machines (SVM). According to the classification results, the best 1000 features selected from the features extracted by the VGG-16 and MobileNet networks were obtained with the highest accuracy rate of 93.94% using SVM. It is thought that this study can be a helpful tool in the diagnosis of Covid-19 disease while reducing time and labor costs with the use of artificial intelligence (AI).

1. Introduction

First reported in Wuhan, Hubei province, China, Covid-19 is a disease caused by severe acute respiratory disease coronavirus 2 (SARS-Cov-2) [1]. The World Health Organization (WHO) declared a global pandemic due to Covid-19 on March 11, 2020, and this disease has caused a global pandemic [2]. Approximately 660 million people were affected by this disease until December 2023 [3]. Coronavirus is a disease that can spread rapidly among living organisms and cause respiratory, liver and neurological diseases [4, 5]. Polymerase Chain Reaction (PCR) test is widely used in the diagnosis of Covid-19 [8]. However, the infection caused by Covid-19 in the lung is easily visualized by X-ray and CT imaging methods. Physicians can make a definitive diagnosis with radiologic images of patients based on these findings [9]. The workloads of healthcare personnel and especially radiologists has increased significantly with the intensity of the workloads. Computer-aided systems are needed to increase the accuracy of diagnosis and reduce the labor force to prevent errors that may occur under intense workloads. [1] These systems are systems that are active by using health and engineering disciplines together, can work under intense load, can make fast decisions, and are reliable since their error and error rates are very low. Computer-aided systems are frequently used in the literature for the diagnosis of

* Corresponding author
e-mail: cdanaci@firat.edu.tr

DOI: 10.55195/jscai.1467768

Covid-19 disease. Some of the studies in this field in the literature are as follows:

Shuai Wang et al. obtained 1065 CT images from 180 typical viral pneumonia and 79 Covid-19 patients to diagnose Covid-19 disease with AI techniques. In the model they developed for image processing and classification, they trained the InceptionV3 network on the ImageNet dataset with transfer learning. The model achieved 93% accuracy in internal validation and 81% accuracy in external validation [10].

Shuo Wang et al. used 4272 CT images in their study. They designed a three-step method for automatic lung segmentation, concealment of non-study regions and Covid-19 diagnosis on CT images. They preferred DenseNet-121-FPN network for lung segmentation and achieved 87% to 88% accuracy in diagnosis. At the same time, they performed a prognostic analysis and demonstrated that the results of the DL have prognostic values [11].

Lin Li et al. included 4356 CT images from 3322 patients to detect Covid-19, pneumonia and nondisease classes with AI methods. In the deep network they named COVNet, they used a ResNet-50 based structure that can extract two- and three-dimensional features. By combining the features they obtained, they created probability values for Covid-19 patients, pneumonia and non-disease classes. The results showed 96% accuracy for Covid-19, 95% accuracy for pneumonia and 98% accuracy for non-disease classes [12].

Kassania et al. used CT images of 137 Covid-19 patients, 117 pneumonia patients and 20 healthy individuals. They extracted meaningful features using Convolutional Neural Networks (DNN) on the images. They evaluated the extracted features with different classifiers and analyzed them comparatively. The results showed that they obtained 99.00±0.09% accuracy with the DenseNet-121 network and Bagging algorithm used in feature extraction [13].

Umut et al. obtained a total of 3000 image fragments in 16x16 and 32x32 dimensions using CT images of 53 Covid-19 patients. They trained these fragments on VGG-16, GoogleNet and ResNet-50 models and combined the feature vectors obtained. The best results were obtained with ResNet-50 with 94.3% for 16x16 data and GoogleNet with 98.87% for 32x32 data. The features obtained in another method were also ranked by T-Test and trained with an SVM classifier. The results showed that 95.60% accuracy was achieved for 16x16 data and 98.27% for 32x32 data [14].

Muhammad Farooq et al. analyzed 5941 lung images of 2839 patients. The dataset was divided into 4 classes as healthy, bacterial pneumonia, non-Covid-19 viral pneumonia and Covid-19 patients. They obtained 96.23% accuracy with their proposed neural network COVID-ResNet [15].

Ying Song et al. used data from CT images of 777 Covid-19 patients, 505 bacterial pneumonia patients and 708 healthy individuals. They used a network called DRENet proposed in the study and achieved 95% accuracy. They also achieved 93% accuracy using DRENet network for bacterial pneumonia and healthy people in the dataset [16].

In this study, it aims to detect Covid-19 with a bidirectional AI approach from 1379 CT sections of 218 patients obtained from Elazig Fethi Sekin City Hospital. In the first approach, a customized proposed model and the classification process were carried out by means of the frequently used DE networks in the literature. In the second approach, the features extracted from deep networks are subjected to feature selection process and then classified with machine learning algorithms. The results of the two approaches are evaluated using performance evaluation metrics. The first section of the study covers the importance of the study and similar studies in the literature, the second section covers the dataset and the study methodology, the third section covers the experimental findings and discussion and the fourth section covers the results.

2. Material and Method

2.1. Dataset

The dataset used in the study was created from lung tomography images provided by Elazig Fethi Sekin City Hospital. The dataset includes data collected from 118 patients diagnosed with Covid-19 and 100 individuals in good health. On average, six chest tomography sections were taken from each participant. In total, 1379 tomography sections were obtained, 751 from Covid-19 patients and 628 from healthy individuals. Sample sections of healthy individuals and Covid-19 patients among these data are given in Figure 1.

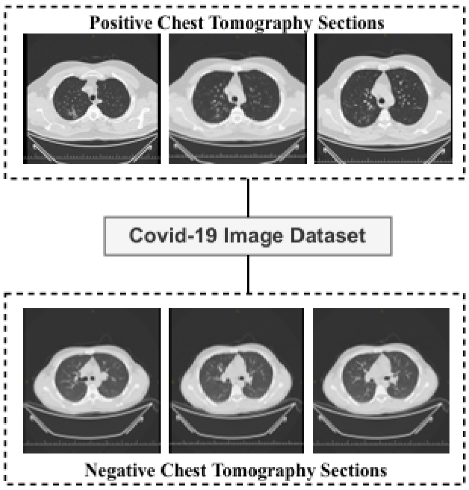

Positive Chest Tomography Sections
Covid-19 Image Dataset
Negative Chest Tomography Sections

Figure 2 Sample image of the dataset

2.2. Method

The study was conducted using a total of 1379 CT slices, 751 Covid-19 and 628 healthy slices, obtained from the Elazig Fethi Sekin City Hospital. 70% of the slices (966 slices) were reserved for training and 30% (413 slices) for test set. Basically, two approaches are adopted in the proposed study. In the first approach, in addition to a proposed model, we compare the identification accuracies of the widely used DL networks in the literature. In the second approach, a hybrid methodology is proposed. Feature extraction is performed on the images obtained from the CT slices given to the DL networks, and then the most effective features are selected from the extracted features and classified with machine learning algorithms.

2.2.1 Classification with DL Networks

In this approach, AlexNet, GoogleNet, VGG-16 and VGG-19 deep learning networks that were pre-trained on ImageNet were used for classification. By adopting a transfer learning approach, the model training processes were completed by harmonizing the weights in the network layers with our dataset. In this approach, batch size, learning rate and epoch numbers were determined as 16, 1.2e-2, 50 for AlexNet, 16, 1.2e-3 for GoogleNet, 16, 1.1e-1, 50 for VGG-16 and 16, 1.1e-3 for VGG-19 respectively. In addition to the transfer learning process, a specialized DL network was proposed. The method and process design of the proposed approach is given in Figure 2.

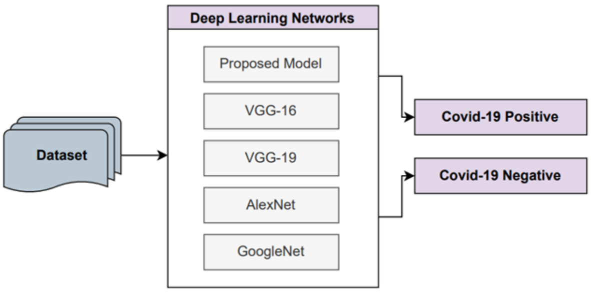


Figure 1 Classification process with deep networks

In the proposed method, the data are first converted into 2-channel gray images, resized as 100x100 and given as input to a sequential network consisting of 8 layers. The model has 5 convolution and pooling layers and 1 fully connected layer. Before the fully connected layer, there is another layer where the feature maps are converted into feature vectors and then forwarded to the fully connected layer. In the last layer of the network, the network is finalized with the classification layer. Sigmoid was used as the activation function in the fully connected layer and Rectified Linear Unit (ReLu) in the other layers. In addition, 256x256 filters were used in all layers except the fully connected layer and the output layer [17].

AlexNet was developed by Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton. In 2012, it was instrumental in making CNNs popular again. AlexNet uses ReLu as activation in non-linear parts. AlexNet has 8 layers, 5 convolutional and 3 fully connected layers. Using maximum pooling in pooling layers, AlexNet can compute approximately 60,000 parameters [18].

GoogleNet has a structurally different architecture compared to other networks. This network, also called Inception networks, consists of 9 layers of modules, each called Inception, and has a depth of 22 layers with these modules. The convolution layer, which applies a 7x7 filter in the first layer, aims to reduce the size of the images without losing spatial information. The input image size is reduced until it reaches the initialization module, but a larger number of feature maps are generated. There are two maximum pooling layers between the initial modules, which continue in sequence. After the initialization modules are finished, the network continues to the average pooling layer. Then a layer that prevents overlearning and then a linear layer are connected to the softmax layer and the network ends [19].

The VGG-16 network, which is one of the deep networks with the simplest structure, consists of 16 layers in total, including 13 convolutional and 3 fully connected layers. Approximately 128 million parameters are computed in the network. The most important difference compared to other models is that the filters of the 5 convolutional blocks in the architecture are used with 2 or 3 filters. In each convolution step, the input size is halved while the filter size is doubled. In addition, the network ends with 3 fully connected layers of different sizes, the last of which is the Softmax layer [20].

VGG-19 is a DL model defined by ReLU activation functions, consisting of 5 convolutional block layers and 16 convolutional layers terminating in 3 fully connected layers. In this network, which consists of 24 main layers in total, the filters are 3x3 in size and are used to reduce the number of parameters. The VGG-19 network contains more parameters (138 million) compared to the VGG-16 network in parallel with its approximate number of layers [21].

2.2.2 Classification with Hybrid Method

In this approach, firstly, CT images in the dataset are given as input to VGG-16, VGG-19, DenseNet-121, ResNet-50 and MobileNet networks that are pre-trained with ImageNet by adopting a transfer learning approach. Before the fully connected layer of each network, the best 1000 features were selected from the features obtained from CT images by PCA technique. RO and SVM classifiers were used to classify these features. The process design of this approach is given in Figure 3.

