

## Research Article

# Cuckoo Search-based SVM (CS-SVM) Model for Real-Time Indoor Position Estimation in IoT Networks

**Amjad Khan,<sup>1</sup> Asfandyar Khan,<sup>1</sup> Javed Iqbal Bangash ,<sup>1</sup> Fazli Subhan,<sup>2</sup> Abdullah Khan ,<sup>1</sup> Atif Khan ,<sup>3</sup> M. Irfan Uddin,<sup>4</sup> and Marwan Mahmoud <sup>5</sup>**

<sup>1</sup>*Institute of Computer Sciences and Information Technology (ICS/IT), The University of Agriculture, Peshawar 25000, Pakistan*

<sup>2</sup>*Department of Computer Science, National University of Modern Languages (NUML), Islamabad 44000, Pakistan*

<sup>3</sup>*Department of Computer Science, Islamia College Peshawar, Peshawar 25000, Pakistan*

<sup>4</sup>*Institute of Computing, Kohat University of Science and Technology, Kohat, Pakistan*

<sup>5</sup>*Faculty of Applied Studies, King Abdulaziz University, Jeddah, Saudi Arabia*

Correspondence should be addressed to Atif Khan; [atifkhan@icp.edu.pk](mailto:atifkhan@icp.edu.pk)

Received 13 December 2020; Revised 13 January 2021; Accepted 8 March 2021; Published 19 March 2021

Academic Editor: Shah Nazir

Copyright © 2021 Amjad Khan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Internet of Things (IoT), an emerging technology, is becoming an essential part of today's world. Machine learning (ML) algorithms play an important role in various applications of IoT. For decades, the location information has been extremely useful for humans to navigate both in outdoor and indoor environments. Wi-Fi access point-based indoor positioning systems get more popularity, as it avoids extra calibration expenses. The fingerprinting technique is preferred in an indoor environment as it does not require a signal's Line of Sight (LoS). It consists of two phases: offline and online phase. In the offline phase, the Wi-Fi RSSI radio map of the site is stored in a database, and in the online phase, the object is localized using the offline database. To avoid the radio map construction which is expensive in terms of labor, time, and cost, machine learning techniques may be used. In this research work, we proposed a hybrid technique using Cuckoo Search-based Support Vector Machine (CS-SVM) for real-time position estimation. Cuckoo search is a nature-inspired optimization algorithm, which solves the problem of slow convergence rate and local minima of other similar algorithms. Wi-Fi RSSI fingerprint dataset of UCI repository having seven classes is used for simulation purposes. The dataset is preprocessed by min-max normalization to increase accuracy and reduce computational speed. The proposed model is simulated using MATLAB and evaluated in terms of accuracy, precision, and recall with K-nearest neighbor (KNN) and support vector machine (SVM). Moreover, the simulation results show that the proposed model achieves high accuracy of 99.87%.

## 1. Introduction

Internet of Things (IoT) is an emerging technology that provides different devices to interconnect and communicate with each other. IoT is becoming an important part of today's world due to its rapid growth. Moreover, the use of machine learning (ML) algorithms in various applications of IoT has attracted researchers from all over the world. For a very long time, location has been extremely useful for humans to navigate outdoor over the sea, air, and land using astrolabe, sextant, and octant to determine their location with respect to various celestial bodies [1]. In the 20th

century, with the advancement in electronics and communication, new technologies are adapted such as Radio Detection and Ranging (RADAR), Long Range Navigation (LORAN), and Global Positioning System (GPS) for localization [1]. GPS remains one of the most dominant technologies among the available technologies to localize an object. It only shows better performance to localize object outdoor and fails to estimate the position of object indoor with acceptable accuracy. Now, people are spending most of their time in an indoor environment, thus needing the positioning system to trace people and objects in the indoor complex environment. Therefore, many applications have

been arised, which need location information such as location detection of products in a warehouse, location detection of personal in hospitals, and localizing fireman in a building.

