InComTech: Jurnal Telekomunikasi dan Komputer
vol.14, no.1, April 2024, 30-38
http://publikasi.mercubuana.ac.id/index.php/Incomtech
P-ISSN: 2085-4811 E-ISSN: 2579-6089

Prediction of Sleep Disorder: Insomnia Using AdaBoost Ensemble Learning Algorithm with Grid Search Optimization

Mochammad Anshori*, Wahyu Teja Kusuma[2], Risqy Siwi Pradini[3]

[1,2,3]*Informatika, Institut Teknologi, Sains, dan Kesehatan RS.DR. Soepraoen Kesdam V/BRW*
Jl. Meruya Selatan, Jakarta 11650, Indonesia
*Email Penulis Koresponden: moanshori@itsk-soepraoen.ac.id

***Abstract*:**
Human health is an important thing to keep. Health has to be maintained with appropriate rest. Lack of rest has a bad impact on the body such as hormonal imbalances. One of the causes of lack of rest is insomnia. Insomnia is a phenomenon that describes someone's difficulty sleeping. Insomnia is often considered trivial, but chronic insomnia puts the sufferer at risk of serious illness physically and psychologically. Some people sometimes don't realize that they have insomnia because they feel like they have trouble sleeping. Therefore, early detection of insomnia is necessary to do. This study uses a machine learning approach to make predictions, namely the AdaBoost + grid search method. AdaBoost is used because of its reliability in making strong classifiers and grid search is applied to tuning parameters from AdaBoost. Parameters that are optimized are the n estimator and learning rate. Parameter tuning by grid search gives n – estimator = 76 and learning rate = 0.1. Some preprocess technique is done, there are normalization and ordinal encoding then data splitting based on the determined ratio. There are 80% for training data and 20% for testing data. On training data, the result is 98% percentage for each accuracy, precision, recall, and f1 score. This value is better than the comparison method, it is LogRegression that only reaches 97% value on each evaluation measure. The model implemented on test data and AdaBoost + grid search obtained 100% accuracy, precision, recall, and f1 score. However, LogRegression only gives 98% result. This study proved that AdaBoost with grid search is sustainable to do early prediction of insomnia.

***Keywords*:**
Mchine learning;
Boosting algorithm;
Healthinformatic;
insomnia;

Riwayat Artikel:
Received Feb 09, 2023
Revised Jan 10, 2024
Accepted Jan 23, 2024

DOI:
10.22441/incomtech.v14i1.19306

1. INTRODUCTION

In terminology, insomnia is often reported as a person's difficulty sleeping. The term insomnia is also used to represent sleep disturbances [1]. Its characteristics are the inability of the body to fall asleep and wake up or wake up from sleep early. Insomnia is not limited to sleep problems, it is even a risk factor for other diseases such as cardiovascular disease, anxiety disorders, sleep apnea, nightmares, and enuresis. [2]. The prevalence rate of insomnia reaches 0 to 60% of the population. There are two kinds of insomnia, namely acute and chronic. Chronic insomnia is associated with a short life span, is offset by neurological disorders such as Parkinson's, and can have physical and psychological effects.[3]. While acute insomnia is the occurrence of insomnia between two weeks to 3 months [4].

Based on fact, in French there are 19% population with insomnia, and in South Korea the are 11.5% population with insomnia [5]. Nowadays not only adults, but the youth people also have potential insomnia. In fact, in South Korea, teenagers aged 15 have experienced insomnia. Insomnia, even though it's not a serious illness, in some cases, people who have acute insomnia can get psychological damage or mental issues [2]. Such as depression, schizophrenia, anxiety disorders, and eating disorders which can make someone choose to commit suicide [5]. This happens because insomnia has a major impact on sleep quality [6].

Poor sleep has an impact on daytime activity [7] because sleep has an important role in cognitive functions such as memory recall. With quality sleep, it makes the reception of new memories in the brain better [8]. Many people misunderstand insomnia, it is difficult to distinguish between insomnia and common sleep problems. As a preventive way, it is necessary to predict insomnia to prevent further effects of insomnia. One way to make predictions is with artificial intelligence (AI) technology. AI refers to machine learning (ML) as a technique that aims to apply human learning capabilities to computers. AI adoption in the US experienced a significant increase from 9% to 16%[9], this shows that AI can replace traditional methods. In this paper, researchers will use AI, especially ML as a technique to predict insomnia early.