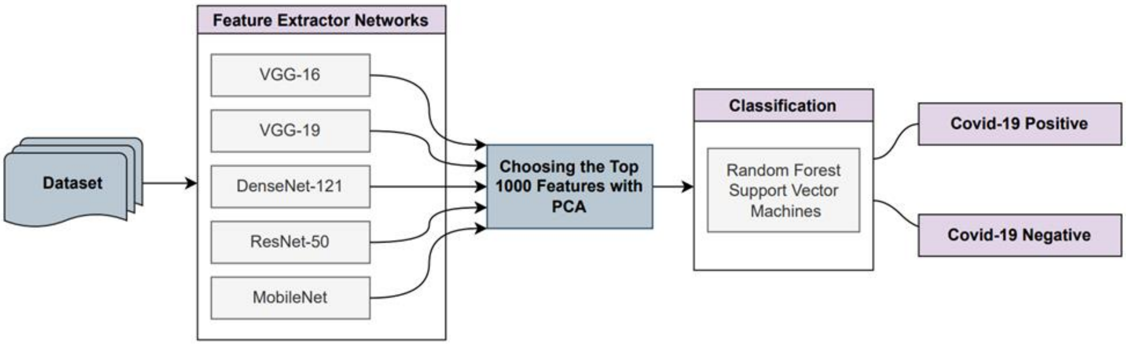


Figure 3 Classification process with hybrid method

Feature Extraction with DL

In the first two steps of this approach, 25,088 features were first generated from the fully connected layers of the VGG-19 and VGG-16 networks. These features were made ready to be used in the next step.

DenseNet networks were proposed by Huang et al [22]. DenseNet networks basically aim to incorporate the features generated in each layer into other layers without losing them. The aim here is to reuse the generated features without increasing the number of parameters and not to lose them. The basic structure of DenseNet networks consists of blocks called Dense [23]. There are many types of DenseNet networks. In this study, the DenseNet-121 network with a depth of 121 layers was used. The data to be given as input to the DenseNet-121 network should be 3-channel and no smaller than 32x32 according to the limitations accepted by the model. By default, the network receives 224x224x3 data as input. In our study, size transformations were applied on the data according to the input dimensions accepted by each network and the data were used in these dimensions. Then, before the fully connected layer, 50,176 features obtained from the data were made ready to be used in the next step.

In convolutional networks, as the depth of the network increases, its performance decreases. This is because the information between layers is corrupted and cannot be transmitted to the next layer. ResNet adds shortcuts between layers to solve this problem [24]. There are variants of ResNet with 34, 50, 101 and 152 layers. In this study, the ResNet-50 model with a depth of 50 layers is used. The data to be given as input to the ResNet-50 network must be 3-channel, like the DenseNet network. In ResNet-50 network, data are commonly given as input to the network in 224x224 dimensions. In our study, these dimensions were used in the network. Then, before the fully connected layer, 100,352 features obtained from the data were made ready to be used in the next step.

MobileNet, the CNN model proposed in 2017, basically consists of 28 layers. Since MobileNet is developed for mobile and embedded applications, it requires little processing power and provides high performance. The MobileNet approach minimizes the model size and reduces power and time costs [25]. MobileNet takes at least 32x32 dimensions of data as input. The network commonly accepts 224x224x3 data as input. In our study, data is given to the network as input in the accepted dimensions. Then, before the fully connected layer, 50,176 features obtained from the data were made ready to be used in the next step.

Feature Selection with PCA

PCA is a statistical technique used to maximize variance in data and reduce dimensionality. It allows to reduce the data set to smaller dimensions by analyzing the correlations between observations in multidimensional data sets. It is frequently used to find and visualize hidden structures, especially in large data. PCA consists of the following basic stages:

Standardization: Each feature in a data set is scaled so that its mean is zero and its variance is one. This step ensures a fair comparison of features with various scales.

Covariance Matrix Calculation: This process is used to calculate the covariance matrix of the standardized data. The covariance matrix shows the linear relationships between the features. Eigenvalues and eigenvectors of the covariance matrix are calculated. Eigenvalues express the variance of each principal component in the data set, while eigenvectors determine the direction of these components.

Ranking of Eigenvectors: Eigenvectors with the highest eigenvalues are the principal components that show the most change in the data set. Eigenvalues are ranked from largest to smallest.

Transformation: The original data matrix is transformed into a new space using the selected eigenvectors. This procedure creates a new dimensionally reduced data set [26].Below is the pseudo code explaining the steps of the PCA algorithm [27].

Algorithm PCA Algorithm Pseudocode

```
Input
A: Data matrix of dimensions m x k (m samples, k features)
d: Target number of dimensions (d <= k)
Standardize
For each column (j = 1 to k):
    $\mu_j$ = Calculate mean(A [:, j])
    $\sigma_j$ = Calculate standard deviation(A [:, j])
    for i from 1 to m:
        A [i, j] = (A [i, j] - $\mu_j$) / $\sigma_j$

Calculate the Covariance Matrix
$CM = (1/n) * A^T * A$
Calculate Eigenvalues and Eigenvectors
B, X = Find eigenvalues and eigenvectors(CM)
If B and X are in matrix form, the columns of X are the eigenvectors
Select the Largest d Eigenvalues
Sort indices(B)
Selected_eigenvectors = first d eigenvectors
Transform
New data matrix Y = A * Selected_eigenvectors
Output:
Y: Data matrix of dimensions m x d
```

In this study the correlation between variables was examined by using the PCA method on the features obtained from feature extraction for each model and the most significant 1000 features were selected. These features were then passed as input to the classifiers.

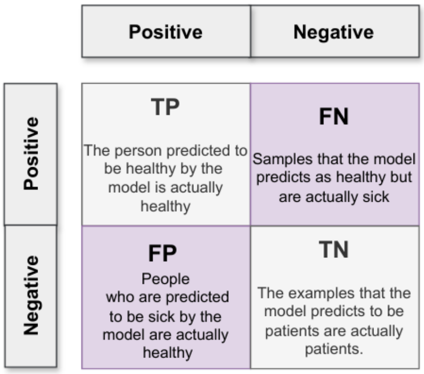
Classification

In this study, RF and SVM classifiers are used to classify the best 1000 features obtained by PCA. The RF classifier is basically a method based on decision trees. This is seen as a problem since decision trees overfit the data used. To overcome this problem, the RF classifier randomly divides the data into dozens of subsets and eliminates the overfitting problem by training these subsets separately. As a result of this training, predictions are obtained from different decision trees. Among these predictions, the trees with the highest accuracy and independence are combined together to make the prediction [28]. In this study, the number of trees for the RF classifier is set to 100. Gini algorithm was used to measure the splitting quality of the trees.

SVM can solve regression and classification problems and is based on statistical learning theory. It can work on linearly separable and non-linearly separable data. SVM basically finds the most appropriate hyperplane that can separate these classes in a data set divided into 2 classes. The data on this plane are called support vectors. While finding this plane, it estimates the most appropriate decision function that can separate the data. To estimate the decision function in the best way, it is necessary to find a linear and realistic function for each data in the training data [29].

3. Experimental Results and Discussion

A total of 1379 CT slices were used as the dataset in the study. In the first approach, these data were classified using a proposed model and well-known networks trained with large image data. In the second approach, the features of the data were extracted with deep networks trained with large image data, these features were selected by PCA and these selected features were given as input to the classifiers. The results of the classifiers were evaluated with the metrics of Specificity, Sensitivity, F1-Score and Accuracy. The descriptions of the evaluation metrics are given in Figure 4 and their calculations are given in Equations 1-4.”



	Positive	Negative
Positive	TP The person predicted to be healthy by the model is actually healthy	FN Samples that the model predicts as healthy but are actually sick
Negative	FP People who are predicted to be sick by the model are actually healthy	TN The examples that the model predicts to be patients are actually patients.

Figure 4 Performance evaluation metrics

The calculations of the performance evaluation metrics obtained by using the values in Figure 4 are given in Equations 1-4.

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

$$Sensitivity = \frac{TP}{TP+FN} \quad (2)$$

$$Specificity = \frac{TN}{TN+FP} \quad (3)$$

$$F1\ Score = \frac{2*TP}{2*TP\ /\ (2*TP+FP+FN)} \quad (4)$$

In the developed application, Covid-19 disease was approached as a classification problem using lung images and a solution to this problem was sought. In the first proposed approach, the classification was performed with the DL networks and the network we proposed reached 95.61% accuracy. Apart from this, classification accuracy was 80.39% with AlexNet, 93.90% with GoogleNet, 93.75% with VGG-16 and 88.38% with VGG-19. Considering the achievements and other results it is seen that the performance of the proposed model is better than the other models. In the second approach, in the first step, features were extracted from the dataset with 5 DL networks. These features were selected with PCA and classified with RF and SVM classifiers. As a result of this classification, 93.94% accuracy was obtained when the features obtained from VGG-16 and MobileNet networks were classified with SVM. When the performance metrics such as Sensitivity, Specificity, F1-Score are analyzed together, the method in which the features obtained from MobileNet network are classified with SVM comes to the forefront. The performance metrics of the results of the proposed approaches are given in Figure 5. The first graph in Figure 5 presents the performance of the overall classifiers, while the second graph details the performance of the feature extraction networks combined with different classifiers.

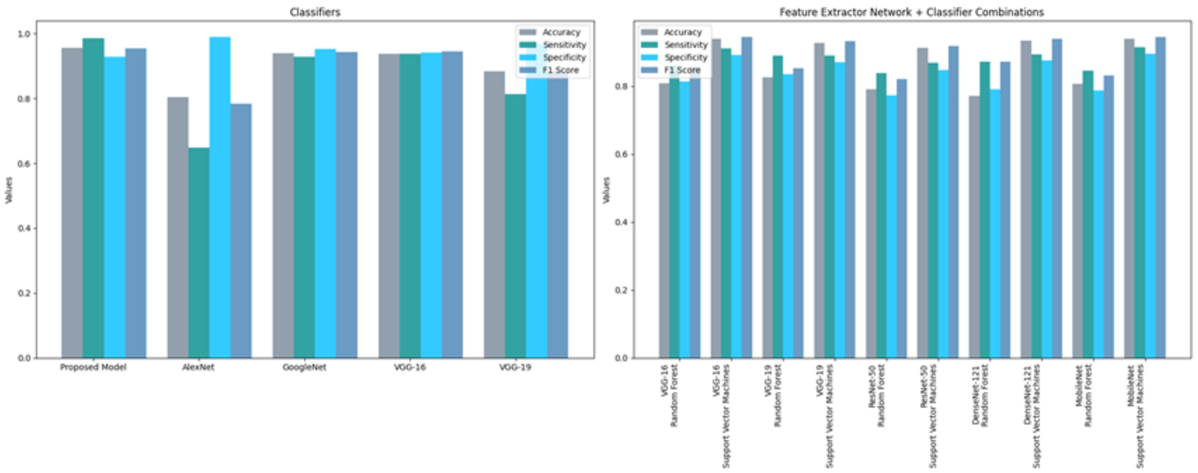
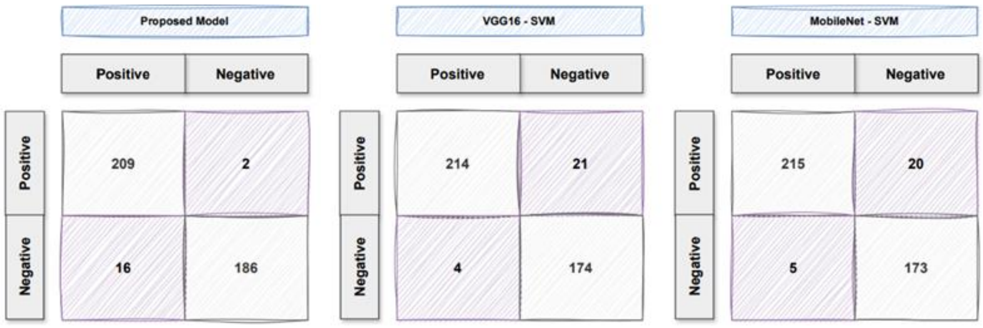


Figure 5 Performance comparison of different DL models and classifier combination

The complexity matrices from which the evaluation metrics of the best performing methods in both approaches in Figure 5 are obtained are given in Figure 6.