To estimate the position of an object either outdoor or indoor, the most usable and powerful technique used was the global positioning system (GPS). GPS estimates the location by measuring the distance between a GPS satellite and a base station using LoS. The GPS-based location estimation techniques fail to achieve high accuracy due to high signal loss inside a complex indoor environment as GPS signals cannot penetrate the walls of buildings and other obstacles [2]. Due to the technological advancement, many other signal-based possibilities have been raised such as camera, sound, infrared, Radio Frequency Identification (RFID) and Bluetooth Tags, and Wi-Fi [2]. Among all these Wi-Fi received more attention from the research community because, in most cases, the site is already calibrated with Wi-Fi routers, which obsolete extra calibration charges and time [3]. Different techniques are developed, such as Triangulation, Trilateration, Proximity, and Fingerprinting, using Angle of Arrival (AoA), Time difference of Arrival (TDoA), Time of Arrival (ToA), and Receive Signal Strength Identification (RSSI) [1]. All these techniques except fingerprinting require LoS, which is not possible in an indoor environment which makes fingerprinting the most reasonable technique for indoor localization [4]. On the contrary, fingerprinting is laborious and time-consuming and the radio map is venerable to environmental changes, leading to high position estimation error. Machine learning-based models are introduced to automate, generalize, and reduce estimation error [5].

Many machine learning algorithms such as support vector machine (SVM), K-nearest neighbor (KNN), extreme learning model (ELM), decision tree (DT), Naive Bayes (NB), and Bayesian Network (BN) were used for location estimation in an indoor environment. The results show that KNN and SVM are outperformers [6, 7] as compared to others. Moreover, SVM is based on the structural risk minimization principle with good generalization ability and can better solve problems with few samples, nonlinear data, avoid local minima, and so on [2]. For high classification accuracy or position estimation machine learning models, SVM depends on their parameter optimization. Therefore, nature-inspired optimization algorithms such as particle swarm, bee, bad reference distribution and cuckoo search can be used [8].

Cuckoo is one of the most recent algorithms inspired by breeding phenomena of the cuckoo bird, which are used to solve the nonlinear optimization problem. Other optimization algorithms have limitations in terms of convergence to the current or local best solution. They may fail to solve the nonlinear optimization or multidimensional optimization problem. In the case of cuckoo search, combining local and search capability increases the probability of global optimal solution using Levy's flight process [9].

In this study, we propose a cuckoo search-based support vector machine (CS-SVM) model for position estimation in an indoor complex environment. Inspired by many state-of-

the-art optimization-based machine learning models, we used a state-of-the-art dataset of the well-known UCI repository, which is the same as in [10], to evaluate its performance. The proposed model is evaluated in terms of accuracy, precision, and recall with KNN and SVM using MATLAB. The KNN and SVM stay good performers achieving room level accuracy up to 98.7% and 98.3%, respectively, while the proposed model achieves high accuracy up to 99.7%.

In Section 1, we elaborated the literature study, and then, in Section 2, the ingredients of the proposed model are discussed along with the proposed model, and in Section 3, results of the proposed model are justified with benchmark results. Section 4 concludes the research article.

## 2. Related Work

In [11], the authors conducted a survey regarding localization techniques, mentioned that GPS and cellular networks are outdoor localization sources, and they failed to localize anything indoor because of the deep shadowing effect. In [12], Subhan et al. proposed an extended gradient predictor and filter to reduce variation in RSSI values. The RSSI values get variation due to various factors such as walls, obstacles, human crowd, and temperature. The results show better performance than the KALMAN filter. On the contrary, Suining and Chan [3] proposed a fingerprinting technique and used Wi-Fi RSSI values to reduce the extra calibration expenses.

