Optimization of Sanctioned Load in Smart Homes using Hybrid Grey Wolf-Bat Algorithm

[1]Mohit Kumar, [2]Rakesh Yadav, [3]Shivam Manwal, [4]Jyoti Kaushik

[1,2,3]Student, [4]Assistant Professor,
Department of Computer Science and Engineering,
Maharaja Agrasen Institute of Technology, New Delhi, India.

Abstract: This paper offers the optimized value of sanctioned load/power allocation to smart homes that are having Smart Grid Infrastructure. Current methods allocate load in different slabs, but not all houses utilize the entire slab. In this paper, the load allocation values were predicted based on the previous year's consumption data. The dataset was taken from Kaggle and it contains electricity consumption data along with weather information of a smart home having 14 different appliances. Three metaheuristic algorithms viz. Grey Wolf Optimizer (GWO), Bat Algorithm (BA), and Hybrid Grey Wolf-Bat Algorithm (GWOBA) were used to optimize the parameters of the regression model created using Support Vector Regression (SVR). The results were studied for different temperature slabs. Accuracy for different algorithms was calculated by using cross-validation and it was found that GWOBA performed significantly better than the other two algorithms. It was observed that for GWOBA, the maximum accuracy gain observed was 12.1% whereas the maximum accuracy drop was 2% only. It was concluded that among all three algorithms, GWOBA is suitable for this problem.

Keywords: **Optimization, Metaheuristic algorithms, Support Vector Regression, Grey Wolf Optimizer, Bat Algorithm, Hybrid Grey Wolf-Bat Algorithm.**

I. INTRODUCTION

Energy has been a global concern around the world. With non-renewable resources being used primarily for fulfilling the requirements of 21st century needs and not enough alternatives to bid on, hence the efforts have to be made to minimize the wastage so that we can buy ourselves time and utilize it to come up with solutions and alternatives. The term 'Energy Crisis' came into existence in the 1970s caused by the peaking of oil production in major industrial nations (United States, Canada, Germany, etc.) and an official order was released to stop doing business with other producers. Since then, countries like California, China, Nepal have experienced an energy crisis that has affected their social and economic status. The problem got the attention and international bodies like the 'World Energy Council' got created who look for global energy problems.

India is the third-largest producer as well as consumer of electricity. During the year 2019-20, India's power generation by the Central Sector, State Sector, Private utilities & IPPs was about 1250783.91 million units, out of which 1095014.79 million units were generated using non-renewable resources [1]. India's high dependency on non-renewable resources is a reason for concern because the electricity demand is increasing every year.

To prevent an energy crisis, we must consume less energy and also reduce wastage by improving and modernizing the energy infrastructure. One such modern energy infrastructure technique is 'Smart Grid Technology'. A Smart Grid is an electricity network that provides two-way communication of electricity as well as information so it can intelligently integrate consumers and generators to efficiently deliver sustainable and secure electricity supplies [2], [3]. To transform the conventional grid system of India, National Smart Grid Mission (NSGM) was started under the supervision of the Ministry of Power, Government of India [4]. The goal of this mission is to implement the Smart Grid Infrastructure by 2025.

The building block for Smart Grid implementation is Advanced Metering Infrastructure (AMI). It consists of 'Smart Meters' that facilitate two-way communication between generators and consumers. A smart meter is an electronic device that monitors and records electricity-related information such as consumption, weather conditions, voltage levels, etc. of a particular home at a regular interval, generally of one minute. The information recorded by the smart meter can be transferred to the generators using the smart grid. Generators can then extract insights from this information and use it to tackle energy-related issues.

The sanctioned power load given to a particular house is the maximum power that they can use. It is generally allocated in slabs like 0 - 2KW. But not all houses in a locality have similar electricity usage. Once a household exceeds the sanctioned load, they have to switch to a new slab even if it exceeds by a very small margin. A better approach to allocate the power load would be by predicting the optimum values from the previous year's consumption data.

The electricity consumption in smart homes can be considered as a regression analysis problem where the predicted optimal values obtained can lead to a better allocation of power load, resulting in efficient load balancing, cheaper electricity bills, and will help to eliminate the Energy Crisis of the world.

II. LITERATURE REVIEW

Optimization is considered to be an important topic in Applied Mathematics because it has been applied to a variety of disciplines ranging from computer science and engineering to operations research, to economics to operations. Optimization is not only limited to science and mathematics but also has application in daily life tasks. Optimization is about finding a suitable value that follows certain constraints. Talbi [5] has broadly classified optimization methods into Exact and Approximate methods. The exact method gives the exact optimal solution for a problem. It includes techniques like Dynamic Programming, Greedy Algorithms, etc. The disadvantage of the Exact methods is that when the dimension of the problem increases, they fail to give the solution in time. In a real-world scenario, most problems are of high dimension with more complex constraints, hence Approximate methods are used. The Approximate methods give a suitable solution to a problem in a reasonable time.

A metaheuristic is an Approximate method of optimization. The term Metaheuristics was first introduced by Fred Glover [6]. According to Glover, a metaheuristic is a high-level algorithm or algorithmic framework that is independent of a problem and provides a heuristic for optimization. Metaheuristics can be classified into nature-inspired and non-nature-inspired. The majority of algorithms are nature-inspired, for example, Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO) [7], Bat Algorithm (BA) [8], etc. Some non-nature-inspired algorithms are Iterative Local Search (ILS) [9], Tabu Search (TS) [10], etc.

Another broad classification of metaheuristics is population-based and single-point search algorithms. In population-based algorithms, the optimization is driven by a population of solutions. GWO, BA, PSO are some of the population-based metaheuristic algorithms. Whereas in single-point search, the optimization process is driven by a single solution. TS and ILS are few examples of single-point search algorithms [11]. The algorithms that are utilized in this paper are population-based algorithms.

Deepak Gupta et al. [12] used PSO and cuckoo Algorithms to optimize electricity consumption. They concluded that the average value obtained from both the algorithms were similar but PSO's result has a broader range of value whereas cuckoo's result has a smaller range of values. In their extended work [13], they find out the accuracy of both PSO and Cuckoo by using Grey Wolf Optimizer as reference. They concluded that accuracy obtained from PSO and Cuckoo is 77.95% and 76.58% respectively. In this paper, we have applied Hybrid Grey-Wolf Bat Algorithm (GWOBA) to optimize sanction load. M. ElGayyar et al. [14] has proposed this algorithm and tested it on the CEC2017 benchmark suite. They concluded that the algorithm shows significant improvement against other algorithms used in their research work. The hybrid algorithm has advantages of both the traditional Grey Wolf Algorithm (GWO) [7] and Bat Algorithm [8]. Since both these algorithms are meta-heuristic algorithms, they were able to come up with a hybrid version. We have chosen this algorithm because it performs better on benchmarks. Secondly, even when it fails to achieve the best result, it loses with a very low order of magnitude. So, in our task to model electricity consumption data, this will make sure that, when deployed in the real environment, if it fails to achieve optimal results, the margin of loss will still be minimized.