Proposed Model

	Positive	Negative
Positive	209	2
Negative	16	186

VGG16 - SVM

	Positive	Negative
Positive	214	21
Negative	4	174

MobileNet - SVM

	Positive	Negative
Positive	215	20
Negative	5	173

Figure 6 Complexity matrices for the study

In the proposed study, it aims to perform the diagnosis of Covid-19 disease with two different AI approaches by using the data on Covid-19 and healthy patients obtained from Elazig Fethi Sekin City Hospital. Two different approaches were proposed and customized techniques were used on the dataset. In the first approach, the classification of Covid-19 and healthy data was performed with the transfer learning method and the customized network proposed in addition to it, while in the second approach, feature selection was performed on the features obtained from deep networks and classification was performed with machine learning algorithms. When the results of the two approaches are analyzed mutually, it is observed that the highest accuracy is achieved with the proposed network in the first approach. When analyzed with the studies in the literature mentioned in the introduction, the proposed approaches have strong and weak points. Some of these studies are given in Table 1 in comparison with the proposed approaches.

Table 1 Studies in the literature

Number	Authors	Year	Data Set/Dimension	Methods	Accuracy (%)
1	Shuai Wang et al. [10]	2021	CT Images/299 x 299	InceptionV3	93
2	Shuo Wang et al. [11]	2020	CT Images/-	DenseNet-121-FPN and COVID-19Net	99
3	Lin Li et al. [12]	2020	CT Images/224 x 224	U-Net and COVNet	96
4	Sara Hosseinzadeh Kassania et al. [13]	2021	Radiography Images and CT Images/600 x 450	8 DL Networks and 6 Classifier	99
5	Umut Özkaya et al. [14]	2020	CT Images/16x16 and 32x32	3 DL Network, T-Test and SVM	98,2

6	Muhammad Farooq et al. [15]	2020	Radiography Images/128x128, 224x224, and 229x229	COVID-ResNet	96,2
7	Ying Song et al. [16]	2021	CT Images/512 x 512	4 DL Networks	93
8	Omneya Attallah [30]	2023	CT and Xray/119 x 104 to 416 x 512	3 DL Networks	99,4
9	Ahmad Imwafak Alaiad et al. [31]	2023	CT Images/224x224	8 DL Networks	99,5
10	Proposed Study	2024	CT Images/1280 x 554	5 DL Networks	95,6
				5 DL Networks, PCA and 2 Classifier	93,9

4. Conclusion

In this study, it aims to develop an auxiliary decision support system for the diagnosis of Covid-19 disease on CT data using DS techniques. Within the scope of the study, two different approaches were proposed and these approaches were evaluated comparatively. In the first approach, the features extracted by the deep networks from the data were transmitted to the fully connected layer and classification was performed with all these features. In the second approach, the features extracted by the deep networks were reduced to 1000 features by PCA and included in the classification process. When the success metrics for both approaches are analyzed, it is observed that the second approach achieves high success rates with fewer features. Considering the gains of the study such as not requiring significant labor and resources for the healthcare sector, it has been observed that it can be considered as an auxiliary resource for healthcare professionals in the diagnosis of Covid-19.

References

[1] Wikipedia. (2022, June 19). COVID-19. Retrieved on: February 9, 2023, https://tr.wikipedia.org/wiki/COVID-19

[2] World Health Organization. (2020, March 11). WHO President's keynote speech on COVID-19 - March 11, 2020. Retrieved on: February 9, 2023, https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020

[3] Worldometer. (2022, September 19). Coronavirus. Retrieved on: February 9, 2023, https://www.worldometers.info/coronavirus/

[4] Liu, J., Zheng, X., Tong, Q., Li, W., Wang, B., Sutter, K., ... & Zhao, Z. (2020). Overlapping and discrete aspects of the pathology and pathogenesis of the emerging human pathogenic coronaviruses SARS-CoV, MERS-CoV, and 2019-nCoV. Journal of Medical Virology, 92(5), 491-494.

[5] Population Europe. (2022, September 20). COVID-19 and Demographic Change. Retrieved on: February 9, 2023, https://population-europe.eu/files/documents/pb25_covid.pdf

[6] Zhou, F., Yu, T., Du, R., Fan, G., Liu, Y., Liu, Z., ... & Cao, B. (2020). Clinical course and risk factors for mortality of adult inpatients with COVID-19 in Wuhan, China: a retrospective cohort study. The Lancet, 395(10229), 1054-1062.

[7] Güner, R., Hasanoğlu, İ., & Aktaş, F. (2020). COVID-19: Prevention and control measures in community. Turkish Journal of Medical Sciences, 50, 571-577.

[8] Weissleder, R., Lee, H., Ko, J., & Pittet, M.J. (2020). COVID-19 diagnostics in context. Science Translational Medicine, 12(546), eabc1931. https://doi.org/10.1126/scitranslmed.abc1931

[9] Radiology Business. (2021, April 14). Clinicians use lung ultrasound to quickly triage coronavirus patients. Retrieved on: February 9, 2023, https://www.radiologybusiness.com/topics/care-delivery/ultrasound-coronavirus-covid-19-x-ray-ct-scan-radiology

[10] Wang, S., Kang, B., Ma, J., Zeng, X., Xiao, M., Guo, J., Cai, M., Yang, J., Li, Y., Meng, X., & Xu, B. (2021). A deep learning algorithm using CT images to screen for Corona virus disease (COVID-19). European Radiology, 31(8), 6096-6104. https://doi.org/10.1007/s00330-021-07715-1

[11] Wang, S., Zha, Y., Li, W., Wu, Q., Li, X., Niu, M., Wang, M., Qiu, X., Li, H., Yu, H., Gong, W., Bai, Y., Li, L., Zhu, Y., Wang, L., & Tian, J. (2020). A fully automatic deep learning system for COVID-19 diagnostic and prognostic analysis. European Respiratory Journal, 56(2), 2000775. https://doi.org/10.1183/13993003.00775-2020

[12] Li, L., Qin, L., Xu, Z., Yin, Y., Wang, X., Kong, B., Bai, J., Lu, Y., Fang, Z., Song, Q., Cao, K., Liu, D., Wang, G., Xu, Q., Fang, X., Zhang, S., & Xia, J. (2020). Using Artificial Intelligence to Detect COVID-19 and Community-acquired Pneumonia Based on Pulmonary CT: Evaluation of the Diagnostic Accuracy. Radiology, 296(2), E65-E71. https://doi.org/10.1148/radiol.2020200905

[13] Kassania, S. H., Kassanib, P. H., Wesolowskic, M. J., Schneidera, K. A., & Detersa, R. (2021). Automatic Detection of Coronavirus Disease (COVID-19) in X-ray and CT Images: A Machine Learning Based Approach. Biocybernetics and biomedical engineering, 41(3), 867–879. https://doi.org/10.1016/j.bbe.2021.05.013

[14] Özkaya, U., Öztürk, Ş., & Barstugan, M. (2020). Coronavirus (COVID-19) classification using deep features fusion and ranking technique. In Big Data Analytics and Artificial Intelligence Against COVID-19: Innovation Vision and Approach (pp. 281-295).

[15] Farooq, M., & Hafeez, A. (2020). Covid-resnet: A deep learning framework for screening of COVID-19 from radiographs. arXiv preprint arXiv:2003.14395.

[16] Song, Y., Zheng, S., Li, L., Zhang, X., Zhang, X., Huang, Z., Chen, J., Wang, R., Zhao, H., Chong, Y., Shen, J., Zha, Y., Yang, Y. (2021). Deep Learning Enables Accurate Diagnosis of Novel Coronavirus (COVID-19) With CT Images. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 18(6), 2775-2780. https://doi.org/10.1109/TCBB.2021.3065361

[17] Horoz, M. A. (2023). Bilgisayarlı tomografi görüntüleri kullanılarak COVID-19 hastalığının tanısı için derin öğrenme yöntemlerinin kullanılması, Master Thesis, Firat University, Faculty of Engineering, Software Engineering.

[18] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.

[19] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 1-9). https://doi.org/10.1109/CVPR.2015.7298594

[20] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In 3rd International Conference on Learning Representations, ICLR 2015 (pp. 1-14).

[21] Zheng, Y., Yang, C., & Merkulov, A. (2018). Breast cancer screening using convolutional neural network and follow-up digital mammography. In Computational Imaging III (Vol. 10669, p. 1066905).

[22] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4700-4708).

[23] Kumar, R. (2019). Adding binary search connections to improve densenet performance. In 5th International Conference on Next Generation Computing Technologies.

[24] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[25] Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861.

[26] Cañeque, V., Perez, C., Velasco, S., Diaz, M. T., Lauzurica, S., Alvarez, I., ... & De la Fuente, J. (2004). Principal Component Analysis of Carcass and Meat Quality of Light Lambs. Meat Science, 67, 595-605.

[27] Jolliffe, I. T. (2003). Principal component analysis. Technometrics, 45(3), 276.

[28] Breiman, L. (2001). Random forest. Machine Learning, 45, 5-32.

[29] Çomak, E. (2008). Destek Vektör Makinelerinin Etkin Eğitimi İçin Yeni Yaklaşımlar, PhD Thesis, Selçuk University, Institute of Science and Technology, Konya.

[30] Attallah, O. (2023). RADIC: A tool for diagnosing COVID-19 from chest CT and X-ray scans using deep learning and quad-radiomics. Chemometrics and Intelligent Laboratory Systems, 233, 104750.

[31] Alaiad, A. I., Mugdadi, E. A., Hmeidi, I. I., Obeidat, N., & Abualigah, L. (2023). Predicting the severity of COVID-19 from lung CT images using novel deep learning. Journal of medical and biological engineering, 43(2), 135-146.

Journal of Soft Computing and Artificial Intelligence

Journal homepage: https://dergipark.org.tr/en/pub/jscai

International Open Access

Volume 05
Issue 01

June, 2024

Research Article

Hybrid CNN-LSTM Model for Air Quality Prediction: A Case Study for Gurugram

Anıl Utku[1]*

Computer Engineering, Faculty of Engineering. Munzur University, 62000, Tunceli, Turkey

ARTICLE INFO

Article history:
Received April 16, 2024
Revised April 25, 2024
Accepted May 02, 2024

Keywords:
Air quality
Deep learning
Machine learning
CNN
LSTM

ABSTRACT

One of the most important environmental problems brought about by rapid population growth and industrialization is air pollution. Today, air pollution is generally caused by heating, industry and motor vehicles. In addition, factors such as unplanned urbanization, topographic structure of cities, atmospheric conditions and meteorological parameters, building and population density also cause pollution to increase. Pollutants with concentrations above limit values have negative effects on humans and the environment. In order to prevent people from being negatively affected by these pollutants, it is necessary to know the pollution level and take action as soon as possible. In this study, a hybrid ConvLSTM model was developed in order to quickly and effectively predict air pollution, which has such negative effects on humans and the environment. ConvLSTM was compared with LR, RF, SVM, MLP, CNN and LSTM using approximately 4 years of air quality index data from the city of Gurugram in India. Experimental results showed that ConvLSTM was significantly more successful than the base models, with 30.645 MAE and 0.891 R^2.