Recently, Wi-Fi RSSI is used to estimate the position of an object in an indoor environment. In [4], Wi-Fi-based approach is proposed using two architectures: client server and standalone. It uses the existing infrastructure of an indoor environment and compares offline fingerprint RSSI measurement with an online RSSI fingerprint to estimate the location of the user. A combination of Wi-Fi and Bluetooth radio technology-based approach is proposed in [13]. It uses KNN with particle filter and shows that indoor estimation error changes by changing the target area. Hossain and Soh [5] highlights that Wi-Fi fingerprinting is laborious and time-consuming, and radio maps are vulnerable to environmental changes. In [14], the authors proposed a multi-dimensional particle filter (MPF) algorithm to estimate the direction of an indoor object. The scheme in [15] is based on Bluetooth technology and uses a machine learning approach to automate the fingerprinting technique. The RSSI variations are smoothed using the filtering algorithm to achieve high accuracy. In [16], the authors presented a learning regression-based filter tracking system using RSSI matrices. It concludes that the particle filter is efficient for the loud and complex indoor environment but expensive than the KALMAN filter.

In [6], the authors compare various machine learning algorithms such as KNN, SVM, NB, BN, DT, and SMO and ensemble algorithms such as Bagging and AdaBoost using fingerprinting technique. The simulation results show that KNN is the best of all. Similarly, Sabanci et al. [7] also compare different machine learning algorithms such as ANN, KNN, ELM, SVM, NB, and DT based on Wi-Fi

fingerprinting. According to the simulation results, the KNN shows the best performance. In [17], the KALMAN filter is used to smooth RSSI values coming from Bluetooth beacon and compare KNN, SVM, and random forest. In [18], it is stated that fingerprinting map changes with change in the environment which leads to high positioning estimation error. For this purpose, KNN, SVM, and DT techniques are used.

Other researchers have also used machine learning algorithms such as KNN and WNN [19], ELM [20], SVM [7], and SVM and DT [21] for position estimation in an indoor environment. According to the literature two machine learning algorithms, both K-NN and SVM show better performance against the other learning model. Compared to SVM, KNN slightly shows good accuracy results in the literature cited.

**2.1. Proposed Methodology.** In the following sections, the components of the proposed model (CS-SVM), i.e., support vector machine (SVM) and cuckoo Search along with the proposed CS-based SVM, are discussed.

**2.2. Cuckoo Search Algorithm.** The cuckoo search algorithm [9] was developed inspired by the breeding process of cuckoo birds. Recently, gaining more attention and becoming very popular over other optimization algorithms such as particle swarm, bat, and hill climbing, these algorithms are also nature-inspired, but they have the limitation of converging to the local or current best solution. So, they are lack in their performance for a nonlinear and multidimensional optimization problem. On the contrary, cuckoo search adopts a different strategy to best fit for the multidimensional and nonlinear problem. It uses Levy's flight process, where the selection of local best through searching capability gives the high chance of global optimal solution. The cuckoo search working process is discussed in the following steps.

At a time, a cuckoo lays only one egg, and the eggs are placed in a nest, which is selected randomly. In a nest, the different eggs represent different solutions, while the new solution is represented by the cuckoo egg.

The nests having high-quality eggs are the best nests, which will be passed to the next generation. The best solutions are represented by these best nests.

There is a fixed number of host nests that are available.  $P \in (0, 1)$  is the probability which represents that the egg laid by a cuckoo is revealed by the host bird. Accordingly, Levy flight mechanism equation (1) is used to estimate the updated nest position of the cuckoo:

$$X_i^{t+1} = X_i^{(t)} + \partial \otimes L(\lambda), \quad i = 1, 2, \dots, n. \quad (1)$$

In equation (1),  $X_i^{(t)}$  represents the position of the nest,  $X_i^{t+1}$  is the new nest position,  $\partial$  is the control value, and  $\otimes$  represents point to point multiplication of the  $L(\lambda)$  Levy flights' process. After updating, the position random value is generated, where  $r \in (0, 1)$ .

If  $r > P$ , new position  $X_i^{t+1}$  changes randomly or it remains in the same position, and the better nest with the new position  $X_i^{t+1}$  is kept for the next generation.