This proposed research is aimed at helping the expertise to accelerate the diagnosis of insomnia problems. The output of this research is a machine learning model using the AdaBoost algorithm to predict insomnia or not. AdaBoost is one of supervised learning that can do classification and can avoid overfitting. To know the better model, this research also compares the performance of AdaBoost with previous research based on the evaluation of classifier metrics.

2. RESEARCH METHOD

Generally, there are five stages in this research, data acquisition to obtain the dataset. Then data preprocessing includes load data and data transformation. The data partition is splitting the dataset by determining the percentage ratio. Next, create a classifier model using the AdaBoost classifier and the last is an evaluation to get the best performance. The stages are shown in Figure 1

Figure 1. Research methodology

The first stage is data acquisition. The dataset is obtained from the Mendeley dataset [10]. Dataset consists of 9 features with 1 target attribute and contains 420 instances of data. Mode all of the data type is categorical with values between yes and no because the dataset was collected by questionnaire. The features are based on indications of insomnia disorder. The details of the dataset are shown in Table 1. Target features are named disorders that contain yes and no insomnia. The target is binary classification. Distribute of class data is shown in Figure 2. Based on the figure, the total for insomnia data is 210 instances, and the last 210 instances for normal without insomnia. It means the data class is distributed and balanced well.

Table 1. Dataset explanation

No	Features	Data Type	Data Range
1	Total_sleep_time(hour)	numerik	0 - 13
2	Satisfaction_of_sleep	categorical	Yes, No
3	Late_night_sleep	categorical	Yes, No
4	Wakeup_frequently_during_sleep	categorical	Yes, No
5	Sleep_at_daytime	categorical	Yes, No
6	Drowsiness_tiredness	categorical	Yes, No
7	Duration_of_this_problems(years)	numerik	0 - 25
8	Recent_psychological_attack	categorical	Yes, No
9	Afraid_of_getting_asleep	categorical	Yes, No
10	Disorder	categorical	Yes, No

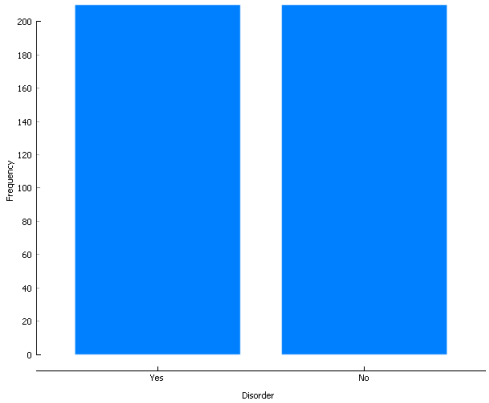

Figure 2. Class distribution

The second stage is data preprocessing. This research implements ordinal encoding and normalization scaler as the preprocessing method. In ordinal encoding, each category is assigned as an integer [11] . In this case the value yes

will be transformed as 1 and no transformed to 0. Then normalization is implemented to make all numerical features have a uniform range [12]. Determine range is lie between 0 to 1. The equation of normalization is shown in Equation 1 below. x is data and x' is new data. min equals to minimum and max equals to a maximum value of each column.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \qquad (1)$$

The third stage is data partition by splitting data based on a determined percentage. The data will be split into two, train data with a ratio of 80% and test data with a ratio of 20% of the dataset. Train data is used to create a model and test data to validate the model. This scenario is based on a comparative study [2]. Then data train will be trained using cross-validation with k-fold = 10. Cross-validation strategy by splitting data into determined k-fold values. Then one segment of its split is used to test a model and another segment is used for training a model. This is happening iteratively till the k-fold value is satisfied [13]. A reason to use cross-validation is the created model is more general and able to avoid the overfitting [14]. The illustration shown at Figure 3 below.

k1	k2	k3	k4	k5	k6	k7	k8	k9	k10
k1	**k2**	k3	k4	k5	k6	k7	k8	k9	k10
k1	k2	**k3**	k4	k5	k6	k7	k8	k9	k10
k1	k2	k3	**k4**	k5	k6	k7	k8	k9	k10
k1	k2	k3	k4	**k5**	k6	k7	k8	k9	k10
k1	k2	k3	k4	k5	**k6**	k7	k8	k9	k10
k1	k2	k3	k4	k5	k6	**k7**	k8	k9	k10
k1	k2	k3	k4	k5	k6	k7	**k8**	k9	k10
k1	k2	k3	k4	k5	k6	k7	k8	**k9**	k10
k1	k2	k3	k4	k5	k6	k7	k8	k9	**k10**