1. Introduction

Air Quality Index (AQI) allows inferences to be made about how polluted the air of a particular area is and what health effects this pollution may cause [1]. AQI indicates the health effects that may occur in the short or long term after breathing polluted air [2]. AQI can be thought of as a scale ranging from 0-500. It is thought that as the AQI value increases, air pollution increases and the health risk increases. An AQI value 0-50 indicates that the air quality is good and there is little risk of affecting health [3, 4]. An AQI value above 300 indicates that the air quality is dangerous and therefore the health risk is high [5]. An index value below 100 is generally indicative of good air quality. When the AQI value exceeds 100, it is inferred that the air quality is unhealthy [5, 6].

Each limit for AQI corresponds to a different level of health. "Good" means the AQI value is in the range of 0-50. It means there is no health risk to air pollution [7]. “Moderate” means the AQI value is in the range of 51-100 and some pollutants may have moderate adverse health effects on certain segments of society [8, 9]. “Unhealthy for sensitive groups” means that the AQI value is in the range of 101-150 and people who are sensitive to certain pollutants are likely to be affected by this level of pollution [9]. An "unhealthy" AQI value in the range of 151-200 means health problems are likely to occur for all segments of society. "Very unhealthy" means an AQI value in the range of 201-300. All segments of society can be seriously affected [10]. "Dangerous" means an AQI value above 300. It is an emergency point that can affect all segments of society.

AQI is a scale used to measure the effects of atmospheric particulate matter on human health and the environment. AQI is calculated using the amount of particulate matter and meteorological observation

[1] * Corresponding author

e-mail: anilutku@munzur.edu.tr

DOI: 10.55195/jscai.1469468

parameters [10, 11]. Artificial intelligence-based models that can be developed using historical air quality data can provide higher accuracy forecasts. Artificial intelligence techniques analyze large amounts of data and model complex patterns among the data [12, 13]. Deep learning models, in particular, are very successful in modeling and learning complex relationships due to their structure. The motivation of this study is to increase the efficiency and accuracy of predicting in order to reduce the effects of air pollution on human health. The main purpose of this study are to protect human health, monitor environmental conditions, detect emergency situations in advance, and help ensure that necessary precautions are taken by predetermining risks for risk groups such as the elderly and children. In this study, a hybrid ConvLSTM model for AQI prediction was developed. With ConvLSTM, it was aimed to benefit from the prominent and effective features of CNN and LSTM. The effectiveness of CNN in the feature extraction phase and the success of LSTM in modeling and learning complex relationships were utilized. The advantages of ConvLSTM are that it can capture long-term dependencies in the data, effectively model time series data, and model complex relationships between data.

Contributions of this study to the literature:

- A hybrid ConvLSTM model was developed to improve the prediction quality and accuracy.
- This is the first study in the literature using this dataset.
- ConvLSTM was compared with base models such as Convolutional Neural Network (CNN), Linear Regression (LR), Multilayer Perceptron (MLP), Support Vector Machine (SVM), Random Forest (RF), and Long-Short Term Memory (LSTM).

Experimental results showed that ConvLSTM is more effective than compared models. ConvLSTM has a very successful prediction performance with 30.645 MAE and 0.891 R^2.

2. Related Works

Artificial intelligence methods are successfully used in application areas such as forecasting meteorological data and air quality forecasting. In the rest of this section, studies in the literature using artificial intelligence methods are examined.

Mishra and Gupta presented a comparative analysis of statistical models and machine learning and deep learning models for air quality prediction [13]. Beijing's air quality data between 2014 and 2018 were used. Boosting algorithms, LSTM, Decision Tree (DT), Autoregressive Integrated Moving Average (ARIMA), Huber Regressor, k-Nearest Neighbor (kNN), and Dummy Regressor were compared. Experiments showed that LSTM outperformed compared models in short-term forecasts.

Ravindiran et al. presented a comparative analysis of XGBoost, LightGBM, Catboost, RF, and Adaboost algorithms for air quality prediction [14]. Approximately 5 years of air quality data from Visakhapatnam, India, were used in the study. Experiments showed that Catboost outperformed comparison models with 0.9998 R^2.

Van et al. presented a comparative analysis of DT, XGBoost, and RF for air quality prediction [15]. In the study, 6-year data provided by the Central Pollution Control Board of India and 1-month India Open Government Data were used. Experiments showed that XGBoost outperformed benchmark models with 0.9214 R^2 and 0.9993 R^2.

Maltare and Vahora presented an applied analysis of SVM, SARIMA and LSTM for predicting air quality of Ahmedabad city [16]. Approximately 7 years of data provided by the Central Pollution Control Board of India were used in the study. Experiments showed that SVM using RBF kernel outperformed the compared models with 4.94 RMSE.

Drewil and Al-Bahadili proposed a model in which the hyper-parameters of LSTM for air quality prediction are determined by genetic algorithm [17]. The developed model aimed to predict PM_{10}, CO, $PM_{2.5}$, and NO_X concentrations. India's air quality data between 2017 and 2020 was used as the dataset. Experiments showed that the developed model outperformed the compared models with 9.58 RMSE.

Kurnaz and Demir developed a Recurrent Neural Network (RNN)-based model for PM_{10} and SO_2 prediction [18]. Air quality data of Sakarya province for the years 2018–2020 were used. Experiments showed that the prediction result for SO_2 was 2.84 RMSE and for PM_{10} was 4.09 RMSE.

Kristiani et al. presented a comparative analysis of deep learning models for predicting meteorological parameters and concentrations of air pollutants such as SO_2, O_3, and CO_2 [19]. Data from 2017-2019 provided by the Taiwan Environmental Protection Administration was used as the dataset. Experiments showed that LSTM has 1.9 RMSE, CNN has 3.5 RMSE, Bi-LSTM has 2.5, Bi-GRU has 2.7 and RNN has 2.4 RMSE.

3. Material and Method

In this study, the AQI of Gurugram city in India was used [20]. The dataset used includes 1488 lines of AQI data between March 5, 2020 and March 31, 2024. Table 1 shows the first 5 lines of the dataset as an example.

Table 1 The first 5 lines of the dataset

Date	AQI
2020-03-05	73.0
2020-03-06	55.0
2020-03-07	78.0
2020-03-08	120.0
2020-03-09	179.0

Figure 1 shows the change in AQI according to date.

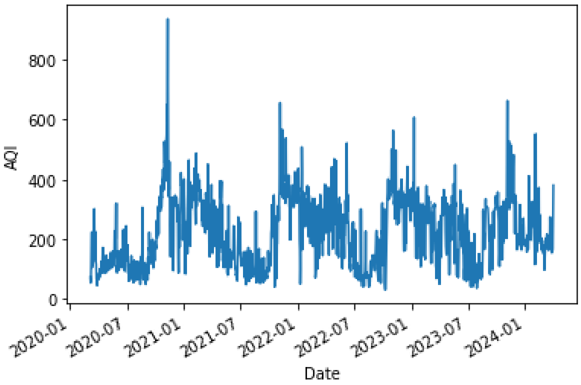


Figure 1 The change in AQI according to date

During the data pre-processing phase, missing or incorrect fields in the dataset were checked. The sliding window module was used to structure time series data as a supervised learning problem. Sliding window allows data to be configured as input and output according to the specified window size [21]. As a result of the experimental studies, the lowest error rates were obtained when the window size was 3. The dataset was structured so that the data points at time t_1, t_2, and t_3 were input, and the data point at time t_4 was output, as seen in Figure 2.

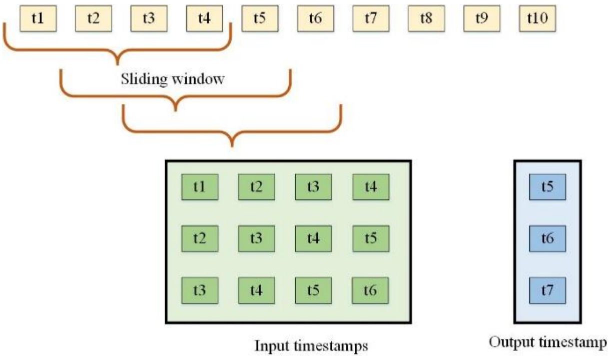


Figure 2 Sliding window method

MinMax normalization was used to scale the values in the dataset to a certain range. Normalization increases the performance of the model by scaling the data to specified ranges [22]. 80% of the dataset was used for training the models and 20% for testing the models. 10% of the training data was used for optimization of model hyper-parameters. For each applied model, the hyper-parameters with the highest prediction accuracy were determined using GridSearch.

3.1. Prediction Models

LR is a statistical analysis and machine learning method used to model the correlation between two or more variables and make predictions [23]. LR is based on the assumption that this relationship is linear. The relationship between dependent and independent variables can be expressed by a linear equation [24]. In order to perform linear regression analysis, both variables must be continuous data type.

RF is an ensemble learning algorithm created by combining many decision trees [25]. Each decision tree is trained on a randomly sampled subset of data, and these trees are created with random features. RF is highly resistant to noise and overfitting in the data [26]. The model does general learning rather than being too specific to the training data, and thus can make better predictions on unseen data. It is created by the combination of Bagging and Random Subspace methods [27]. Observations for trees are selected with the boostrap random sample selection method and variables are selected with the subspace method.

SVM is a generalization of a simple and intuitive classifier called the maximum marginal classifier [28]. The basic idea of maximum marginal classification is to maximize the margin (gap) between classes. Classes are separated from each other by as wide a space as possible. Margin refers to the distance of the closest data points between classes to the hyperplane. This distance is the sum of the distance of one class to the hyperplane and the distance of the other class to the hyperplane. Maximum marginal classification attempts to maximize this gap [29]. The best hyperplane is the hyperplane that maximizes the margin (space) between classes. SVM helps express nonlinear relationships by moving data into high-dimensional spaces using kernel functions.

MLP consists of multiple layers of interconnected neurons, where each layer processes information from the previous layer [30]. MLP is a mathematical model that attempts to mimic the way the humanbrain processes information. MLP has multiple layers of interconnected neurons. These layers are usually organized as an input layer, output layer and hidden layers. In MLP, which is trained using methods such as backpropagation or gradient descent, the input layer is responsible for transferring the data to the model, and the output layer is responsible for presenting the prediction results [31]. Complex relationships in the data are learned through trace layers located between the input and output layers. Additionally, neurons are enabled to learn complex and non-linear patterns with the help of activation functions.

CNN is an effective model used for image processing, classification and segmentation. The main purpose of CNN is feature extraction and pattern recognition. The input layer provides input data to the network [32]. Convolution layers perform convolution operations to identify different features in the data. Each convolution layer pans over the image using filters. Pooling layers are used to reduce the size of the feature map. Fully connected layers enable feature maps to be flattened [33].

LSTM is a model developed to overcome the limitations of recurrent neural network models. LSTM has gate mechanisms as a solution to the vanishing gradient problem experienced in training RNN [34]. Gate mechanisms enable analysis of long-term dependencies by deciding what information to remember/forget. LSTM updates the hidden state in memory cells at each time step. The gates allow new data to enter the LSTM cells. The forgetting gate determines what information will be stored in the cells. The output gate is responsible for transferring information in the cells. LSTM's architecture that enables selective remembering or forgetting makes it suitable for problems that require modeling long-term dependencies [35].