**2.3. Support Vector Machine.** Support vector machine (SVM) can be used for classification as well as a regression problem. It works on the principle of structural risk minimization (SRM). It balances the linear separable space data into nonlinear separable feature space. Equation (2) gives the linearly separable sample set for the binary classification problem:

$$(x_i, y_i), \quad i = 1, 2, \dots, n, \quad x \in R^d. \quad (2)$$

In case  $x_i$  belongs to the first class, then it is denoted by  $y_i = 1$ , while in case  $x_i$  is belongs to the second class, then it is denoted by  $y_i = -1$ .

Here, the division line called hyperplane classifies two classes without error, a margin line which specifies class boundary and the distance between the margin lines of two opposite classes called class interval or marginal distance and the data point of either class. The line which is nearest to the hyperplane is known as the support vector. In the high-dimensional space, the marginal distance makes the hyperplane more optimal which results in the optimal division line into the optimal division plane. Radial basis function (RBF), sigmoid kernel, polynomial kernel, and linear kernel are the commonly used kernels. Considering practical applications, the classification problems belong to multiclass category problems. The indoor positioning problem with multiple class dataset belongs to multiclass category problems. Therefore, the establishment of an SVM multiclassifier is required. Directed acyclic graph, one-versus-all, and one-versus-one are multiclassification methods.

**2.4. Cuckoo Search-Based SVM (CS-SVM).** Appropriate parameter selection is very important as both the generalization and learning performance ability of SVM depend on it. Moreover, the perdition ability of the model and its precision has a direct relation with the appropriate parameter selection. Therefore, the SVM parameters can be optimized using different methods such as grid search method, genetic algorithm, and particle swarm optimization algorithm. Both the genetic and particle swarm optimization algorithms face the problem of local extremes. On the contrary, the grid search method is time-consuming as over the hyperparameter space an exhaustive search is required. Recently, the cuckoo search (CS) algorithm is proposed which is a metaheuristic algorithm. It has a strong ability to global search, requires fewer parameters, and has a good search path. To solve those problems, having multiobjective is a powerful tool. The flow of the proposed is shown in Figure 1.

The performance of the SVM classifier is dependent mainly on the kernel parameter  $\sigma$  and the penalty factor  $C$ . The following steps are used to optimize the SVM parameters, which are also shown in Figure 2:

- (1) Training dataset is selected to train the SVM.

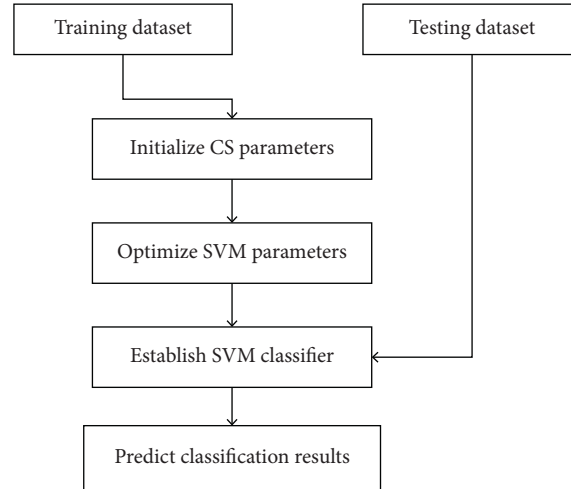


FIGURE 1: Architecture of the proposed model.