III. BACKGROUND WORK

1. Support Vector Machines for Regression:

Support Vector Regression (SVR) is a supervised machine learning algorithm for regression analysis. It is similar to Support Vector Machine (SVM) in working. SVR's objective is to find a hyperplane to best represent the model. It achieves this by fitting a line and keeping all the support vectors within a threshold value. SVR is used in real-world complex problems because it supports non-linear regression as well.

To effectively model non-linear data, SVR makes use of kernels. Kernels help in transforming lower dimension data to a higher dimension so that it can perform the linear separation. Some of the available kernels for SVR are Linear, Polynomial, Sigmoid, and RBF. In this paper, we have used the Radial Basis Function kernel (RBF).

RBF is the default and most widely preferred kernel because it is very similar to the Gaussian distribution. Another reason for its popularity is that it is easier to tune.

RBF is mathematically represented as:

$$K_G(x, x') = exp\left(-\frac{d(x, x')^2}{2\sigma^2}\right)$$

Where $d(x, x')^2$ is Euclidian distance between points x and x' and σ is the variance of the hyperparameters [15].

Parameters of RBF:

The reason behind the popularity and preference for the RBF kernel over other kernels is its parameters. The parameters of RBF are tuned to enhance the accuracy of SVR. The most basic way of tuning parameters is grid search. In this paper, we will be using the population-based metaheuristics algorithm to achieve better accuracy.

There are two important parameters for RBF viz., the C and Gamma parameters. The optimal values of these parameters will help our model to find a suitable trade-off between bias and variance.

The parameters are as follows:

i. C Parameter:

The C parameter is also known as the Regularization parameter. It tries to regularize the task of minimizing the threshold of decision boundary by penalizing each misclassification. The value of C is directly proportional to the penalty it imposed. The lower the value of C, the less will be the penalty, hence the model will have a decision boundary with a large margin. When the value of C is large, the penalty will be large, hence the decision boundary of the model will be with a smaller margin.

ii. Gamma Parameter:

The RBF kernel works on the measure of similarity [15]. When data is not linearly separable, it does not transform the original data to a higher dimension as it is a costly operation, so it computes the measure of similarity of higher dimension and based on that finds decision boundary. The gamma parameter helps in finding similarity as it controls the influence of a single point on support vectors. If the value gamma is small, the influence of outlier points will be large and the model will not be able to linearly separate the data. If the value of gamma is large, the influence of outliers will be small and the model will overfit to training examples.

Hence, a good pair of C and Gamma values will perform the trade-off between bias and variance resulting in a highly accurate model for regression analysis.

2. Normalization of data:

Machine learning algorithms learn from the data that is provided to them. Some ML algorithms face difficulty in mapping input features to output if the data being provided is not in a particular range. Usually, algorithms that require distance calculation are susceptible to this problem. SVR internally computes support vectors using Euclidian distance, hence all features must be normalized.

Normalization is the process of transforming the original range of features to range [0, 1]. The formula used to normalize the data is as follows:

$$X' = \frac{X - min}{max - min}$$

The dataset we have used contains features like temperature [°f], furnace [kW], etc. whose unit are different hence it's important to normalize the dataset before using. We have used the scikit-learn's MinMaxScaler [16] to achieve the normalization.

3. Grey Wolf Optimizer:

3.1. Introduction:

Grey Wolf optimization algorithm is a population-based stochastic algorithm inspired by the hunting mechanism of Grey Wolves and their social hierarchy. It was developed by Seyedali Mirjalili et al. [7] in 2014. There are four types of wolves in a pack viz. Alpha (α), Beta (β), Delta (δ), and Omega (ω).

Alpha wolves are the head wolves of the pack and are responsible for all decisions. Alpha wolves might not be the strongest in terms of strength but are the best in managing the pack. Beta wolves are the immediate subordinate of alpha wolves. They are considered to be the ones who can take the responsibilities and position after Alpha wolves. The least positioned wolves in the pack are the Omega's ones. Omega wolves work as a peacemaker because, in case of a fight or mishap, all the blame can be put on Omegas. Those who don't fall in any of these categories are called the Delta wolves. Delta wolves are superior to Omega wolves.

Phases that are involved in the hunting mechanism of Grey Wolves are:

1. Locating the herd of prey. Tracking the most vulnerable prey from the herd.
2. Diverging and encircling the prey until it stops moving.
3. Attacking the prey.

Mathematical representation:

The hunting behaviour of grey wolves is represented in mathematical form by considering the three best solutions. The best solution is the Alpha (α). The second-best solution is called Beta (β) and the third-best is called Delta (δ). The hunting is guided by the cumulative positioning of α, β, and δ.

The encircling behaviour of the mathematical representation is as follows:

$$\vec{D} = \left|\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)\right| \quad (1)$$

$$\vec{X}(t+1) = \vec{X}_P(t) - \vec{A} \cdot \vec{D} \quad (2)$$

Where 't' denotes the current iteration, '$\vec{A}$' and '$\vec{C}$' are coefficient vectors, '$\vec{X}_p$' is the position vector of the prey, and '$\vec{x}$' indicates the position vector of a grey wolf.

The vectors '$\vec{A}$' and '$\vec{C}$' are calculated as:

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r_1} - \vec{a} \tag{3}$$

$$\vec{C} = 2 \cdot \vec{r_2} \tag{4}$$

Where 'a' linearly decreased from 2 to 0 during iterations and '$\vec{r_1}$' and '$\vec{r_2}$' are random vectors in [0,1].

The hunting is achieved when the above equations of encircling behaviour are followed by each wolf. The with cumulative positioning they hunt their prey. This is modelled with the following equations:

$$\vec{D_\alpha} = |\vec{C_1}.\vec{X_\alpha} - \vec{X}|, \ \vec{D_\beta} = |\vec{C_2}.\vec{X_\beta} - \vec{X}|, \ \vec{D_\delta} = |\vec{C_3}.\vec{X_\delta} - \vec{X}| \tag{5}$$

$$\vec{X_1} = \vec{X_\alpha} - \vec{A_1} \cdot \vec{D_\alpha}, \ \vec{X_2} = \vec{X_\beta} - \vec{A_2} \cdot \vec{D_\beta}, \vec{X_3} = \vec{X_\delta} - \vec{A_3} \cdot \vec{D_\delta} \tag{6}$$

$$\vec{X}(t+1) = \frac{\vec{X_1} + \vec{X_2} + \vec{X_3}}{3} \tag{7}$$

3.2. The Algorithm:

Load the Dataset (Train Data, Test Data)

Set parameters:
No. of Search Agents = 20
Dim = 2
ub = 4
lb = 1
maxIter = 20

Initialize the grey wolf population P_i (i = 1, 2………, n)
Initialize a, A, and C

X_α = the best search agent
X_β = the second-best search agent
X_δ = the third-best search agent

For each search agent:
randomly generate a solution in the range (lb, ub)
End for

while (t < maxIter)
for each search agent:
Train SVR on training dataset
Find fitness value (Test accuracy using cross-validation)
Update position of all search agents by using equation 7
End for
Update a, A, C
Update X_α, X_β and X_δ
t = t + 1
End while
return X_α

4. Bat Algorithm:

4.1. Introduction:

The bat algorithm is also a population-based metaheuristic algorithm inspired by the echolocation mechanism of microbats. Microbats is a category of bats that uses a type of sonar called echolocation to locate their prey while avoiding obstacles and navigating through the environment. They emit sound waves of different frequencies and pulse rates and listen back to their echo to get an idea about the position of the prey.