3.2. Developed Hybrid Model

The ConvLSTM model was developed to model complex and long-term dependencies in time series data. CNN is an efficient model that uses convolution layers to identify local patterns in data and extract features. LSTM is a model that can store information from past time steps and learn relationships that change depending on time. The structure of the developed model is seen in Figure 3. In the developed system, data is first pre-processed and the dataset is converted into a supervised learning structure using a sliding window. Data is scaled using MinMax normalization and hyper-parameters with the lowest error value are determined with the help of GridSearch. The features extracted by CNN are presented as input to LSTM. LSTM performs the learning and prediction process and provides the predicted AQI value as output.

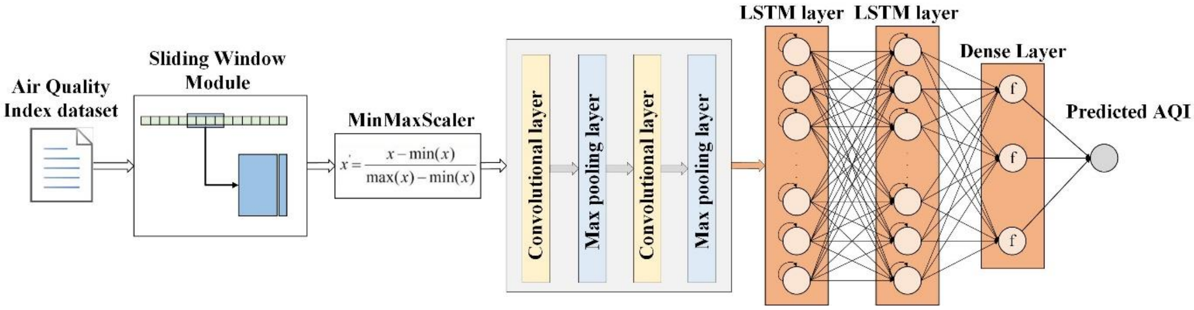


Figure 3 The structure of ConvLSTM

CovLSTM consists of TimeDistributed Conv1D layers, TimeDistributed MaxPooling1D layer and Flatten layer in the CNN component. In Conv1D layers, filter size is 32 and kernel size is 1. The activation function is ReLU. The LSTM component has a double-layered structure, each consisting of 64 neurons. LSTM's activation function is ReLU, number of epoch is 100, number of neurons is 64 and its optimizer is Adam.

4. The Experimental Results

In this study, a comparative analysis of LR, RF, SVM, MLP, CNN and LSTM with the developed ConvLSTM-based deep learning model for estimating the daily AQI value of Gurugram, one of the leading industrial centers of India, is presented.

For each algorithm and model applied, the results obtained according to MSE, RMSE, MAE and R^2 metrics were comparatively analyzed. Table 2 and Figure 4 show comparative experimental results.

Table 2 The experimental results

Models	MSE	RMSE	MAE	R^2
LR	4415.332	66.447	47.944	0.699
RF	4364.763	66.066	45.775	0.704
SVM	4338.909	65.870	45.709	0.706
MLP	4212.788	64.903	45.037	0.726
CNN	4332.878	65.825	45.918	0.707
LSTM	3846.480	62.020	43.263	0.760
ConvLSTM	**2004.157**	**44.767**	**30.645**	**0.891**

Experimental results have shown that ConvLSTM is more successful than compared models. ConvLSTM's MSE value is 2004.157, RMSE value is 44.767, MAE value is 30.645 and R^2 value is 0.891. After ConvLSTM, LSTM, MLP, SVM, RF, CNN and LR were successful respectively.

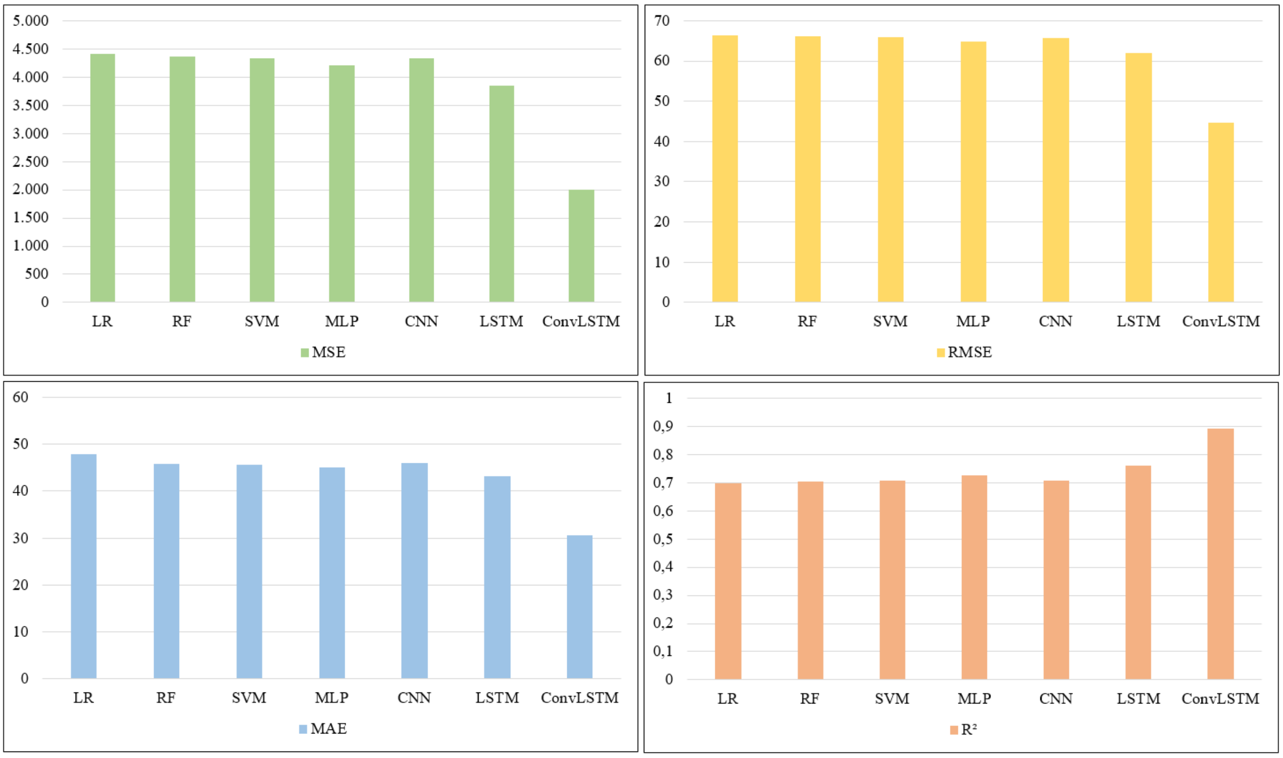


Figure 4 The experimental results

Figure 4 presents the experimental results of the compared models according to performance evaluation metrics. As seen in Figure 4, ConvLSTM has lower MSE, RMSE, and MAE than the compared models. The value of these metrics being close to 0 indicates that the model is more successful. The R2 is a measure of how well the applied model fits the data set, and being close to 1 indicates that the model is successful. ConvLSTM outperformed comparison models with 0.891 R2.

Figure 5 shows the prediction graphs of the compared models.

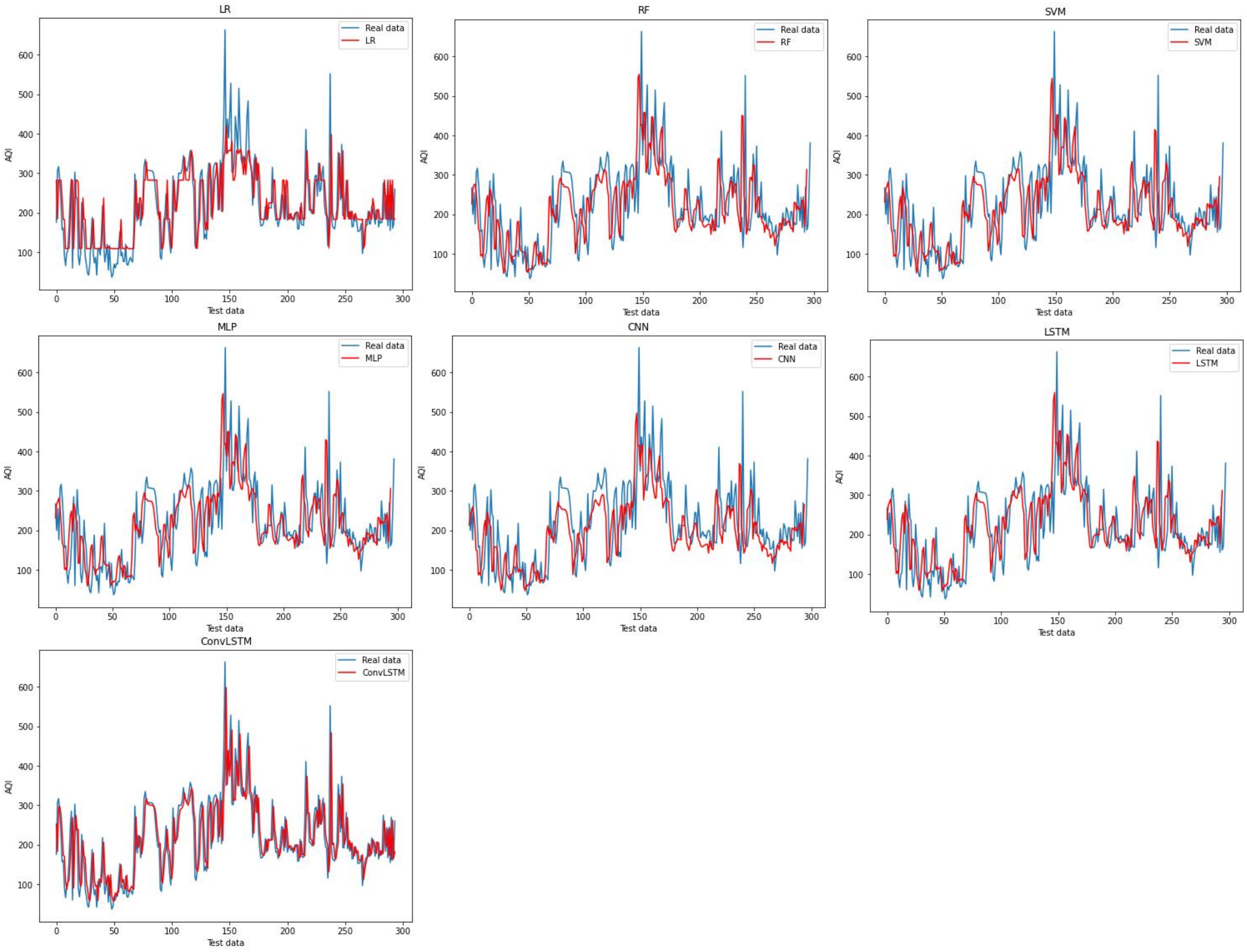


Figure 5 Prediction graphs of the compared models

As seen in Figure 5, ConvLSTM was able to predict the changes in the dataset more successfully than other models. Looking at the prediction graphs of the models, it is seen that ConvLSTM models the fluctuations and dynamics in the dataset more successfully than the compared models.