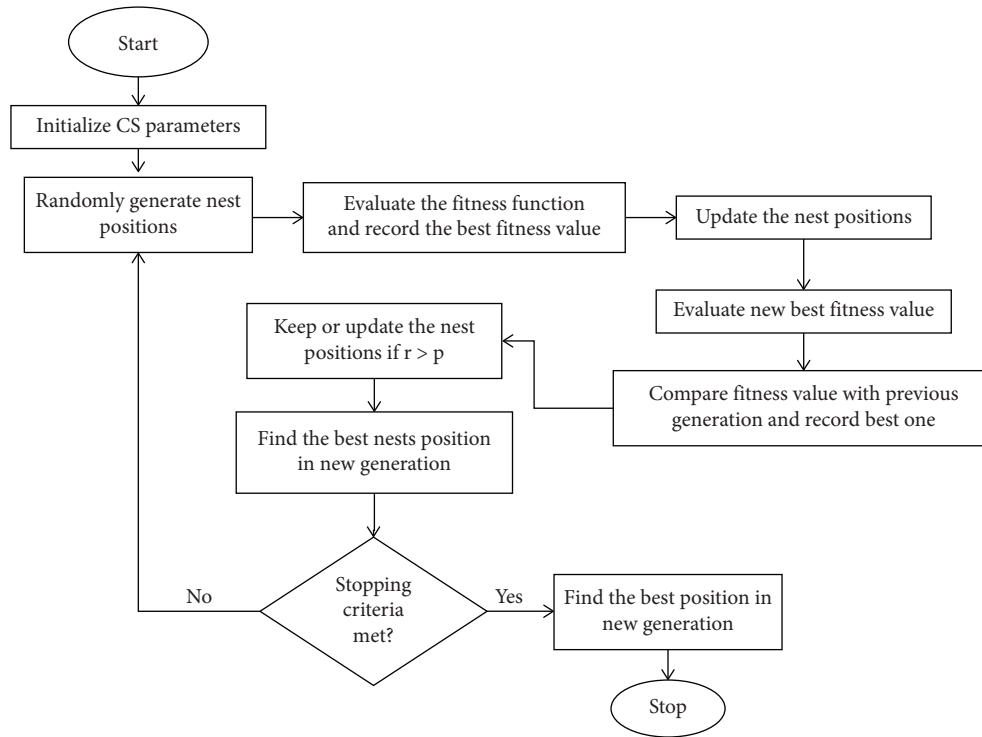


FIGURE 2: Flow diagram of the cuckoo search-based SVM (CS-SVM) model.

- (2) The CS parameters such as probability “P,” no. of nests “n,” the number of iteration, and SVM parameter ranges are initialized.
- (3) Using solution space the initial population of ‘n’ host nests is generated randomly using

$$P_{t-1} = [x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}]^T. \quad (3)$$

After that, the eggs are placed there and the group of parameters  $(C, \sigma)$  represents the nest position.

- (4) The qualities of the group of parameters  $(C, \sigma)$  which represents the nest positions (fitness functions) are evaluated to determine the current best nest  $x_b^{(0)}$  (fitness value) and carry over it to the next generation.
- (5) Equation (1) is used to update the positions of all other nests, and the qualities of the nest positions are evaluated belonging to the new group.
- (6) The nest positions of this new group are compared with the last group using

$$p_{t-1} = [x_1^{(t-1)}, x_2^{(t-1)}, \dots, x_n^{(t-1)}]^T. \quad (4)$$

Once the comparison is done, the group having the worse nest positions are replaced with the group having the better nest positions to get a group of better nest positions (fitness value) using

$$K_t = [x_1^{(t-1)}, x_2^{(t-1)}, \dots, x_n^{(t-1)}]^T. \quad (5)$$

- (7) In case  $r$  (random number) is greater than  $//P, //$  keep the nest positions having low probability in  $k_t$  using equation (5), update the nest positions having a high probability, and evaluate the qualities of the nest positions belonging to the new group. The nest positions belonging to this group are compared with those in  $k_t$ . Once the comparison is done, the group having the worse nest positions is replaced with the group having the better nest positions to get a group of better nest positions (fitness value) using

$$p_t = [x_1^{(t)}, x_2^{(t)}, \dots, x_n^{(t)}]^T. \quad (6)$$

- (8) Determine the best nest position (fitness value)  $x_b^{(t)}$  in  $p_t$  using equation (6).  
 (9) Check whether the number of iterations has reached the threshold level of a number of iterations or the level of a certain precision has been achieved. In case none of the aforementioned conditions is true, go back to step (4) and continue. In case any one of the aforementioned conditions is true, stop searching, and  $x_b^{(t)}$  is the best nest position.  
 (10) SVM parameters  $(C, \sigma)$  correspond to the best nest position  $x_b^{(t)}$ .