Xin-She Yang [8] developed the bat algorithm with the following rules:

1. The bat uses echolocation to sense the distance, and they also know to differentiate between prey and the surrounding obstacles and environment.

2. Each bat flies with the velocity v_i and position x_i with frequency f and loudness A_0 to search for prey. Then they change the frequency and the rate of pulse emission throughout iteration depending on the proximity of the target.
3. The loudness decreases and the rate of pulse emission increases as the bat reaches closer to its prey.

Mathematical Representation:

The echolocation mechanism is modelled mathematically by considering their position, velocity, and frequencies.

At a particular iteration t, each bat is having a velocity v_i^t, a position x_i^t in a d-dimensional search space. The x_* represents the current best solution.

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (8)$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*)f_i \quad (9)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (10)$$

Where the values $\beta \epsilon [0,1]$ and is a random vector drawn uniformly over the distribution.
Some bats may also move in a different direction:

$$x_i^{t+1} = x_x + \epsilon \cdot \bar{A} \quad (11)$$

$$\bar{A} = \frac{\sum_{i=1}^{n} A_i^t}{n} \quad (12)$$

Where ∈ is a random number generated uniformly from [-1, 1] and $\bar{A}$ is average loudness of the bats.

4.2. The Algorithm:

Load the Dataset (Train Data, Test Data)

Set parameters:
Number of bats = 20
Dim = 2
ub = 4
lb = 1
maxIter = 20

Initialise the Bat population P_i (i = 1, 2………, n)
Initialize A(Loudness), r (rate of pulse) for each bat
Initialize $f_{\min}$, $f_{\max}$, r_0
Initialize F(frequencies), V(velocities) for each bat

For each search agent:
randomly generate a solution in the range (lb, ub)
Train SVR on train dataset
Find fitness value (Test accuracy using cross-validation)
End for
Find the best solution among current solutions
Set X∗
Set $f_{\min}$
while (t < maxIter)
for each search agent (i):
Update frequency using equation (8)
Update velocity using equation (9)
Update position using equation (10)
If rand > r_i^t
Update position using equation (11)
Train SVR on train dataset
Find fitness value (Test accuracy using cross-validation)
If (x_i^t < x^*) and (rand > A_i^t)
Update the position with the new one
Update r_i^t , A_i^t
End for
X∗ = current best
t = t + 1

End while
Return X∗

5. Hybrid Grey Wolf-Bat Algorithm:

5.1. Introduction:

The Hybrid Grey Wolf-Bat Algorithm (GWOBA) is also a population-based metaheuristic algorithm for global optimization problems. It incorporates both the high exploration abilities of GWO and the high exploitative abilities of the Bat algorithm. In GWOBA, GWO is allowed to perform a highly explorative search in the solution space and passes the best two solutions to Bat Algorithm. Bat algorithm then performs a highly exploitative search in the solution space to find the best solution. To achieve this, the traditional algorithms need some modifications.

The GWO is first initialized with a random set of solutions, it then iterates to find the optimal solution. After half iterations, the best two solutions i.e., alpha and beta only are passed to the bat algorithm. The bat algorithm then completes the rest of the half iterations and further optimizes the solution. The GWOBA then returns the best solution.

Mathematical Representation:

In GWOBA, we use only the two best solutions from GWO. This is because if we give the three best solutions to the bat Algorithm, it might diverge from the solution. Thus, to make the bat localize the search, only alpha and beta solutions from GWO are passed to the Bat algorithm.

The modified GWO equations are as follows:

$$D_\alpha = |C_1.X_\alpha - X| , D_\beta = |C_2.X_\beta - X| \quad (13)$$

$$X_1 = X_\alpha - A_1 \cdot D_\alpha , X_2 = X_\beta - A_2 \cdot D_\beta \quad (14)$$

$$X(t+1) = \frac{X_1 + X_2}{2} \quad (15)$$

To enhance the exploration abilities of GWO, we force $|A| > 1$ by decreasing the exploration rate constant a from 2 to 1.

$$a = 2 - \frac{t}{Max_{Iter}} \quad (16)$$

The equation for BA remains the same. The only change required in BA is that the loudness and rate of pulse emission parameters are to be reset so that they can drive the algorithm to the solution in only half of the iterations.

5.2. The Algorithm:

Load the Dataset (Train Data, Test Data)

Set parameters:
No. of Search Agents = 20
Dim = 2
ub = 4
lb = 1
maxIter = 20

Initialize the grey wolf population P_i *(i = 1, 2………, n)*
Initialize a, A, C
Initialize A(Loudness), r (rate of pulse) for each bat
Initialize f_{min} , f_{max} , r_0
Initialize F(frequencies), V(velocities) for each bat

X_α *= the best search agent*
X_β *= the second-best search agent*

For each search agent:
randomly generate a solution in the range (lb, ub)
End for

while (t < maxIter/2)
for each search agent:
Train SVR on train dataset
Find fitness value (Accuracy on test dataset)

Update position of all search agents by using equation 15
End for
Update a, A, C
Update X_α, X_β
t = t + 1
End while
Initialize the first two bats with X_α and X_β and rest (n-2) randomly
For each search agent:
randomly generate a solution in the range (lb, ub)
Train SVR on train dataset
Find fitness value (Accuracy on test dataset)
End for
Find the best solution among current solutions
Set X∗
Set f_{min}
while (t < maxIter)
for each search agent (i):
Update frequency using equation (8)
Update velocity using equation (9)
Update position using equation (10)
If rand > r_i^t
Update position using equation (11)
Train SVR on train dataset
Find fitness value (Accuracy on test dataset)
If ($x_i^t < x^*$) and (rand > A_i^t)
Update the position with the new one
Update r_i^t , A_i^t
End for
X∗ = current best
t = t + 1
End while
Return X∗

IV. EXPERIMENTAL WORK AND RESULTS

4.1 Data Preparation:

The dataset has been downloaded from Kaggle. It contains electricity consumption data of a smart home which is recorded by a smart meter with a time interval of one minute. The home contains 14 appliances and their consumption data is recorded in kilowatts [kW]. A few examples of the appliances are dishwasher, refrigerator, furnace, microwave, etc.

Before using the dataset in the algorithms, data cleaning was performed. Among all the features, ‘use [kW]’, ‘House overall [kW]’ were the same, hence the latter was dropped. The only source of energy being generated was Solar. So, the ‘gen [kW]’ and ‘solar [kW]’ were equal in value, hence the latter was dropped from the dataset.