Figure 3. Cross-validation illustration

The fourth stage is creating a prediction model using AdaBoost (adaptive boosting) algorithm. AdaBoost is a method that boosting its performance for binary classification[15]. AdaBoost works by committee the weak classifier and enhances its, model, to get optimum performance. The method's name in AdaBoost is Stagewise Additive Modeling using a Multi-class Exponential loss function (SAMME) [16]. The goal of this classifier is to combine the weak classifier and integrate them into a robust model [17], [18]. The decision tree is used for the estimator of a weak classifier in AdaBoost. The advantage of AdaBoost is able to avoid overfitting because the boosting method uses a decision tree with a limited height of the tree, so it can reduce the overfit problems [19]. The decision tree is formed from Gini and information gained to create a branch for decision making [20]. The algorithm of SAMME will be shown below [15].

a. h(x) is weak, and C(x) is strong classifier
b. Initialize the weight *b* and *m* is length
c. For n = 1 to N:
   - Fit classifier h(x) using weight *b*
   - Compute error

- Computer weight
- Update a new weight
- Re-normalize weight

d. Output C(x)

In this phase, grid search is implemented to find the optimum parameter of AdaBoost. Some machine learning algorithms are sensitive to parameter tuning because the model performance is dependent on it [21]. Grid search is an approach technique of hyperparameters optimization by building and evaluating a model based on the combination of parameter [22]. The scenario of grid search by a determined range of hyperparameter into the grid of search space that contains each point that represents a set of parameter [23]. AdaBoost parameters that will be optimized are *n estimators* and *learning rate*. The *n estimator* is used to train the model sequentially, and the *learning rate* is used to determine how the exactitude of the model. The value of the learning rate between 0 to 1.

The last phase is to evaluate the performance of the classifier. Some evaluation metric used in this paper is the confusion matrix. The use of the confusion matrix is to calculate accuracy, precision, sensitivity, F1 score, and ROC. The confusion matrix is formed from true positive (TP), true negative (TN), false positive (FP), and false negative). The illustration of confusion is shown in Figure 4 below.

		Predicted	
		positive	**negative**
Actual	**positive**	True positive (TP)	False negative (FP)
	negative	False negative (FN)	true negative (TN)

Figure 4. Confusion matrix explanation

Accuracy (acc) to know the percentage of correct prediction compared with total data, the formula showed in Equation 2. Precision is the percentage of accurately predicted total positive, the formula shown in Equation 3. Sensitivity also known as the recall is to know the percentage of properly classified among all instances within its class, the formula showed in Equation 4. The F1 score is the harmonic result between precision and recall [24] the formula is shown in Equation 5.

$$acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$precision = \frac{TP}{TP+FP} \quad (3)$$

$$recall = \frac{TP}{TP+FN} \quad (4)$$

$$F1 = \frac{2 \times Precision \times Recall}{precision+recall} \quad (5)$$

3. RESULTS AND DISCUSSION

Appropriate the methodology, data is preprocessed from raw data until the data is ready to fit the AdaBoost model. In this research, the researcher use Orange tools [25] to do the AdaBoost task. AdaBoost implementation using decision tree estimator with SAMME algorithm. The results are considered from evaluation measures, such as accuracy, precision, recall, and F1 score. The first thing to do is to split the dataset into train data and test data based on the determined percentage, which is 80%:20%. The amount of data can be seen in Table 2. Based on the table, there are 336 data for training and 84 data for testing. The data is split and distributed well, for class insomnia and non-insomnia have the same amount of data. It means the data is balanced.