### 3. Results and Discussion

Inspired by the many state-of-the-art optimization-based machine learning models, we used a state-of-the-art dataset of the well-known UCI repository which is the same as in [10]. The dataset is preprocessed by min-max normalization to increase accuracy and reduce computational speed. It was divided into 70% training and 30% testing. The proposed model is simulated using MATLAB R 2018 b on Window 8 OS with 4 GB RAM.

Different training and testing experiments were performed on three models, i.e., support vector machine (SVM), K-nearest neighbor (KNN), and cuckoo search-based support vector machine (CS-SVM). These models were evaluated in terms of precision, recall, and accuracy.

In the classification process with KNN, high accuracy values are achieved by optimizing the parameters. In machine learning (ML), the parameters that need to set the algorithm to start are known as hyperparameters.  $k$  and

distance type to be calculated are hyperparameters in the case of KNN. As a result of the hHyperparameter optimization, the best  $k$  value is calculated as 1. The distance type that gives the best correctness is determined as Euclidean. Before training the KNN classifier, we divide the dataset into 70% training and 30% testing using holdout cross-validation. Test the trained KNN model over 600 observations, which are 30% of all total observations. Testing observation is distributed among the four classes. Each class is having 150 observations. The class-wise and average output prediction of the model in terms of precision, recall, and accuracy is shown in Table 1. The average precision, recall, and accuracy values are slightly different which are 0.987, 0.98675, and 0.98675, respectively. This result of the model showed slightly better performance than SVM.

Using an SVM classifier, first of all, the entire multiclass problem is converted into the binary class problems. The binary class problems are solved with binary classifiers, and the solution can be merged to get the solution of the multiclass problem. One-versus-one (OVO) method of SVM is used in such cases. In the OVO method, all possible combinations of the multiple class problems are divided into binary class problems. After that, the classifier is trained for each binary class problem. Then, the outputs of these binary class classifiers are merged to estimate the output multiple class problems. The SVM classification using the OVO method results in the error matrix. Likely, the SVM model tested was over 600 observations, which are 30% of all total observations. Testing observation is distributed among the four classes. Each class is having 150 observations. The class-wise and average output prediction of the model in terms of precision, recall, and accuracy is shown in Table 2. The average precision, recall, and accuracy values are slightly different from each other which are 0.98375, 0.98325, and 0.983, respectively. This result of the model is slightly behind the results of KNN.

From Tables 1 and 2, it is clear that SVM is slightly behind in their results against K-NN in this research experiment. Therefore, the CS-SVM trained model is tested over 600 samples of the dataset which is the same as the simple SVM and KNN. In this testing process, cuckoo search optimizes the parameter of SVM over the 6th iteration; the last iteration results are the final results of the CS-SVM model which are better than those of simple KNN and SVM, and now, SVM takes over the KNN results. The intended precision, recall, and accuracy results of the final iteration are 0.9900, 0.9980, and 0.9967, respectively, as shown in Table 3.

The performance of the proposed CS-SVM model in terms of precision, recall, and accuracy as compared to KNN and SVM is given in Table 4 and Figure 3. From Table 4 and Figure 3, it is clear that the proposed model CS-SVM surpasses the benchmark models.

According to the literature study and our implementation result, KNN gives a slightly better result than

TABLE 1: Precision, recall, and accuracy for KNN.

Classes	Precision	Recall	Accuracy
One	0.98	1.0	1.0
Two	1.0	0.967	0.967
Three	0.968	0.993	0.993
Four	1.0	0.987	0.987
Average	0.987	0.98675	0.98675

TABLE 2: Precision, recall, and accuracy for SVM.

Classes	Precision	Recall	Accuracy
One	1.0	0.993	0.993
Two	0.986	0.96	0.96
Three	0.949	0.987	0.986
Four	1.0	0.993	0.993
Average	0.98375	0.98325	0.983

TABLE 3: Precision, recall, and accuracy for CS-SVM.

Iteration	Precision	Recall	Accuracy
1	0.9950	0.9990	0.9983
2	0.9900	0.9980	0.9967
3	0.9950	0.9990	0.9983
4	0.9900	0.9980	0.9967
5	0.9900	0.9980	0.9967
6	0.9900	0.9980	0.9967

TABLE 4: Comparison of precision, recall, and accuracy of KNN, SVM, and CS-SVM.