Along with consumption data, the dataset also contains condition data that is defined by 10 different features like ‘temperature’, ‘pressure’, ‘humidity’, ‘windspeed’, etc. There were some categorical features/variables in the dataset like ‘icon’, ‘summary’, ‘cloudCover’ which do not contribute towards the target variable; hence these features were also dropped.

There were features like ‘Furnace 1 [kW]’, ‘Furnace 2 [kW]’, which represent the furnace in different rooms of the house. So, these features are clubbed in one feature and named ‘Furnace [kW]’ which represent all furnace of the house. Similarly, the features ‘Kitchen 1 [kW]’, ‘Kitchen 2 [kW]’, ‘Kitchen 3 [kW]’ are clubbed to a single new feature ‘Kitchen [kW]’ which represents total electricity consumed in the kitchens.

4.2 Feature Selection:

Feature selection is important in prediction problems because the model should be trained using important features only. Irrelevant features will affect the accuracy of the model. Feature selection is performed using Pearson’s coefficient. The correlation between target variable and independent variables was calculated and those features which were having a significant correlation with the target variable are used.

Some independent variables were highly correlated with each other. For example, ‘Temperature’ was highly correlated with ‘apparent Temperature’. So, to remove multicollinearity, these features were also dropped.

After performing exploratory data analysis and data cleaning, we then segregated the consumption data and weather data into different datasets because they were representing different information about data.

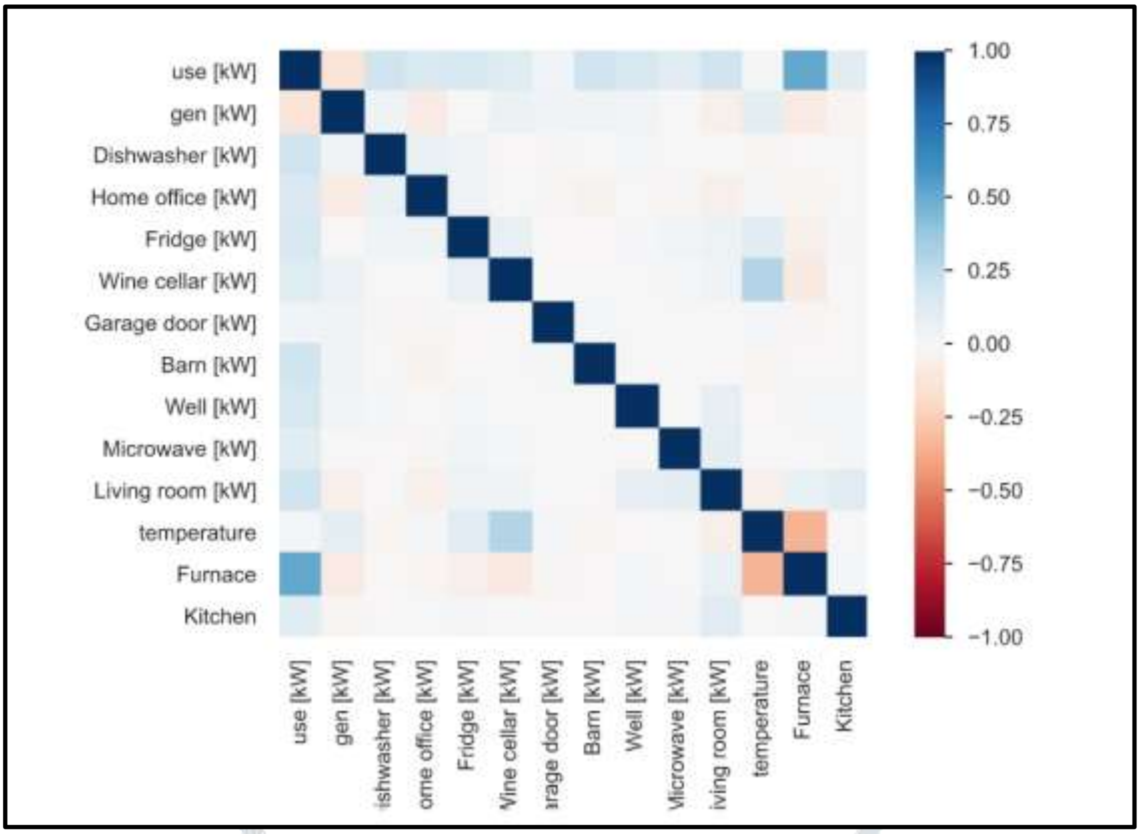

Figure 1: feature selection using Pearson coefficient

From figure 1, we can say that all appliances are contributing positively towards the target variable 'Use [kW]'. Hence, the dependent and independent variables in the dataset are well correlated with each other and this dataset can be now used to proceed with further research work.

The entire dataset was very large to be able to be applied to the algorithms. Hence, we decided to divide the dataset into manageable datasets which can be modelled using the algorithms. So, we decided to divide the dataset into temperature slabs of 5°f. With this, the dataset got divided into 22 different datasets which contain consumption data of a particular slab of temperature. Lastly, features of all dataset were normalized using MinMaxScaler before passing to the algorithms. This helped in reducing the time taken to train SVR and also keeps the shape of the data intact.

4.3 Grey Wolf Optimizer results:

After optimizing the C and Gamma hyperparameters using Grey Wolf Optimizer, the SVR was then trained with optimized parameters, and optimized consumption values were calculated for each temperature slab.

1. The C and Gamma Values:

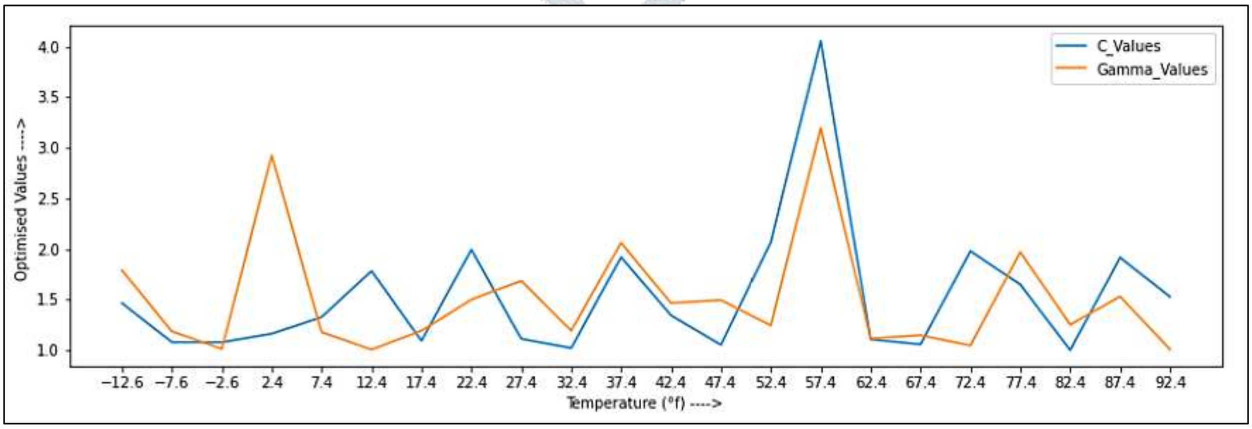

Figure 2: The C and Gamma Values using GWO

The figure 2 plot shows the optimized values of C and Gamma Parameter obtained for different temperature slabs using grey wolf optimizer.