Table 2. Dataset explanation

	training 80%	testing 20%
Insomnia	168	42
non insomnia	168	42
Total	336	84

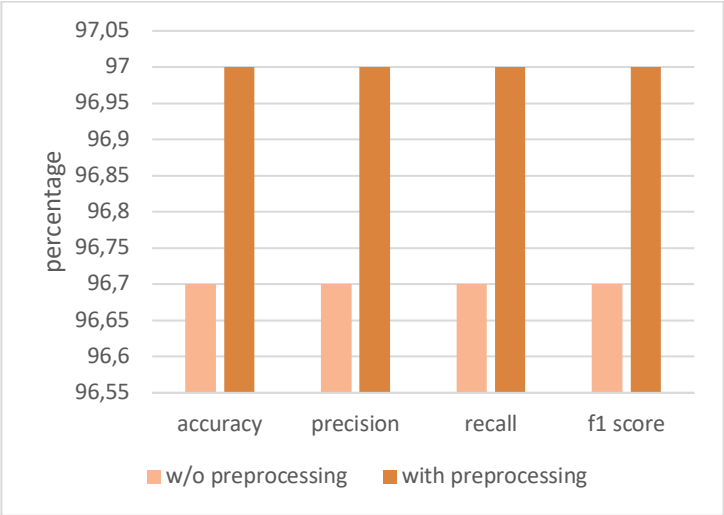

Figure 5. Performance of preprocessed data

The first experiment is comparing between models from preprocessed and without preprocessed data. In this step, a model is created by cross-validation using k-fold = 10. The parameters that we use are default parameters, n_estimators = 50, and learning rate = 1. The result is shown in Figure 5. In the model using raw data without preprocessing using ordinal encoding and normalization, the mean of accuracy, precision, recall, and F1 score is 96.7%. In the model with preprocessed data, the mean evaluation value reaches 97%. It shows if preprocessing using ordinal encoding and normalized dataset can raise up the model performance. It is about 0.3% better than without doing some preprocessing.

The trained model in this experiment gives 97% accuracy, 97% precision, 97% recall, and 97% f1 score. The next experiment is AdaBoost parameter

optimization using the grid search method. In The next experiment, the dataset after preprocessing is used. The parameters that will be optimized are the n estimator and learning rate. The hyperparameter in the search space can be seen in Table 3. The table showed the n estimator is set between 2 and 100 with 2 steps, which means only an even value will be used in the grid search. The result is based on accuracy value, the optimum parameters for n estimators = 76 and learning rate = 0.1

Table 3. Parameters in search space

	n estimators	**learning rate**
search space	2 - 100 (2 step)	0,1 - 0.9 (0.1 step)
optimum parameters	**76**	**0.1**

By fitting parameter tuning obtained from grid search, now the accuracy can reach higher than before. The result is shown in Table 4. Based on the table, using the default parameter (n-estimator = 50, learning rate = 1) only gives a percentage = 97% in the overall performance measure. With optimized parameters with n-estimator = 76 and learning rate = 0.1, it showed can be raised up the model performance. The evaluations are accuracy = 98%, precision = 98%, recall = 98% and f1 score = 98%. It is about 1% higher than the prior model performance and it is proven that parameter optimization can be done in this research.

Table 4. Parameter tuning result

	accuracy	**precision**	**recall**	**f1score**
default param (n estimator = 50, learning rate = 1)	97%	97%	97%	97%
Optimized param (n estimator = 76, learning rate = 0.1	**98%**	**98%**	**98%**	**98%**

Then the model of AdaBoost + grid search will be used to predict the test data. The result will be compared with prior research to justify the better model [2]. Comparison results are shown in Figure 6. Based on the table, the prior study gets the best performance by LogRegression with accuracy = 98%, precision = 98%, recall = 98%, and f1score = 98%. The average performance value is 98%. In this study, AdaBoost + grid search gives a better performance than LogRegression. The average value of this model is 100% on test data. Accuracy = 100%, precision = 100%, recall = 100%, dan f1score = 100%.

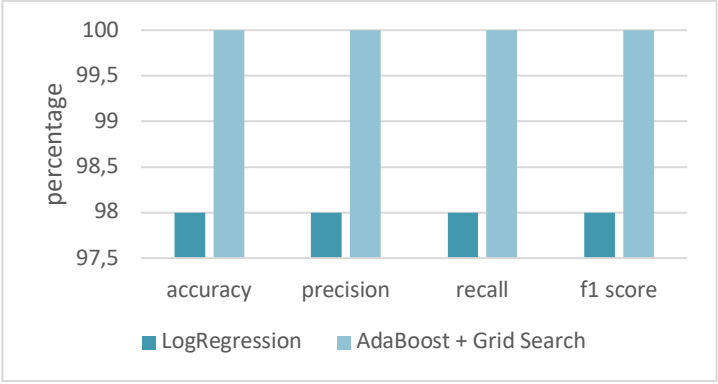

Figure 6. Comparison performance between LogRegression and AdaBoost + grid search

Comparison results between LogRegression and AdaBoost + Grid Search are shown in Figure 6. Based on Figure 6 above, the graphic describes that LogRegression performance doesn't good enough if it's compared with AdaBoost + grid search. The model prediction yields about 2% better than LogRegression. The model of combining AdaBoost with grid search has much good report of accuracy, precision, recall, and f1 score.