Model	Precision	Recall	Accuracy
KNN	0.98375	0.98325	0.983
SVM	0.98375	0.98325	0.983
CS-SVM	0.9900	0.9980	0.9967

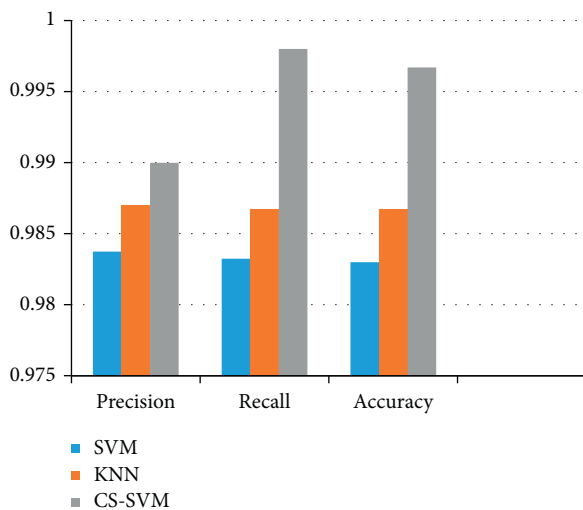


FIGURE 3: Precision, recall, and accuracy comparison of CS-SVM with SVM and KNN.

SVM. Now, by optimizing SVM through a nature-inspired evolutionary cuckoo search algorithm, the SVM improves results over KNN.

## 4. Conclusions

In this research work, we propose the cuckoo search-based support vector machine (CS-SVM) model for position estimation in an indoor complex environment. SVM is based on the structure risk minimization principle with good generalization ability and can better solve problems with few samples, nonlinear data, avoid local minima, and so on. Cuckoo is one of the most recent algorithms inspired by breeding phenomena of the cuckoo bird, which are used to solve the nonlinear optimization problem. Other optimization algorithms have limitations in terms of convergence to the current or local best solution. A state-of-the-art dataset of the well-known UCI repository is used to evaluate the performance of the proposed CS-SVM model. The dataset is composed of the RSSI values of seven Wi-Fi access points collected from four different rooms. The variation in RSSI values of Wi-Fi access point dramatically decreases classification accuracy and effect value of other performance parameters. Furthermore, the formation of fingerprinting RSSI radio map is expensive in terms of labor and time. The proposed model is evaluated in terms of accuracy, precision, and recall with KNN and SVM using MATLAB. The proposed CS-SVM model achieves high accuracy of up to 99.7% as compared to KNN (98.7%) and SVM (98.3%).

## Data Availability

The dataset is available at IndoorIndustrialLocalisationDataset (<https://github.com/vauchey/IndoorIndustrialLocalisationDataset/>).

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

- [1] W. Sakpere, M. Adeyeye-Oshin, and N. B. Mlitwa, "A state-of-the-art survey of indoor positioning and navigation systems and technologies," *South African Computer Journal*, vol. 29, no. 3, pp. 145–197, 2017.
- [2] W. Xue, K. Yu, X. Hua, Q. Li, W. Qiu, and B. Zhou, "APs' virtual positions-based reference point clustering and physical distance-based weighting for indoor wi-fi positioning," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3031–3042, 2018.
- [3] S. He and S. H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: recent advances and comparisons," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490, 2015.
- [4] W. K. Zegeye, S. B. Amsalu, Y. Astatke, and F. Moazzami, "WiFi RSS fingerprinting indoor localization for mobile devices," in *Proceedings of the 2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 1–6, New York, NY, USA, October 2016.