2. The Accuracy:

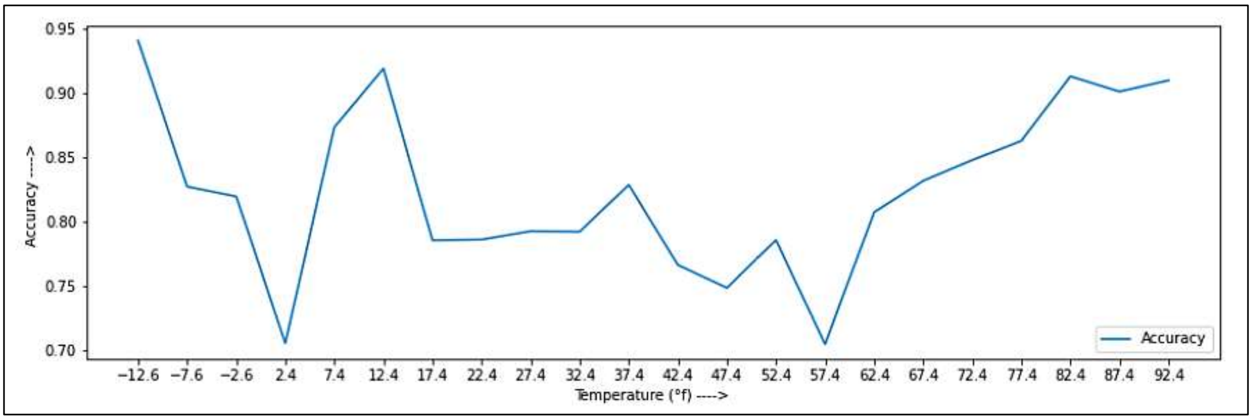

Figure 3: The Accuracy curve for GWO

Figure 3 shows the accuracy of SVR achieved while optimizing C and Gamma hyperparameters for the given range of temperature slabs. The minimum and maximum accuracy observed is 70.5% and 94.08%.

The minimum accuracy is observed at slab [2.4 to 7.4]. The reason for this is the overfitting of the model at that slab. The c and gamma are 1.2 and 2.9 respectively. The large value of gamma made the model develop an affinity towards the training set examples such that when we try to validate on the test dataset, it deteriorates the performance.

The mean accuracy obtained by applying Grey Wolf Optimizer is 82.52%.

3. The optimized consumption values:

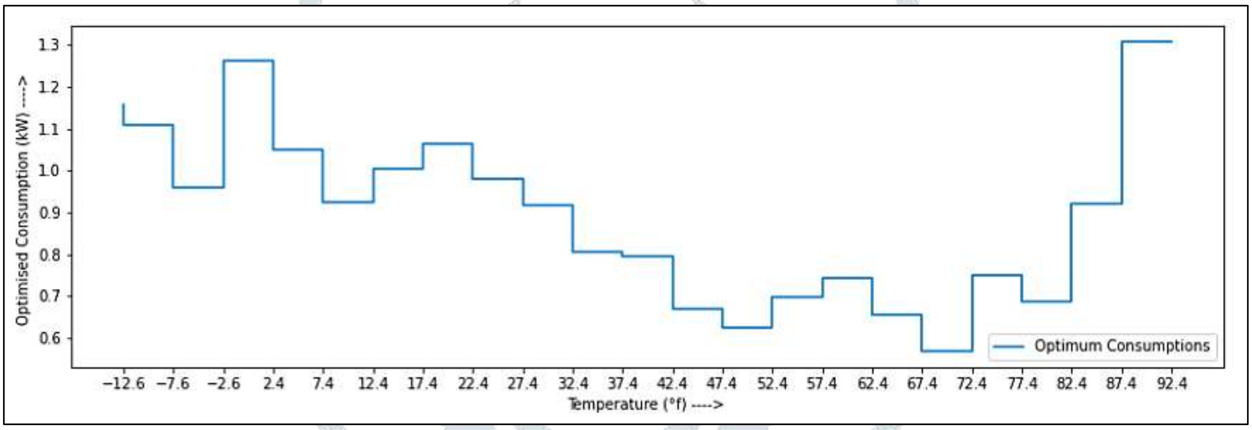

Figure 4: Optimized Electricity Consumption using GWO

Figure 4 shows the optimal electricity consumption obtained for different temperature slabs. The consumption is more in extreme temperatures like [-12.6°f to 2.4°f] and [87.4°f to 92.4°f].

4.4 Bat Algorithm results:

After optimizing the C and Gamma hyperparameters using Bat Algorithm (BA), the SVR was then trained with optimized parameters, and optimized consumption values were calculated for each temperature slab.

1. The C and Gamma Values:

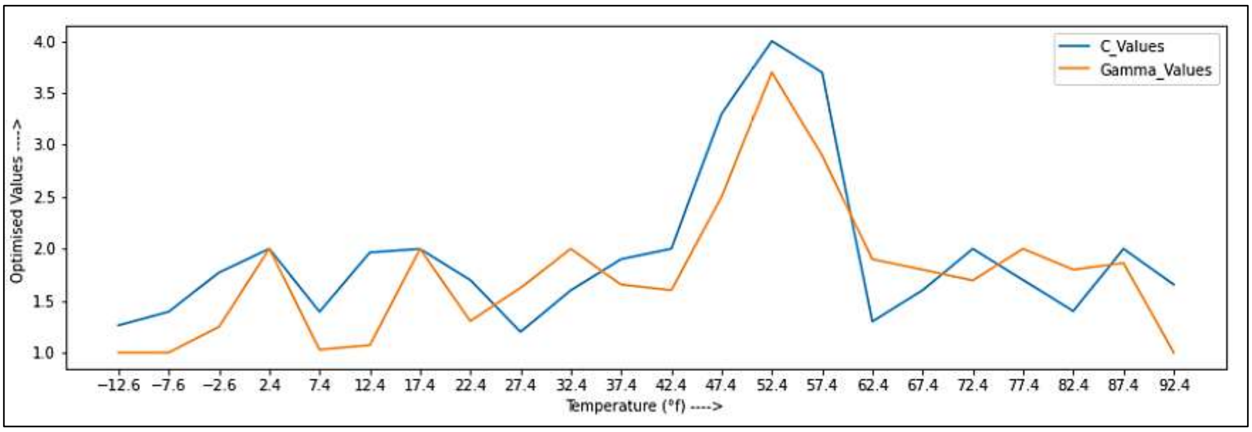

Figure 5: The C and Gamma Values using Bat Algorithm

Figure 5 shows the optimized values of C and Gamma Parameter obtained for different temperature slabs obtained using Bat Algorithm.