4. CONSLUSION

Refers to the result and discussion above, AdaBoost + grid search is proven can do early prediction on insomniacs. Some experiment is done and compared to prior research to get the best model. The first thing to do is experiment with preprocessing data to show if preprocessed data using normalization and ordinal encoding give an impact on the classifier. In this experiment cross-validation with k-fold = 10 is implemented and gives the result of 97% for each accuracy, precision, recall, and f1 score. The parameter tuning is done by grid search. The parameters are n estimator and learning rate. The optimum parameter gives a percentage performance of about 98% for each accuracy, precision, recall, and f1 score with n estimator = 76 and learning rate = 0.1. To prove this model is good, the comparison is needed by comparing AdaBoost + grid search with the LogRegression model from the previous study. The result is LogRegression performance on test data gives 97% on each evaluation measure and AdaBoost + grid search reaches 100% percentage on each accuracy, precision, recall, and f1score.

REFERENCES

[1] T. Roth, "Insomnia: Definition, prevalence, etiology, and consequences," *J. Clin. Sleep Med.*, vol. 3, no. 5 SUPPL., pp. 3–6, 2007, doi: 10.5664/jcsm.26929.

[2] M. M. Islam, A. K. M. Masum, S. Abujar, and S. A. Hossain, "Prediction of chronic Insomnia using Machine Learning Techniques," *2020 11th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2020*, 2020, doi: 10.1109/ICCCNT49239.2020.9225570.

[3] A. Seth, B. Satish Babu, and S. S. Iyenger, "Machine Learning Model for Predicting Insomnia Levels in Indian College Students," *CSITSS 2019 - 2019 4th Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut. Proc.*, pp. 1–6, 2019, doi: 10.1109/CSITSS47250.2019.9031041.

[4] M. Angelova, C. Karmakar, Y. Zhu, S. P. A. Drummond, and J. Ellis, "Automated Method for Detecting Acute Insomnia Using Multi-Night Actigraphy Data," *IEEE Access*, vol. 8, pp. 74413–74422, 2020, doi: 10.1109/ACCESS.2020.2988722.

[5] A. A. bin B. Samsiah Jayos, Lau Ngie Hing, "Insomnia mental health issues pandemic covid-19," *Psycho Idea*, vol. 20, pp. 1–10, 2022, doi: http://dx.doi.org/10.30595/psychoidea.v20i1.10878.

[6] D. Yulia *et al.*, "Insomnia During COVID-19 Pandemic," *J. Ilm. Kesehat. Sandi Husada*, vol. 9, no. 2, pp. 1111–1116, 2020, doi: 10.35816/jiskh.v10i2.483.

[7] M. D. Manzar *et al.*, "Insomnia symptoms and their association with anxiety and poor sleep hygiene practices among ethiopian university students," *Nat. Sci. Sleep*, vol. 12, pp. 575–582, 2020, doi: 10.2147/NSS.S246994.

[8] Y. Sincihu and N. Kurniawati, "Insomnia is not a Risk Factor for Impaired Cognitive Function in Elderly," *Qanun Med. - Med. J. Fac. Med. Muhammadiyah Surabaya*, vol. 2, no. 2, 2018, doi: 10.30651/jqm.v2i2.1446.

[9] S. S. Kim, "Recent Trends of Artificial Intelligence and Machine Learning for Insomnia Research," *Chronobiol. Med.*, vol. 3, no. 1, pp. 16–19, 2021, doi:

https://doi.org/10.33069/cim.2021.0008.

[10] M. M. I. Siam, "Dataset of Insomniac and normal people," vol. 1. Mendeley Data, Oct. 2021, doi: 10.17632/JR5N4PRGFV.1.

[11] Kedar Potdar, Taher S. Pardawala, and Chinmay D. Pai, "A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers," *Int. J. Comput. Appl.*, vol. 175, no. 4, p. 375, 2017.