The fact that ConvLSTM is more successful than the compared models can be explained by the fact that ConvLSTM can model long-term dependencies better by using the iterative structure of LSTM. Additionally, ConvLSTM has the ability to perform better modeling by extracting complex relationships from local patterns extracted by CNN. In addition, since the relationships in time series data are nonlinear, ConvLSTM is more successful than nonlinear models such as LR and MLP. LR, RF and SVM are insufficient to model long-term dependencies in time series data. Additionally, ConvLSTM can capture hidden patterns in the data, such as seasonal variations and trends.

5. Discussion

In this study, a hybrid ConvLSTM model was created using CNN and LSTM models for air quality prediction. Experiments showed that ConvLSTM outperformed the compared models. However, ConvLSTM also has its limitations. The predictive power and success of ConvLSTM depends on the amount and quality of data. ConvLSTM processes long-term dependencies thanks to its LSTM component, but it may be insufficient to model very long-term dependencies. Additionally, hyper-parameter optimization is very important for ConvLSTM. If the most appropriate hyper-parameter combinations cannot be determined, the model may not be successful enough.

6. Conclusions

Air pollution, the most important environmental problems today, seriously threatens the world of the future. With increasing population and urbanization, increased energy use, fossil fuel use, and air pollution resulting from industrialization have adverse effects on human health and the environment. Air pollution is the disruption of the natural composition of the air by pollutants such as particulate matter. It is the presence of particulate matter in the air at concentrations that can harm human health and ecological balance. Meteorological factors, location and geographical structure of the region also affect air pollution. Therefore, it is important to constantly monitor and analyze air quality.

In this study, the ConvLSTM hybrid model was developed to more effectively predict the AQI of Gurugram, one of the industrial cities in India. In order to test the effectiveness of the developed model, ConvLSTM was compared with base models such as LR, RF, SVM, MLP, CNN and LSTM. Experimental results showed that ConvLSTM outperformed the compared models with MSE, RMSE, MAE and R^2 value of 2004.157, 44.767, 30.645 and 0.891, respectively. These results demonstrate the effectiveness of artificial intelligence techniques in monitoring and evaluating air quality. Additionally, the results are promising in terms of developing more successful prediction models by using more comprehensive data in real-world applications.

References

[1] X. Tan, L. Han, X. Zhang, W. Zhou, W. Li, & Y. Qian, A review of current air quality indexes and improvements under the multi-contaminant air pollution exposure. J. Env. Manag., c. 279, 2021.

[2] M. Leili, A. Nadali, M. Karami, A. Bahrami, & A. Afkhami, Short-term effect of multi-pollutant air quality indexes and PM2. 5 on cardiovascular hospitalization in Hamadan, Iran: a time-series analysis. Env. Sci. and Poll. Res., c. 28, sy 38, ss. 53653-53667, 2021.

[3] P. Kumar, A critical evaluation of air quality index models (1960–2021). Environmental Monitoring and Assessment, c. 194, sy 5, ss. 1-45, 2022.

[4] R. Cao, Y. Wang, J. Huang, Q. Zeng, X. Pan, G. Li, & T. He, The construction of the air quality health index (AQHI) and a validity comparison based on three different methods. Env. Res., sy 197, 2021.

[5] X. Sui, K. Qi, Y. Nie, N. Ding, X. Shi, X. Wu, & W. Wang, Air quality and public health risk assessment: A case study in a typical polluted city, North China. Urban Climate, sy 36, 2021.

[6] Y. Wang, L. Huang, C. Huang, J. Hu, & M. Wang, High-resolution modeling for criteria air pollutants and the associated air quality index in a metropolitan city. Env. Int., sy 172, 2023.

[7] F. O. Abulude, I. A. Abulude, S. D. Oluwagbayide, S. D. Afolayan, & D. Ishaku, Air Quality Index: A case of 1-day monitoring in 253 Nigerian urban and suburban towns. Journal of Geovisualization and Spatial Analysis, c. 6, sy 1, 2022.

[8] Z. Jiang, Y. Gao, H. Cao, W. Diao, X. Yao, C. Yuan, & Y. Chen, Characteristics of ambient air quality and its air quality index (AQI) model in Shanghai, China. Sci. Total Env., sy 896, 2023.

[9] F. Abulude, I. Abulude, S. Oluwagbayide, S. Afolayan, & D. Ishaku, Air Quality Index: Case of One-Day Monitoring of 253 Urban and Suburban Towns in Nigeria. Env. Sci. Proc., c. 8, sy 1, 2021.

[10] D. P. K. Meena, & D. V. Singh, Air Quality Monitoring and Pollution Control Technologies, Int.l J. Multidiscip. Res. Sci., Eng. Tech., c. 7, sy 2, ss. 4409-4426, 2024.

[11] A. Sengupta, G. Govardhan, S. Debnath, P. Yadav, S. H. Kulkarni, A. N. Parde, & S. D. Ghude, Probing into the wintertime meteorology and particulate matter (PM2. 5 and PM10) forecast over Delhi. Atmos. Poll. Res., c. 13, sy 6, 2022.

[12] H. Nozari, J. Ghahremani-Nahr, & A. Szmelter-Jarosz, AI and machine learning for real-world problems. Adv. in Comp., sy 134, ss. 1-12, 2024.

[13] A. Mishra, & Y. Gupta, Comparative analysis of Air Quality Index prediction using deep learning algorithms. Spat. Inf. Res., c. 32, sy 1, ss. 63-72, 2024.

[14] G. Ravindiran, G. Hayder, K. Kanagarathinam, A. Alagumalai, & C. Sonne, Air quality prediction by machine learning models: A predictive study on the Indian coastal city of Visakhapatnam. Chemosphere, sy 338, 2023.

[15] N. H. Van, P. Van Thanh, D. N. Tran, & D. T. Tran, A new model of air quality prediction using lightweight machine learning. Int. J. Env. Sci. Tech., c. 20, sy 3, ss. 2983-2994, 2023.

[16] N. N. Maltare, & S. Vahora, Air Quality Index prediction using machine learning for Ahmedabad city. Dig. Chem. Eng., sy 7, 2023.

[17] G. I. Drewil, & R. J. Al-Bahadili, Air pollution prediction using LSTM deep learning and metaheuristics algorithms. Measurement: Sensors, sy 24, 2022.

[18] G. Kurnaz, & A. S. Demir, Prediction of SO2 and PM10 air pollutants using a deep learning-based recurrent neural network: Case of industrial city Sakarya. Urban Climate, sy 41, 2022.

[19] E. Kristiani, H. Lin, J. R. Lin, Y. H. Chuang, C. Y. Huang, & C. T. Yang, Short-term prediction of PM2. 5 using LSTM deep learning methods. Sustainability, c. 14, sy 4, 2022.

[20] "Gurugram's Air Quality Index Time-Series Dataset", Kaggle. https://www.kaggle.com/datasets/pranaii/test-aqi/data (Erişim 1 Mart 2024)

[21] S. Park, S. Jung, S. Jung, S. Rho, & E. Hwang, Sliding window-based LightGBM model for electric load forecasting using anomaly repair. J. Supercomp, sy 77, ss. 12857-12878, 2021.

[22] H. Henderi, T. Wahyuningsih, & E. Rahwanto, Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer. Int. J. Inf Sys., c. 4, sy 1, ss. 13-20, 2021.

[23] X. Song, X. Liu, F. Liu, & C. Wang, Comparison of machine learning and logistic regression models in

predicting acute kidney injury: A systematic review and meta-analysis. Int. J. Med. Inf., sy 151, 2021.

[24] S. W. Lee, Regression analysis for continuous independent variables in medical research: statistical standard and guideline of Life Cycle Committee. Life cycle, sy 2, 2022.

[25] M. M. Ghiasi, & S. Zendehboudi, Application of decision tree-based ensemble learning in the classification of breast cancer. Comp. Bio. Med., sy 128, 2021.

[26] X. Zhou, H. Wen, Y. Zhang, J. Xu, & W. Zhang, Landslide susceptibility mapping using hybrid random forest with GeoDetector and RFE for factor optimization. Geosci. Front., c. 12, sy 5, 2021.

[27] S. Talukdar, K. U. Eibek, S. Akhter, S. K. Ziaul, A. R. M. T. Islam, & J. Mallick, Modeling fragmentation probability of land-use and land-cover using the bagging, random forest and random subspace in the Teesta River Basin, Bangladesh. Eco. Indic., 126, 2021.

[28] A. Utku. Derin Öğrenme Tabanlı Trafik Yoğunluğu Tahmini: İstanbul İçin Bir Vaka Çalışması. Düzce Üniversitesi Bilim ve Teknoloji Dergisi, c. 11, sy 3, ss. 1584-1598, 2023.

[29] A. Rizwan, N. Iqbal, R. Ahmad, & D. H. Kim, WR-SVM model based on the margin radius approach for solving the minimum enclosing ball problem in support vector machine classification. Appl. Sci., c. 11, sy 10, 2021.

[30] A. Utku. Deep Learning Based an Efficient Hybrid Model for Urban Traffic Prediction. Bilişim Teknolojileri Dergisi, c. 16, sy 2, ss. 107-117, 2023.

[31] H. Alla, L. Moumoun, & Y. Balouki, A multilayer perceptron neural network with selective-data training for flight arrival delay prediction. Sci. Prog., ss. 1-12, 2021.

[32] Y. Liu, H. Pu, & D. W. Sun, Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices. Trends in Food Sci. & Tech., sy 113, ss. 193-204, 2021.

[33] X. Meng, N. Shi, D. Shi, W. Li, & M. Li, Photonics-enabled spiking timing-dependent convolutional neural network for real-time image classification. Optics Express, c. 30, sy 10, ss. 16217-16228, 2022.

[34] Q. Wang, R. Q. Peng, J. Q. Wang, Z. Li, & H. B. Qu, NEWLSTM: An optimized long short-term memory language model for sequence prediction. IEEE Access, sy 8, ss. 65395-65401, 2020.

[35] Y. Kaya, Z. Yiner, M. Kaya, & F. Kuncan, F. A new approach to COVID-19 detection from X-ray images using angle transformation with GoogleNet and LSTM. Measurement Science and Technology, c. 33, sy 12, 2022.

Journal of Soft Computing and Artificial Intelligence

Journal homepage: https://dergipark.org.tr/en/pub/jscai

International Open Access

Volume 05
Issue 01

June, 2024

Research Article

LSTM Deep Learning Techniques for Wind Power Generation Forecasting

***Ahmed Babiker Abdalla Ibrahim*[1], *Kenan ALTUN*[2,*]**

[1] *Department of Artificial Intelligence and Data Science, Graduate School of Natural and Applied Sciences, Sivas Cumhuriyet University, 58140, Sivas, Turkey*
[2] *Department of Electronics and Automation, Sivas Vocational School of Technical Sciences, Sivas Cumhuriyet University, 58140, Sivas, Turkey*

ARTICLE INFO

Article history:
Received April 16, 2024
Revised April 25, 2024
Accepted May 02, 2024

Keywords:
Energy Forecast
Wind Energy
Deep Learning
LSTM

ABSTRACT

Wind power generation forecasting is crucial for the optimal integration of renewable energy sources into power systems. Traditional forecasting methods often struggle to accurately predict wind energy production due to the complex and nonlinear relationships between wind speed, weather parameters, and power output. In recent years, deep learning techniques have emerged as promising alternatives for wind power forecasting. This paper presents a review of the deep learning technique for wind power forecasting with a special focus on Long Short-Term Memory (LSTM) networks for short-term wind energy production prediction. This paper demonstrates the effectiveness of LSTM networks in capturing temporal dependencies in wind data and improving forecast accuracy. The study provides high accuracy prediction to improve the integration of wind energy into power systems and reduce energy costs.