2. The Accuracy:

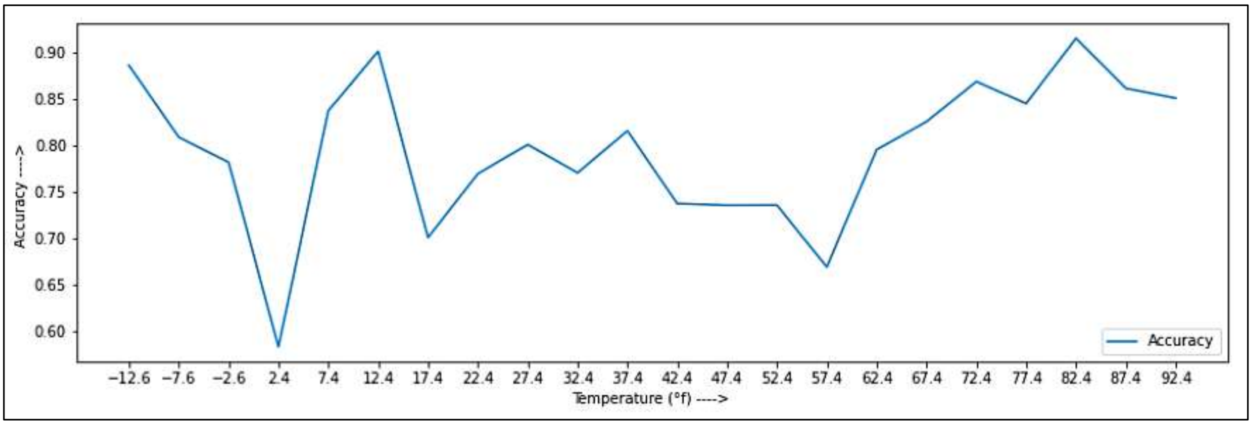

Figure 6: The Accuracy curve for BA

Figure 6 shows the accuracy of SVR achieved while optimizing C and Gamma hyperparameters for the given range of temperature slabs. The minimum and maximum accuracy observed are 58.31% and 91.49%.

The minimum accuracy is observed at slab [2.4 to 7.4]. The reason for this is similar overfitting of the model that was also observed with GWO. The C and Gamma are 1.98 and 2.0 respectively. Although the value of C is increased compared to GWO but still fails to regularize the model for this slab.

The mean accuracy obtained by applying Bat Algorithm is 79.50%.

3. The optimized consumption values:

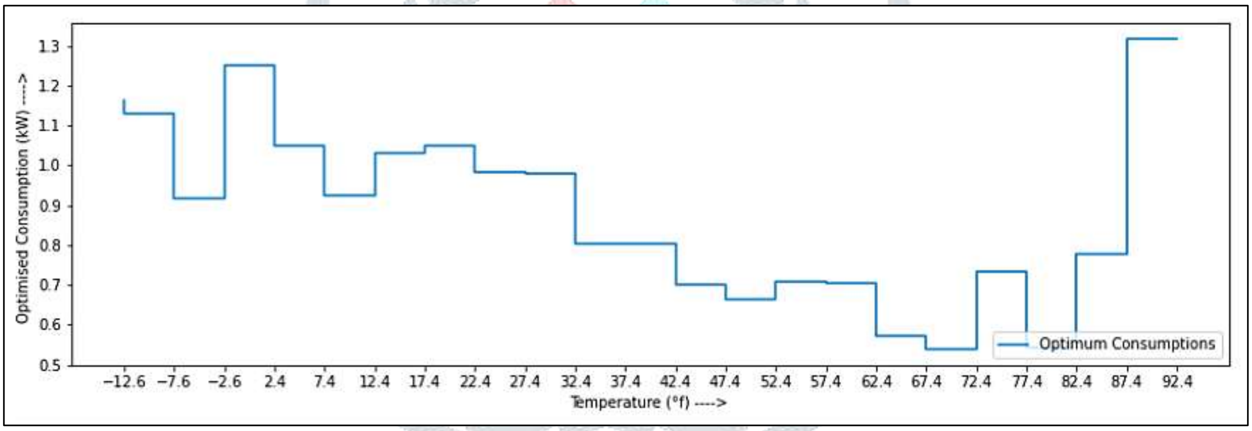

Figure 7: Optimized Electricity Consumption using BA

Figure 7 shows the optimal electricity consumption obtained for different temperature slabs. The consumption is more in extreme temperatures like [-12.6°f to 2.4°f] and [87.4°f to 92.4°f].

4.5 Hybrid Grey Wolf-Bat Algorithm results:

After optimizing the C and Gamma hyperparameters using Hybrid Grey Wolf-Bat Algorithm (GWOBA), the SVR was then trained with optimized parameters, and optimized consumption values were calculated for each temperature slab.

1. **The C and Gamma Values:**

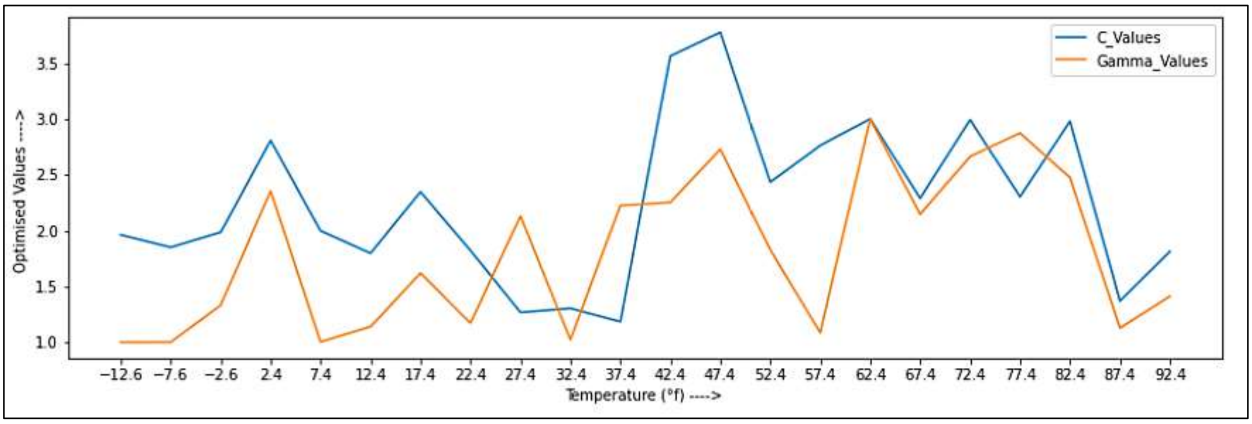

Figure 8: The C and Gamma Values using GWOBA

Figure 8 shows the optimized values of C and Gamma Parameter obtained for different temperature slabs obtained using GWOBA.