- [5] A. K. M. M. Hossain and W.-S. Soh, "A survey of calibration-free indoor positioning systems," *Computer Communications*, vol. 66, pp. 1–13, 2015.
- [6] S. S. Mohar, S. Goyal, and R. Kaur, "A survey of localization in wireless sensor network using optimization techniques," in *Proceedings of the 2018 4th IEEE International Conference On Computing Communication And Automation (ICCCA)*, pp. 1–6, Greater Noida, India, December 2018.
- [7] K. Sabanci, E. Yigit, D. Ustun, A. Toktas, and M. F. Aslan, "Wifi based indoor localization: application and comparison of machine learning algorithms," in *Proceedings of the 2018 IEEE XXIIIrd International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED)*, pp. 246–251, Tbilisi, Georgia, September 2018.
- [8] J. Minlan, L. Jingyuan, and Z. Xiaokang, "Research on algorithm of three-dimensional wireless sensor networks node localization," *Journal of Sensors*, vol. 2016, Article ID 2745109, 9 pages, 2016.
- [9] X.-S. Yang and S. Deb, "Cuckoo search: recent advances and applications," *Neural Computing and Applications*, vol. 24, no. 1, pp. 169–174, 2014.
- [10] M. A. E. Aziz and A. E. Hassanien, "Modified cuckoo search algorithm with rough sets for feature selection," *Neural Computing and Applications*, vol. 29, no. 4, pp. 925–934, 2018.
- [11] A. Yassin, Y. Nasser, M. Awad et al., "Recent advances in indoor localization: a survey on theoretical approaches and applications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1327–1346, 2016.
- [12] F. Subhan, S. Ahmed, and K. Ashraf, "Extended gradient predictor and filter for smoothing RSSI," in *Proceedings of the 16th IEEE International Conference on Advanced Communication Technology*, pp. 1198–1202, Pyeong Chang, February 2014.
- [13] R. Bruha and P. Kriz, "Different approaches to indoor localization based on Bluetooth low energy beacons and wi-fi," in *Proceedings of the International Conference on Computational Collective Intelligence*, pp. 305–314, Nicosia, Cyprus, September 2017.
- [14] L. Pei, D. Liu, D. Zou, R. Lee Fook Choy, Y. Chen, and Z. He, "Optimal heading estimation based multidimensional particle filter for pedestrian indoor positioning," *IEEE Access*, vol. 6, pp. 49705–49720, 2018.
- [15] P. Sthapit, H. S. Gang, and J. Y. Pyun, "Bluetooth based indoor positioning using machine learning algorithms," in *Proceedings of the 2018 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pp. 206–212, Jeju, June 2018.
- [16] A. Xiao, R. Chen, D. Li, Y. Chen, and D. Wu, "An indoor positioning system based on static objects in large indoor scenes by using smartphone cameras," *Sensors*, vol. 18, no. 7, pp. 1–17, 2018.
- [17] J. Y. Hsieh, C. H. Fan, J. Z. Liao, J. Y. Hsu, and H. Chen, "Study on the Application of Indoor Positioning Based on Low Power Bluetooth Device Combined with Kalman Filter and Machine Learning," *EasyChair Preprint*, pp. 1–9, 2019.
- [18] S. P. Rana, J. Prieto, M. Dey, S. Dudley, and J. M. Corchado, "A self regulating and crowdsourced indoor positioning system through wi-fi fingerprinting for multi storey building," *Sensors*, vol. 18, no. 11, pp. 1–15, 2018.
- [19] A. K. Taskan, "Precise Indoor Positioning Using Bluetooth Low Energy (BLE) Technology," *Middle East Technical University*, vol. 21, no. 3, 971 pages, 2019.
- [20] X. Jiang, Y. Chen, J. Liu, Y. Gu, and L. Hu, "FSELM: fusion semi-supervised extreme learning machine for indoor localization with Wi-Fi and Bluetooth fingerprints," *Soft Computing*, vol. 22, no. 11, pp. 3621–3635, 2018.
- [21] G. Li, E. Geng, Z. Ye, Y. Xu, and H. Zhu, "An indoor positioning algorithm based on RSSI real-time correction," in *Proceedings of the 2018 14th IEEE International Conference on Signal Processing (ICSP)*, pp. 129–133, Beijing, China, August 2018.