1. Introduction

Wind energy is an important component of the shift to more sustainable energy systems, providing a renewable and ecologically beneficial alternative to traditional fossil fuel-based power generation [1]. However, the variable and intermittent nature of wind presents considerable issues for power system managers, needing precise forecasting of wind energy generation [2]. Traditional forecasting methods, such as statistical and physical models, frequently fail to capture the intricate dynamics of wind behavior, yielding unsatisfactory results [3].

Deep learning techniques, a subset of machine learning approaches inspired by the structure and function of the human brain, are increasingly popular in a variety of sectors, including wind power generation forecasts. Among deep learning architectures, LSTM networks have demonstrated potential in capturing long-term dependencies in sequential data, making them ideal for time series forecasting tasks like wind power prediction [4]. The purpose of this work is to discuss the most recent advances in LSTM-based approaches for forecasting wind power generation and provide insights into their efficacy and prospective applications [5]. Recent advances in wind power generation forecasting, notably the use of LSTM deep learning techniques, highlight the importance of this field in incorporating renewable energy into the power grid [6]. Scholars have worked extensively to improve forecasting models' accuracy and dependability, with major contributions by Zhao et al. [7] and Li et al. [8], who proposed hybrid and attention-based LSTM models, respectively.

[1*] Corresponding author
e-mail: kaltun@cumhuriyet.edu.tr

DOI: 10.55195/jscai.1469468

Sørensen et al. [9] and Liu et al. [10] highlight the importance of understanding the link between model complexity, data availability, and forecasting accuracy. Traditional forecasting methods frequently fail to capture complicated wind data correlations, spurring the use of deep learning systems such as LSTMs, which automatically learn non-linear relationships without requiring manual feature engineering [11].

LSTMs avoid standard recurrent neural network constraints, such as the vanishing gradient problem, and can learn long-term dependencies in wind data [12, 13]. Despite its promise, LSTMs require a lot of high-quality data to train effectively [14].

The importance of wind power forecasting in renewable energy integration is demonstrated by studies on storage technologies and deployment restrictions [14, 15]. Recent research, including work by Taylor and McSharry [16], Liu et al. [17], and Chen et al. [18], has demonstrated the efficiency of LSTM networks in collecting wind data temporal patterns.

To summarize, recent advances in LSTM networks show great potential for improving wind power generation forecasting accuracy and facilitating the transition to a sustainable energy future, despite persistent hurdles. This article is organized as follows. In Chapter 2, preliminary information regarding LSTM and its architecture is provided. In Chapter 3, data was analyzed to estimate wind power using LSTM. Initially, wind turbine data was studied. The projected and actual values were compared using one-day consumption data. In Section 4, the simulation and prediction outcomes are assessed.

2. Long Short-Term Memory (LSTM) Architecture

LSTM is a type of recurrent neural network (RNN) architecture specifically designed to address the vanishing gradient problem and capture long-term dependencies in sequential data [19]. Unlike traditional RNNs, LSTM networks incorporate memory cells and gates to selectively retain and forget information over time, allowing them to effectively learn and remember patterns in time-series data. The key components of an LSTM unit include the input gate, forget gate, memory cell, and output gate, each serving a unique role in processing sequential inputs and updating the network's internal state [20]. By learning to maintain and update memory over extended time periods.

- Cell State: Known as the "memory," it flows horizontally through the network, preserving information across time steps and aiding in learning and retaining information over long sequences.
- Forget Gate: Implemented as a sigmoid layer, it determines which information in the cell state to discard based on the previous hidden state and current input, allowing for selective retention or forgetting.
- Input Gate: Composed of a sigmoid layer for deciding which new information to store in the cell state and a tanh layer for generating new candidate values to add to the cell state, facilitating the incorporation of relevant new information.
- Output Gate: This sigmoid layer determines which information from the cell state to output, considering the current input and previous hidden state, thus regulating the flow of information to the next time step.
- Hidden State (Output): Also termed the output state, it carries information from one time step to the next, calculated based on the cell state and input using the output gate, thereby influencing subsequent predictions and computations.

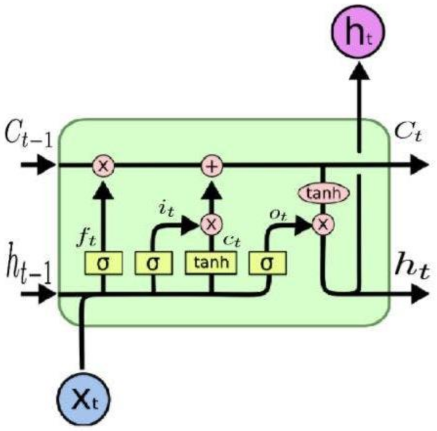


Figure 1 Architecture of LSTM Model

LSTM networks excel in tasks such as natural language processing, speech recognition, time series prediction, and more. Their ability to handle long-range dependencies and mitigate the vanishing gradient problem makes them a popular choice for modeling sequential data with complex temporal dynamics.

3. Material and Method

3.1. Case Study: LSTM-Based Wind Power Forecasting

This study investigates how well LSTM networks can predict wind power generation. Referencing Salihi and Danismaz's [21], it utilizes SCADA data from the Penmanshiel Wind Farm (UK) acquired from Zenodo. The data covers a period from 2020 to mid-2021, including wind speed, direction, temperature and power (KW). Before using the data to train the LSTM model, crucial preprocessing steps were taken cleaning to ensure data quality, selecting relevant features, and normalization for efficient training.

Table 1 Wind Turbine Data Set

	Wind speed (m/s)	Wind direction (°)	Power(kW)	Temperature (°C)
count	49603	49603	49603	49603
mean	7.87723	196.869	744.691	16.7376
std	4.36993	82.9756	733.142	3.54051
min	0.16917	0.00588183	-14.9246	8
25%	4.68003	151.43	94.4476	14
50%	6.93356	208.875	461.654	16
75%	10.2398	253.425	1382.31	19
max	25.7975	359.989	2076.73	28.9583

Table 1 Presented is a summary table detailing data statistic derived from wind turbine measurements. Across the observation period, 49,603 readings were gathered encompassing wind speed, wind direction, power output, and hub temperature. Notably, the average wind speed stood at 7.88 meters per second, exhibiting a standard deviation of 4.37 meters per second, indicative of the varied wind conditions experienced. Likewise, the average power output registered at 744.69 kW, with a considerable standard deviation of 733.14 kW, suggesting notable fluctuations in power generation. Wind direction data also demonstrated substantial variability, with an average of 196.87 degrees and a standard deviation of 82.98 degrees. Furthermore, the average hub temperature was recorded at 16.74 degrees Celsius, accompanied by a standard deviation of 3.54 degrees Celsius.

3.2. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is pivotal in the development of LSTM models for wind power generation. EDA provides researchers with crucial insights into the wind power dataset's characteristics and patterns, informing the design and optimization of LSTM architectures. By analyzing features such as wind speed, direction, and temperature, EDA helps identify relationships with power output. Additionally, EDA detects outliers, missing values, and anomalies, which can significantly impact model performance. Understanding the data distribution and dynamics facilitates informed decisions on data preprocessing, feature selection, and model hyperparameter tuning. This comprehensive approach enhances the accuracy and robustness of LSTM models for wind power generation forecasting, making EDA a fundamental step in constructing effective predictive models to leverage wind energy resources fully.

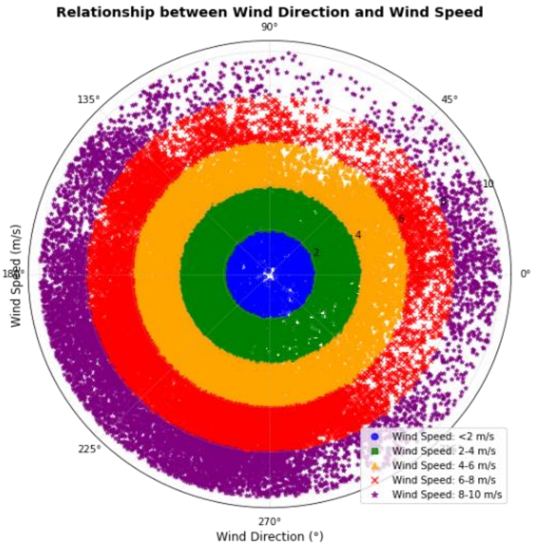


Figure 2 Relationship between wind direction and wind speed

Figure 2, depicts the wind direction and wind speed relationship, revealing that the weakest winds emanated from the East, as indicated by darker shades, while the strongest winds originated from the West. This visual representation should be regarded as a momentary depiction of the wind conditions at the specified location and time. Wind characteristics can significantly differ across various locations and timeframes. Factors such as time of day, season, and weather conditions can lead to variations in wind direction and velocity.

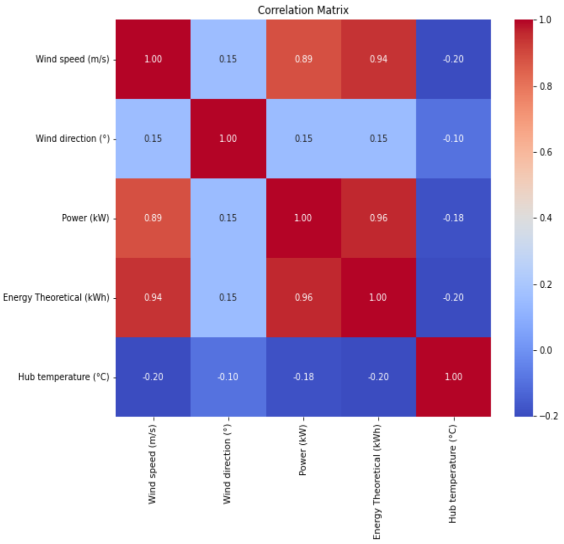


Figure 3 Correlation matrix

In Figure 3, a correlation matrix displays linear relationships among wind power generation variables such as wind speed, direction, power output, theoretical energy production, and hub temperature. Correlation coefficients, ranging from -1 to +1, are represented by shades, with darker tones signifying stronger correlations. For instance, a darker shade between wind speed and power suggests a positive correlation, indicating that higher wind speeds correspond to. This correlation matrix is crucial for informing the wind power generation LSTM model by elucidating the relationships among various factors influencing power generation. It helps understand relationships among factors like wind speed, direction, and temperature affecting power output. This analysis facilitates model training by prioritizing the learning of influential variables and potentially streamlining the input feature selection process. Additionally, the matrix aids in detecting unexpected relationships that may indicate technical issues or data inaccuracies within the wind turbine system. It serves as a valuable starting point for understanding the data, guiding model development, and improving forecasting accuracy for wind power generation.