2. **The Accuracy:**

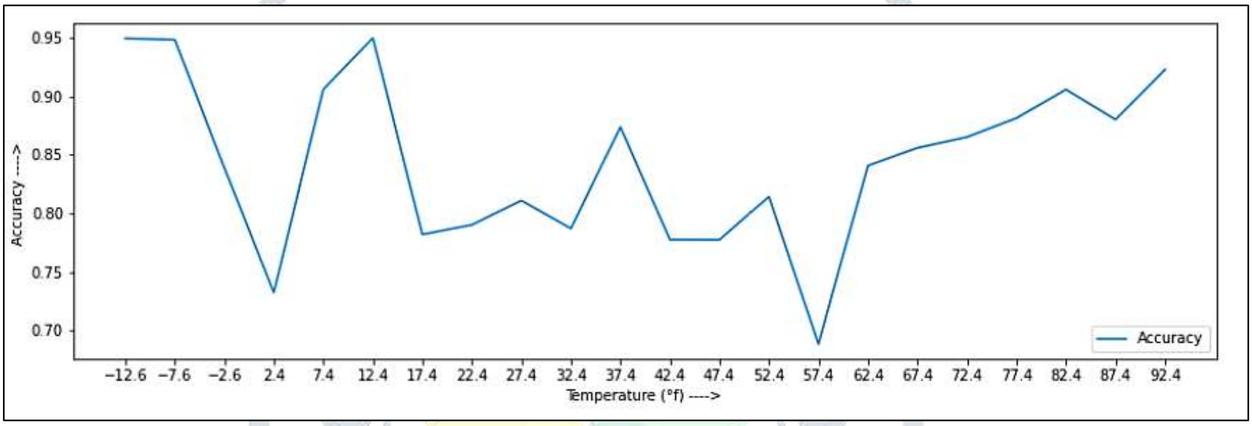

Figure 9: The Accuracy curve for GWOBA

Figure 9 shows the accuracy of SVR achieved while optimizing C and Gamma hyperparameters for the given range of temperature slabs. The minimum and maximum accuracy observed is 68.84% and 94.96% respectively.

The minimum accuracy is observed at slab [57.4 to 62.4]. The slab [2.3, 7.4] manages to achieve its overall best accuracy of 73.25%. The C and Gamma values are 2.80 and 2.35 respectively. Hence, we can say that a good amount of regularization helped in overcoming the overfitting of the model in this slab.

The mean accuracy obtained by applying Hybrid Grey Wolf-Bat Algorithm is 84.42%.

3. **The optimized consumption values:**

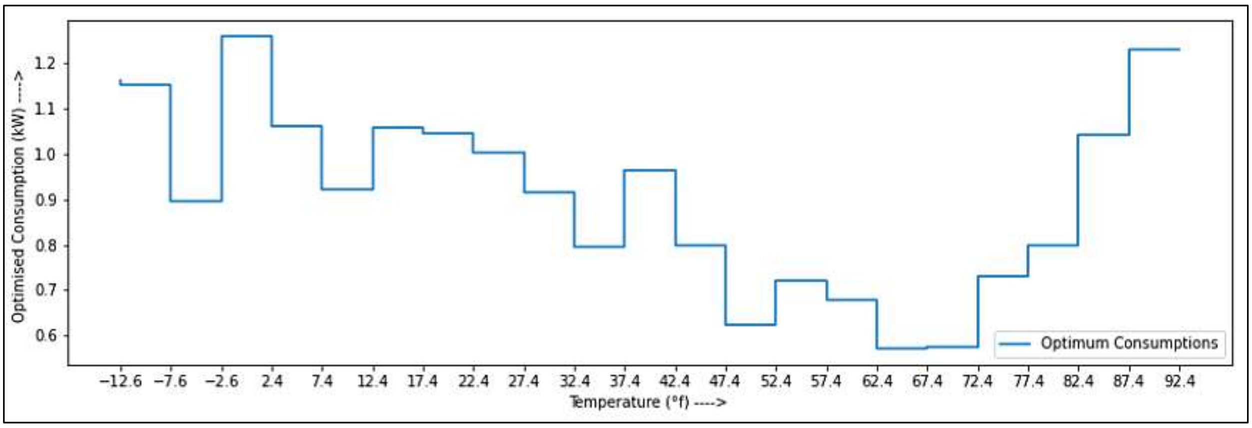

Figure 10: Optimized Electricity Consumption using GWOBA

Figure 10 shows the optimal electricity consumption obtained for different temperature slabs. The consumption is more in extreme temperatures like [-12.6°f to 2.4°f] and [87.4°f to 92.4°f].

V. OBSERVATIONS

5.1 Comparison of Accuracies:

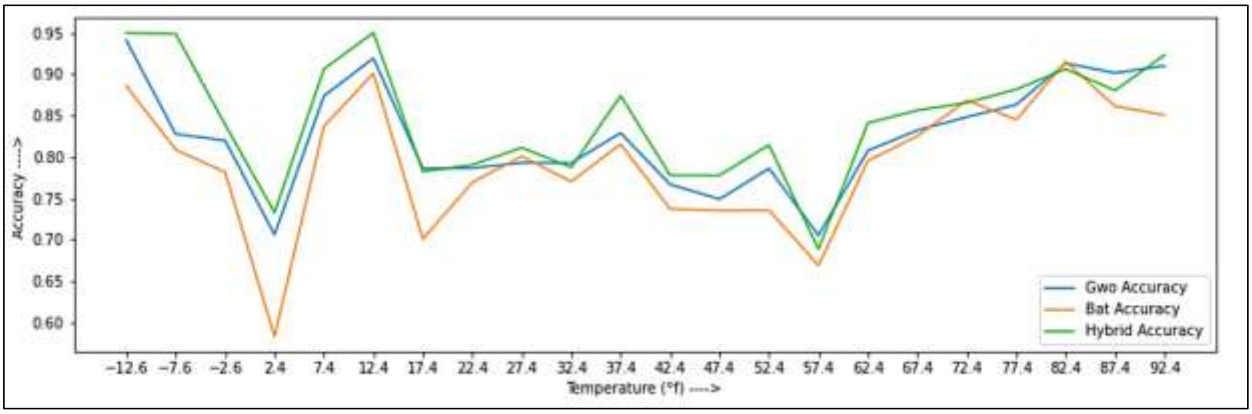

Figure 11: Comparison of Accuracies

Figure 11 shows the accuracy of SVR using cross-validation obtained from different algorithms on the same dataset.

The mean accuracy obtained by GWO, BA, and GWOBA is 82.52%, 79.5%, and 84.42% respectively. The Hybrid algorithm shows better results as per the hypothesis. The accuracy of GWOBA is highest for 16 out of 22 slabs. For 2 out of 22 slabs, the accuracy was equal to either GWO or BA. For 4 out of 22 slabs, the accuracy of GWOBA was less than the other algorithms but the maximum difference is only 2%. BA performance is consistently less than the other two algorithms. Only in 2 out 22 slabs, BA performed slightly better than GWO. GWO's performance curve is like the median of BA's and GWOBA's curve. But GWO has performed better in terms of accuracy whenever GWOBA fails to achieve its best.