[12] V. Gajera, R. Gupta, P. K. Jana, and I. S. Member, "An Effective Multi-Objective Task Scheduling Algorithm using Min-Max Normalization in Cloud Computing," *2016 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol.*, pp. 812–816, 2016, doi: 10.1109/ICATCCT.2016.7912111.

[13] K. A. Widagdo, K. Adi, and R. Gernowo, "Kombinasi Feature Selection Fisher Score dan Principal Component Analysis (PCA) untuk Klasifikasi Cervix Dysplasia," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 3, p. 565, 2020, doi: 10.25126/jtiik.2020702987.

[14] M. Anshori, F. Mahmudy, and A. A. Supianto, "Preprocessing Approach for Tuberculosis DNA Classification using Support Vector Machines (SVM)," *J. Inf. Technol. Comput. Sci.*, vol. 4, no. 3, pp. 233–240, 2019, doi: https://doi.org/10.25126/jitecs.201943113.

[15] H. Singh, D. Gupta, and A. K. S. Kushwaha, "Multiclass Object Recognition and Classification Using Boosting Technique," *2018 9th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2018*, pp. 1–6, 2018, doi: 10.1109/ICCCNT.2018.8493890.

[16] G. Chen, Y. Zhang, and D. Tang, "A noise classification algorithm based on SAMME and BP neural network," *2018 IEEE 3rd Int. Conf. Big Data Anal. ICBDA 2018*, no. 4, pp. 274–278, 2018, doi: 10.1109/ICBDA.2018.8367691.

[17] E. O. Ogunseye, C. A. Adenusi, A. C. Nwanakwaugwu, S. A. Ajagbe, and S. O. Akinola, "Predictive Analysis of Mental Health Conditions Using AdaBoost Algorithm," *ParadigmPlus*, vol. 3, no. 2, pp. 11–26, 2022, doi: 10.55969/paradigmplus.v3n2a2.

[18] M. I. Ramadhani, A. E. Minarno, and E. B. Cahyono, "Vehicle Classification using Haar Cascade Classifier Method in Traffic Surveillance System," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 3, no. 1, pp. 57–64, 2017, doi: 10.22219/kinetik.v3i1.546.

[19] V. Chouvatut, C. Yotsombat, R. Sriwichai, and W. Jindaluang, "Multi-view hand detection applying viola-jones framework using SAMME AdaBoost," *Proc. 2015-7th Int. Conf. Knowl. Smart Technol. KST 2015*, pp. 30–35, 2015, doi: 10.1109/KST.2015.7051476.

[20] M. Anshori, F. Mar'i, and F. A. Bachtiar, "Comparison of Machine Learning Methods for Android Malicious Software Classification based on System Call," *Proc. 2019 4th Int. Conf. Sustain. Inf. Eng. Technol. SIET 2019*, pp. 343–348, 2019, doi: 10.1109/SIET48054.2019.8985998.

[21] P. Liashchynskyi and P. Liashchynskyi, "Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS," no. 2017, pp. 1–11, 2019, doi: 10.48550/arXiv.1912.06059.

[22] G. S. K. Ranjan, A. Kumar Verma, and S. Radhika, "K-Nearest Neighbors and Grid Search CV Based Real Time Fault Monitoring System for Industries," *2019 IEEE 5th Int. Conf. Converg. Technol. I2CT 2019*, pp. 9–13, 2019, doi: 10.1109/I2CT45611.2019.9033691.

[23] Y. Sun, S. Ding, Z. Zhang, and W. Jia, "An improved grid search algorithm to optimize SVR for prediction," *Soft Comput.*, vol. 25, no. 7, pp. 5633–5644, 2021, doi: 10.1007/s00500-020-05560-w.

[24] S. Gupta and M. K. Gupta, "Computational Prediction of Cervical Cancer Diagnosis Using Ensemble-Based Classification Algorithm," *Comput. J.*, vol. 65, no. 6, pp. 1527–1539, 2022, doi: 10.1093/comjnl/bxaa198.

[25] Z. B. Demsar J, Curk T, Erjavec A, Gorup C, Hocevar T, Milutinovic M, Mozina M, Polajnar M, Toplak M, Staric A, Stajdohar M, Umek L, Zagar L, Zbontar J, Zitnik M, "Orange: Data Mining Toolbox in Python," *J. Mach. Learn. Res.*, vol. 14, pp. 2349–2353, 2013, [Online]. Available: http://jmlr.org/papers/v14/demsar13a.html.