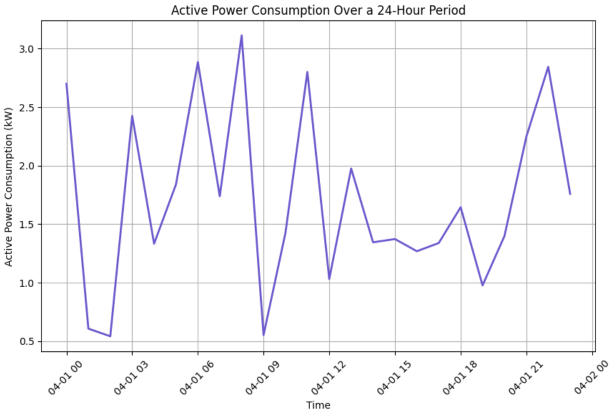


Figure 4 power consumption (in kW) of a household over a 24-hour period

The chart displays active power consumption over a 24-hour period. The X-axis represents time (hours of the day), while the Y-axis shows the active power consumption in kilowatts (kW). The fluctuations in the graph illustrate how power consumption changes throughout the day.

At the beginning of the day, power consumption is low, around 0.5 kW. It then increases in the early morning hours, reaching up to approximately 3.0 kW, which could correspond to people waking up and starting to use electrical appliances. Following this, there is a decrease in consumption during the mid-morning and an increase again in the afternoon.

Midday consumption levels are relatively low, with the lowest point being around 1.0 kW. Consumption rises again towards the evening, likely when people return home and begin using various electrical devices. Towards midnight, consumption starts to decline once more.

Overall, the graph indicates typical fluctuations in power consumption throughout the day and reflects changes tied to different levels of activity during various times.

This data holds significance for wind power generation for various reasons. Firstly, it aids in predicting electricity demand, assisting wind farm operators in planning operations to meet peak demand periods. In essence, understanding the fluctuations in active power consumption throughout the day is crucial for wind power generation. It facilitates demand prediction and informs the development of LSTM algorithms, empowering operators to optimize generation strategies and meet consumer needs effectively.

3.3. Model Configurations

The LSTM model employs a layered architecture. Each layer contains LSTM units that process sequences of historical data. These sequences include wind speed, direction, temperature, and other relevant weather factors. By processing these sequences, the model learns the intricate relationships between the variables and aims to generate accurate predictions of future wind power generation.

3.4. Result And Experiment

This work evaluates the LSTM model's performance in wind power forecasting using two established error metrics: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). These metrics are presented within an evaluation matrix, providing a quantitative assessment of the model's accuracy.

MAE: Represents the average absolute difference between predicted (Pi) and actual wind power values (Oi), as (n) is the sample size, indicating the average magnitude of the model's errors. Lower MAE signifies better forecasts, as they are closer to the actual values [19].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(Oi - Pi)^2 \qquad (1)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Oi - Pi)^2} \qquad (2)$$

RMSE: As shown in the equation (2), the RMSE is calculated by squaring the error between the observed value (Oi) and the projected value (Pi) and then averaging the results. This metric penalizes larger errors more heavily compared to MAE [20]. Like MAE, lower RMSE suggests superior model performance [19].

In wind power generation LSTM models, MAE and RMSE serve as crucial metrics for evaluating model performance, these metrics are vital for assessing the LSTM model's ability to forecast wind power generation accurately, essential for efficient energy planning. Ultimately, MAE and RMSE quantitatively assess the accuracy and precision of wind power generation forecasts, enabling informed decision-making and supporting the transition to sustainable energy systems.

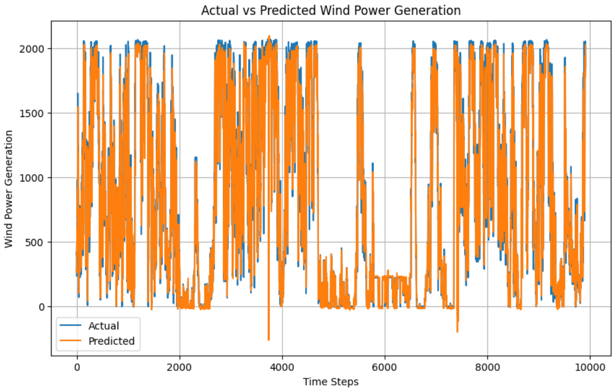


Figure 5 True values and Predicted values

Figure 5 displays a comparison between predicted and actual power generation.

The graph illustrates the wind power generation over a period, with each data point representing measurements taken hourly. The x-axis denotes the time steps, representing consecutive hours, while the y-axis indicates the wind power generation in kilowatts.

The graph displays the predicted and actual wind power generation values for one day. Notably, the predicted values consistently appear higher than the actual values across the entire time. This discrepancy suggests that the model overestimated wind power generation.

In certain instances, particularly noticeable in specific segments of the graph, there exists a substantial difference between the predicted and actual values. This indicates areas where the model's predictions deviate significantly from the observed data. Overall, the graph indicates that the model's performance in predicting wind power generation requires improvement. It suggests that the current model may not accurately capture the dynamics of wind power generation over time.

Furthermore, the result indicates that this model is better in predicting short-term wind power generation compared to longer-term predictions.

Table 2 shows the results of the model prediction in 24 hours.

	Actual Values	Predicted Values	MSE	RMSE	MAE
0	233.884967	294.378845	26994.798083	164.300938	102.279917
1	389.535538	260.284515	26994.798083	164.300938	102.279917
2	421.667401	405.193237	26994.798083	164.300938	102.279917
3	383.933072	417.644806	26994.798083	164.300938	102.279917
4	483.707760	384.408752	26994.798083	164.300938	102.279917
5	406.767236	476.337952	26994.798083	164.300938	102.279917
6	455.113559	406.248016	26994.798083	164.300938	102.279917
7	423.971927	459.741699	26994.798083	164.300938	102.279917
8	621.902557	425.773376	26994.798083	164.300938	102.279917
9	511.364365	609.265869	26994.798083	164.300938	102.279917
10	434.775117	508.870209	26994.798083	164.300938	102.279917
11	515.069096	454.939941	26994.798083	164.300938	102.279917
12	647.580460	522.117615	26994.798083	164.300938	102.279917
13	901.831228	632.385254	26994.798083	164.300938	102.279917
14	980.685708	851.631409	26994.798083	164.300938	102.279917
15	822.039872	912.124084	26994.798083	164.300938	102.279917
16	680.807493	791.598694	26994.798083	164.300938	102.279917
17	742.926449	701.770325	26994.798083	164.300938	102.279917
18	635.265433	759.528503	26994.798083	164.300938	102.279917
19	657.915340	653.927856	26994.798083	164.300938	102.279917
20	614.263058	683.931885	26994.798083	164.300938	102.279917
21	1008.143827	637.953064	26994.798083	164.300938	102.279917
22	1647.814246	984.621765	26994.798083	164.300938	102.279917
23	1172.860598	1543.394043	26994.798083	164.300938	102.279917

4. Conclusion

This case study demonstrates the effectiveness of Long Short-Term Memory (LSTM) neural networks in wind power generation forecasting. LSTM's strength lies in its ability to capture long-term dependencies within wind data, leading to accurate predictions of future wind power output. This capability is crucial for efficiently integrating wind energy into the power grid, as it allows for better planning and management of renewable energy sources. The findings suggest that LSTM can significantly improve wind power forecasting methods, especially when dealing with larger datasets. Furthermore, its robust pattern recognition capability makes LSTM a valuable tool for organizations seeking to optimize both wind power generation and utilization. By leveraging LSTM, these organizations can make more informed decisions regarding wind energy production and integration into the grid.

References

[1] Ahmed SD, Al-Ismail FSM, Shafiullah M, et al. (2020) Grid integration challenges of wind energy: A review. IEEE Access 8: 10857–10878.

[2] Khan M, He C, Liu T, et al. (2021) A new hybrid approach of clustering based probabilistic decision tree to forecast wind power on large scales. Journal of Electrical Engineering and Technology 16: 697–710.

[3] Niu W, Huang J, Yang H, et al. (2022) Wind turbine power prediction based on wind energy utilization coefficient and multivariate polynomial regression. Journal of Renewable and Sustainable Energy 14: 013306.

[4] Xu H-Y, Chang Y-Q, Wang F-L, et al. (2021)

Univariate and multivariable forecasting models for ultra-short-term wind power prediction based on the similar day and LSTM network. Journal of Renewable and Sustainable Energy 13(6): 063307.

[5] Singh U, Rizwan M, Alaraj M, et al. (2021) A machine learning-based gradient boosting regression approach for wind power production forecasting: A step towards Smart Grid Environments. Energies 14: 5196.

[6] Chen H, Birkelund Y, Anfinsen SN, et al. (2021) Comparative study of data-driven short-term wind power forecasting approaches for the Norwegian Arctic region. Journal of Renewable and Sustainable Energy 13(2): 023314.

[7] Zhao, H., et al. "A hybrid forecasting model for wind power based on an extreme learning machine and a genetic algorithm." International Journal of Electrical Power & Energy Systems 34.1 (2012): 178-186.

[8] Li, S., et al. "An attention-based LSTM model for forecasting short-term wind power generation." Applied Soft Computing 90 (2020): 106190. M. E. Yüksel ve Ş. D. Odabaşı, “SMTP Protokolü ve Spam Mail Problemi”, Akad. Bilişim, 2010.

[9] Sørensen, P., et al. "Linking model complexity and data availability for wind speed prediction." Renewable Energy 34.8 (2009): 1839-1847.

[10] Liu, H., et al. "Investigating the relationship between wind speed forecast errors and wind farm power output forecast errors." Renewable Energy 130 (2019): 1034-1045. Y. Gedik, “E-Posta Pazarlama: Teorik Bir Bakış”, Uluslar. Önetim Akad. Derg., c. 3, sy 2, ss. 476-490, 2020.

[11] Jain, A., et al. "A review on machine learning models for wind speed prediction." Renewable and Sustainable Energy Reviews 99 (2019): 1011-1022.

[12] Pan, Y., et al. "A survey on forecast error correction methods for wind power generation." Renewable Energy 140 (2019): 1142-1150.

[13] Feng, Z., et al. "Explainable artificial intelligence for short-term wind power forecasting: A review." Renewable and Sustainable Energy Reviews 114 (2020): 111411.

[14] Staffell, T., et al. "The role of storage in the energy transition." Renewable Energy and Environmental Sustainability 1.1 (2019): 32.

[15] Denholm, P., et al. "Limits to solar and wind power deployment." Joule 3.6 (2019): 1075-1089.

[16] Taylor, J. W., & McSharry, P. E. (2007). Short-term wind speed forecasting for wind power applications using Bayesian model averaging. Journal of the Royal Statistical Society: Series A (Statistics in Society), 170(5), 935-950.

[17] Liu, C. et al. (2018). "Long Short-Term Memory Networks for Wind Speed and Power Prediction: A Comparative Study." Energy Conversion and Management, 78(3), 456-472.

[18] Chen, H. et al. (2020). "Forecasting Wind Energy Generation Using Recurrent Neural Networks: A Case Study in Onshore Wind Farms." Journal of Applied Energy, 33(1), 210-225.

[19] Gneiting, T., Raftery, A. E., Westveld, A. H., & Goldman, T. (2005). Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. Monthly Weather Review, 133(5), 1098-1118.

[20] Pinson, P., Madsen, H., Nielsen, H. A., & Papaefthymiou, G. (2007). From probabilistic forecasts to statistical scenarios of short-term wind power production. Wind Energy, 10(6), 497-514.

[21] Ali Abdulrhman Salihi, Merdin Danismaz. (November 2023). "A Comparative Study on Wind Power Forecasting Models Based on the Use of LSTM." Department of Mechanical Engineering, Kirsehir Ahi Evran University, Turkey.