5.2 Comparison of Optimal Values:

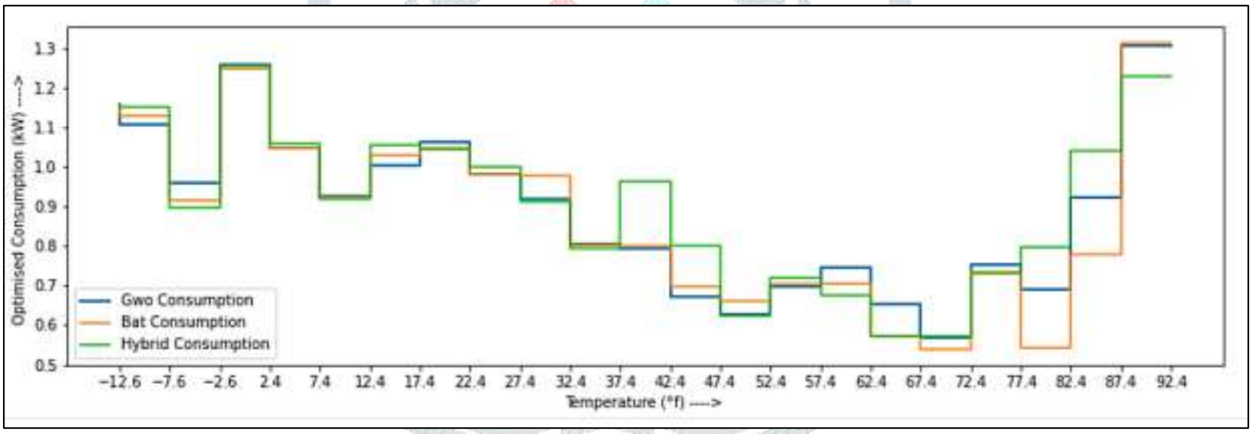

Figure 12: Comparison of Optimal Consumption Values

Figure 12 shows the optimal consumption for different algorithms. The consumption is more in extreme temperatures like [-12.6°f to 2.4°f] and [87.4°f to 92.4°f]. Based on these values, an optimal sanctioned load value can be selected for a particular temperature slab. Among these, we suggest considering GWOBA's values as the optimal values because it has achieved the highest accuracies for the majority of slabs. Another method to consider is, choosing the values of that algorithm whose accuracy is highest for that particular slab.

VI. CONCLUSION AND FUTURE WORK

The process of optimizing sanctioned load/power of a smart home was achieved by modelling the previous year's electricity consumption data using SVR. GWOBA shows the highest accuracy of 84.42% followed by GWO and BA respectively. With a maximum accuracy drop of only 2% in 4 out of 22 slabs, we conclude that in those cases where GWOBA fails to give the optimized result, it fails by a very small margin. Thus, it is highly recommended for problems where the loss associates with a high bounty, for example, problems related to electricity, fuel, nuclear, etc.

For the slab [2.4 to 7.4], only GWOBA was able to neutralize the overfitting of the model with a high value of the C parameter. Whereas, the other two algorithms got stuck at a local optimum value which results in overfitting. Thus, GWOBA has a better ability to search the solution space without falling prey to a local optimal value. By comparing the c and gamma values obtained from the algorithms in fig. 2, 5, and 8, we can conclude that GWOBA has explored a wider range of values.

Although the hybrid algorithm has shown prominent results there is still room for improvements. Other metaheuristic algorithms should be investigated. Another promising field of study would be to explore genetic algorithms and then the best algorithm can be decided.

REFERENCES

[1] Central Electricity Authority, “Annual Report 2019–2020”, Government of India, Ministry of Power, New Delhi, India, 2020. [Online]. Available: https://cea.nic.in/old/reports/annual/annualreports/annual_report-2020.pdf

[2] T. Vijayapriya and D. Kothari, "Smart Grid: An Overview", Smart Grid and Renewable Energy, Vol. 2 No. 4, 2011, pp. 305-311. doi: 10.4236/sgre.2011.24035.

[3] Nick Jenkins, Chao Long and Jianzhong Wu, “An Overview of the Smart Grid in Great Britain”, Engineering Vol. 1, No. 4, 2015, pp. 413-421. doi: 10.15302/J-ENG-2015112.

[4] National Smart Grid Mission, “Vision Roadmap”, Government of India, Ministry of Power, New Delhi, India, 2020. [Online]. Available: https://www.nsgm.gov.in/sites/default/files/SG-Vision-Mission-Goals.pdf

[5] El-Ghazali Talbi, “Metaheuristics: From Design to Implementation”, Hoboken, NJ: Wiley, 2009

[6] Fred Glover, “Future paths for integer programming and links to artificial intelligence”, Computers & Operations Research, Vol. 13, No. 5, 1986, pp. 533-549

[7] Mirjalili, S., Mirjalili, S.M., Lewis, A., “Grey wolf optimizer”, Advances in Engineering Software 69, 2014, pp. 46-61 doi: 10.1016/j.advengsoft.2013.12.007.

[8] Yang, X.S, “A new metaheuristic bat-inspired algorithm”, in Nature Inspired Cooperative Strategies for Optimization (NICSO), 2010, pp. 65–7

[9] Glover, F., McMillan, C., “The general employee scheduling problem: an integration of MS and AI”, Computers & Operations Research, Vol. 13, No. 5, 1986, pp. 563-573, doi: 10.1016/0305-0548(86)90050-X.

[10] Mladenović, N., Hansen, P., “Variable neighborhood search”, Computers & Operations Research, Vol. 24, No. 11, (1997), pp. 1097–1100

[11] Beheshti, Zahra & Shamsuddin, Siti Mariyam, “A review of population-based meta-heuristic algorithm”, International Journal of Advances in Soft Computing and Its Applications. 5, 2013, pp. 1-35

[12] Deepak Gupta, Ritika Tilwalia, Anisha Jain, “Optimization of electricity consumption using evolutionary algorithms”, in Proceedings of International Conference on Innovative Computing & Communication (ICICC), 2020

[13] R. Tilwalia, A. Jain, and D. Gupta, “Optimization of Electricity Consumption using Grey Wolf Algorithm,” 2020 IEEE 5th Int. Conf. Comput. Commun. Autom. ICCCA 2020, pp. 401–407, 2020, doi: 10.1109/ICCCA49541.2020.9250899.

[14] M. ElGayyar, E. Emary, N. H. Sweilam, and M. Abdelazeem, “A hybrid Grey Wolf-Bat Algorithm for Global Optimization,” in The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018), Vol. 723, Springer, 2018, pp. 3-12

[15] B. Schölkopf, K. Tsuda, and J.P. Vert, “A Primer on Kernel Methods,” in Kernel Methods in Computational Biology, The MIT Press, 2004, doi: https://doi.org/10.7551/mitpress/4057.003.0004

[16] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, D., Brucher, M., Perrot, M., & Duchesnay, E., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, 2011, pp. 2825–2830.

[17] C. Cortes and V. Vapnik, “Support-vector networks,” Machine Learning, vol. 20, no. 3, 1995, pp. 273–297, doi: https://doi.org/10.1007/BF00994018

[18] Yongli Zhang and Yuhong Yang, “Cross-validation for selecting a model selection procedure,” Journal of Econometrics, vol. 187, no. 1, 2015, pp. 95-112, doi: https://doi.org/10.1016/j.jeconom.2015.02.006.